NTT DATA at TREC-7: system approach for ad-hoc and filtering

Hiroyuki Nakajima, Toru Takaki, Tsutomu Hirao and Akira Kitauchi
Laboratory for Information Technology
NTT DATA Corporation
Kowa Kawasaki Nishi-guchi Bldg., 66-2 Horikawa-cho,
Saiwai-ku, Kawasaki-shi, Kanagawa 210-0913 Japan
{nakajima,takaki,hirao,kitauchi}@lit.rd.nttdata.co.jp

1 Introduction

In TREC-7, we participated in the ad-hoc task (main task) and the filtering track (sub task). In the ad-hoc task, we adopted a scoring method that used co-occurrence term relations in a document and specific processing in order to determine which conceptual parts of the documents should be targeted for query expansion. In filtering, we adopted a machine-readable dictionary for detecting idioms and an inductive learning algorithm for detecting important co-occurrences of terms. In this paper, we describe the system approach and discuss the evaluation results in brief for our ad-hoc and filtering in TREC-7.

2 Ad-hoc Track

This section describes the method we adopted that allowed by the ad-hoc task to obtain the output results.

2.1 System description

Figure 1 shows the processing procedure in our system. The structure of the databases and the action of each processing module are explained as follows.

(a) Databases

As for each data set of the *Financial Times*-1991-1994 (FT), *Federal Register*-1994 (FR94), *Foreign Broadcast Information Service* (FBIS) and the *LA Times* (LATIMES) retrieved by TREC-7 ad-hoc, the index is made respectively.

(b) Processing module

(1) Query term extraction module

First, a term that will be used for retrieval and scoring is extracted from the input topics, and a list of retrieval terms is made. The stopwords (550 words) are then deleted, and the extracted terms are converted into root words.

(2) Query term limitation module

To do the score calculation processing efficiently, the terms are limited from the term list that was obtained by the extraction module. In this term limitation processing, a term's degree of importance is defined by the *idf* (Inverse Document Frequency) value, and terms which have a low degree of importance are deleted from the query term list before the retrieval is processed. The *idf* value is

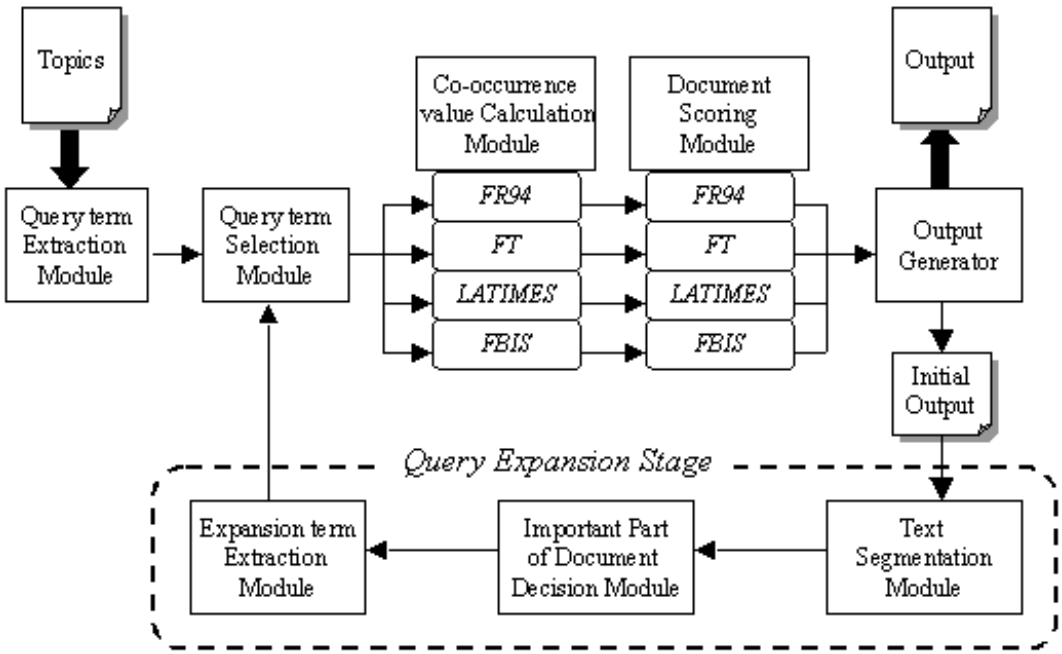

Figure 1: Flow of the processing procedure(ad-hoc task)

calculated from the corpus that consists of all data sets used for TREC-7 ad-hoc. Terms that have low degrees of importance are disregarded because it is assumed that they do not have a big influence on the score of each document for topics. In addition, the degree of importance degree of each division, title, description and narrative, in the input topics is determined, and these values were used to set the degree of importance of the query term and in the scoring. When the degree of importance of a term which appeared in the title was assumed to be 1.0, this system adjusted the degree of importance of description and narrative to 0.1.

(3) Co-occurrence value calculation module

The terms used for retrieval and scoring are extracted from the input topics. Processing is done to add to the score of the document in consideration of the importance of co-occurrence, when two related terms appear in the same document. In this module, the degree of importance of each co-occurrence in a document is calculated. Details of the processing method are described later.

(4) Document scoring module

We used a scoring method based on $tf - idf$ as a basis of the score calculation. Here, the degrees of importance of the co-occurrence terms calculated in the previous module are added, and a final document score is calculated. The score of each document is calculated using the document frequency of the data set to which the document belongs.

(5) Output generator module

The output generator module merges each result from a data set into one result. This system does not normalize the score between documents included in each data set. Each document is sorted in order of score, and the module generates a formatted output.

(6) Query term expansion module

To decrease retrieval leakage, we adopted the query term expansion. The basis of this query term expansion is a local feedback method, which involves the top ranked documents retrieved by the original query. Terms to be included in the expanded term are selected from previously the processed parts of the document that are considered to be important enough. We will discuss how to determine the level of importance later.

2.2 Degree of importance of co-occurrence appearance

We think that when query terms co-occur in the documents, it is indication that the document is more relevant to the query than documents in which more than once query terms appear. The degree of importance of the co-occurrence is defined according to this co-occurrence condition, and processing by which a co-occurrence important value adds to the score calculated by usual frequency of the term is executed further. The co-occurrence important values are decided according to the following parameters:

- The distance between adjacent terms: ρ (It is important if two terms appear near each other)
- The relative relationship between terms: σ (It is not important even if no related terms co-occurred)
- The importance of the co-occurred term: τ (Co-occurrence with a term that is not considered important is not crucial)

The degree of importance $cw(t_i, t_j)$ of the co-occurrence of terms t_i and t_j is defined by the next expression.

$$cw(t_i, t_j) = \rho(t_i, t_j) \times \sigma(t_i, t_j) \times \tau(t_i) \times \alpha \tag{1}$$

where,

$$\rho(t_i, t_j) = \begin{cases} \dfrac{\lambda - d_{ij}}{\lambda} & if : \lambda > d_{ij} \\ 0 & :\text{otherwise} \end{cases} \tag{2}$$

$$\sigma(t_i, t_j) = \frac{rtf_{ij}}{atf_i} \tag{3}$$

$$\tau(t_i) = \log(\frac{N}{df_j}) \tag{4}$$

Here,
λ is a parameter of the adjacent appearance distance,
d_{ij} is the distance between t_i and t_j (in number of words),
rtf_{ij} is frequency in the database of t_i that t_j appears with an adjacent appearance distance of λ words or less,
atf_i is the appearance frequency of t_i in the database,
N is the total number of the documents in the database,
df_j is the number of documents in which appears t_j

The degree of the co-occurrence importance of these words is calculated between two words.

2.3 Query term expansion

We applied the local feedback method as a query term expansion. This local feedback analyzes the document retrieved by the initial query and usually obtains the expansion terms. An extended term can be obtained from the entire document using this method but we assumed that its could be obtained from an important part of the document. An important part is determined from the segment which is the unit of a consecutive sentence. There is a method of expanding the query term from the sentence, from which the degree of importance of each sentence is calculated and the importance degree is high. However, sentences which do

not include the query term cannot be extracted by this method. Moreover, there is a method of deciding which parts of the document are important by dividing the document into chapters and paragraphs, etc. This method has a problem in that parts containing unrelated subjects may be used to expand the query term when multiple subjects are included in the unit of the document structure.

The degree of importance of each sentence is calculated, and the change in the degree of importance is used to determine the segment's range. In general, the degree of importance of the sentence in one document performs the change. The range of the segment is determined by assuming the part where the degree of importance of the sentence is low to be a gap in meaning and then dividing the document into the segments. The sum total of the degrees of importance of the sentences in each segment is assumed to be the degree of importance of the segment, and the segment with a high degree of importance becomes a query term extended object. The terms included in the selected segment become the candidate of extended terms. But all terms are not used for query expansion. We assumed the term which has higher value of *idf* to be an extended word.

2.4 Results

We submitted three processing results to NIST. The method of each processing of the submitted result is as follows. Table 1 shows the evaluation result of TREC-7 ad-hoc by each method.

[A] *nttdata7Al0* (judged)

The query term was generated from all fields of topics (title, description, and narrative). The method of the term frequency base ($tf - idf$) was used for scoring, and, in addition, co-occurrence information was applied to the score element. Here, we did not use feedback for the expansion of the query term.

[B] *nttdata7Al2* (judged)

This query expansion processing was executed after processing the *nttdata7Al0*. Twenty high-ranking documents of the initial retrieval result were targeted for local feedback. In addition, we executed the query term expansion by specifically processing an important part in the document. Thirty words were selected in order of the value of idf and these words were assumed to be extended query terms.

[C] *nttdata7At1* (submitted but NOT judged)

The processing method was similar to that of *nttdata7Al2*. Only the title filed of the topics was used for query term generation.

[X] *nttdata7Anorm* (NOT submitted: for comparison)

This method only scored by the tf-idf base by processing the *nttdata7Al0*. The degree of importance of the co-occurrence was not considered.

nttdata7Anorm vs *nttdata7Al0*

First, we compared the *nttdata7Anorm* and *nttdata7Al0* which applied the degree of importance of the co-occurrence. The average precision improved by 4.5% when the degree of the importance of the co-occurrence was used. Moreover, the relevant-retrieved number improved from 2580 to 2624. When the degree of importance of the co-occurrence was applied, a decrease in the precision was observed in the lower recall part (0.0,0.1,0.2) and the top document part.

RUN ID	[X] *nttdata7Anorm* (NOT submited)	[A] *nttdata7Al0* (judged)		[B] *nttdata7Al2* (judged)	
Relevant-Retrieved	2580	2624		2656	
Recall		(%Change vs [X])		(%Change vs [A])	
at 0.00	0.7384	0.6738	(-8.75%)	0.6715	(-0.34%)
at 0.10	0.4495	0.4366	(-2.87%)	0.4476	(+2.52%)
at 0.20	0.3592	0.3469	(-3.42%)	0.3523	(+1.56%)
at 0.30	0.2675	0.2816	(+5.27%)	0.2947	(+4.56%)
at 0.50	0.1567	0.1818	(+16.02%)	0.1934	(+6.38%)
at 0.70	0.0764	0.1007	(+31.81%)	0.1090	(+8.24%)
at 1.00	0.0005	0.0013	(+160.00%)	0.0003	(-76.92%)
Average	0.1943	0.2032	(+4.58%)	0.2113	(+3.99%)
At 5 docs	0.4800	0.4400	(-8.33%)	0.4280	(-2.73%)
At 10 docs	0.4340	0.4100	(-5.53%)	0.4080	(-0.49%)
At 20 docs	0.3490	0.3480	(-0.29%)	0.3973	(+6.03%)
At 30 docs	0.3053	0.3080	(+0.88%)	0.3200	(+5.46%)
At 100 docs	0.1932	0.1972	(+2.07%)	0.2050	(+3.90%)
At 500 docs	0.0832	0.0525	(+1.92%)	0.0862	(+1.38%)
R-Precision	0.2398	0.2493	(+3.96%)	0.2483	(-0.40%)

Table 1: TREC-7 ad-hoc result

nttdata7Al0 **vs** *nttdata7Al2*

Retrieval accuracy was improved when query term expansion was used. When *nttdata7Al2* was compared with *nttdata7Al0*, the average precision improved by 4%. In particular, an improvement in recall was observed in the middle recall range. The improvement of the recall of 30 top-ranking documents is high. In general, it is called 20 or 30 top-ranking documents of the retrieval result at most that the user reads, thus this method might be useful.

3 Filtering Track

Our filtering track system is based on Rocchio feedback[8] and dynamic feedback optimization (DFO)[1]. Rocchio feedback and DFO have shown fine results in past routing tasks in TREC. In recent years, several methods have been proposed that enhance the Rocchio feedback and DFO so that they are able to handle the weights of co-occurring pairs of terms, and they succeeded in improving the precision of results.

We think that co-occurring pairs of original query terms are especially important in relevance feedback, because:

1. We can use filtering profiles to show important terms to users. But users find it is very hard to imagine the relationships between terms. If we can show important co-occurrence pairs to users, they will find it easier to grasp the relationships between terms.

2. Once indices of terms are read into the memory of system, the system can check the co-occurrence of terms in memory, and this doesn't strongly affect processing time. In many cases, the terms in original queries are important, and therefore, they should be used in relevance feedback, and read into the memory. If the system can correctly find important pairs of query terms, these pairs improve the results in a short processing time.

We have constructed a system that is very similar to that reported by AT&T at TREC-6[9], but with additional features to handle co-occurrence pairs.

1. We added an idiom dictionary that is constructed using an English-Japanese dictionary to enable idiom indexing. We consider idioms to be a special case of term co-occurrence.

2. We added an inductive-learning algorithm to detect the co-occurrence of more than two terms.

3.1 System

We give a brief description below.

3.1.1 Building inverted files

Stopwords and stemming

We used the stemming algorithm of Porter's[6], and removed terms in stopword lists of freeWAIS[5] and SMART.

Idioms and phrases

In general, we construct inverted files of documents by using term-based indexing. However, some terms have special meanings that are quite different from the meaning of each term, i.e., their meaning is idiomatic. If we index idioms by their constituent terms, we lose the opportunity to use their special meanings.

The meanings of terms and idioms can often be clarified by paraphrasing them, or, translating them into another language. At TREC-7, we used an English-Japanese dictionary to first translate expressions into Japanese and then translate their constituent terms. If the meanings are quite different, the expression is identified as idiomatic, and it is added to our idiom dictionary.

This idiom dictionary is then used when we build inverted files from documents. The idiomatic expressions are marked, and these markers are used to build idiom indices. The constituent terms are not indexed.

After this idiom processing, any pair of adjacent words that are neither stopwords nor idioms is regarded as a phrase, and both term and phrase indices are built.

Term weighting

We used 'lnc' and 'ltc' schemes, as in SMART in TREC2[2].

3.1.2 Refining term weights through feedback

We used 'multi-pass' Rocchio feedback[9] and 2-pass DFO for refining term weights. We used 1000 terms and 100 phrases, and set the size ratio of the vectors of original query (α), relevant documents (β), and non-relevant documents (γ) to be 1:8:-8 in Rocchio feedback.

3.1.3 Detecting term pairs and weightings

Co-occurrence of 2 terms

After 1st-pass of Rocchio feedback, we detected co-occurrence pairs of 2 terms. We regarded any pairs of original query terms and top 100 terms (weighted by the Rocchio formula) as co-occurrence pairs, and calculated their weights in the same way that the term weights of Rocchio feedback are calculated. $'tf'$ and $'idf'$ of pairs are calculated as follows:

tf : the smaller value of tf of the two terms.
idf : calculate the number of documents (N_{pair}) which include the pair of the term#1 and term#2 as follows:

$$\begin{aligned} N_{pair} &= collection\_size \\ &\times (number\_of\_document\_including\_term\#1/collection\_size) \\ &\times (number\_of\_document\_including\_term\#2/collection\_size) \end{aligned} \quad (5)$$

We used 200 positively weighted pairs.

Detecting co-occurrence of more than 2 terms and phrases

Co-occurrence of 2 terms has reported to be effective in improving the precision of results. We assumed if we apply the same methods for the co-occurrence of more than two terms, it will improve the precision. However, calculating weights of co-occurrence of many terms needs a lot of computation, and most pairs have quite small weights.

To find pairs that have large weights, we used an inductive learning algorithm (based on $C4.5$[7], added some modifications) to detect terms and phrases pairs ('rules') that appear frequently in relevant documents and do not appear in irrelevant documents. We calculated tf and idf of each pairs (rules) as follows:

tf : smallest value of tf in the rule (We neglected negative terms in calculating tf).
idf : calculate the number of documents N_{rule} that include the rule as:

$$\begin{aligned} N_{rule} &= collection\_size \\ &\times \prod_{each\_term\_appears\_in\_the\_rule} f(term) \end{aligned} \tag{6}$$

We defined $f(term)$ as below:

If the term is positive one in the rule,

$$f(term) = number\_of\_documents\_including\_term / collection\_size$$

If the term is negative one in the rule,

$$f(term) = (collection\_size - number\_of\_documents\_including\_term) / collection\_size$$

3.2 Results

We submitted results of routing and filtering tracks to NIST. In the routing track, we made a slight mistake in computing the idf of pairs ('nttd7rt1'), so we later put the fixed result ('nttd7rt2').

Table 2 shows the results of the routing track. Our results were the best of all participants in TREC-7 routing track. (The results of nttd7rt1 are the same as those of nttd7rt2).

Run	Average Precision	Best	> average	Average	< average	Worst
nttd7rt2	.5139	30	11	7	2	0

Table 2: Results for nttd7rt2, routing

We tested the effects of our methods experimentally. Table 3 shows the effects of various parameters on the Rocchio feedback, without using our new methods. (Parameters '$\alpha : \beta : \gamma = 2 : 4 : -1$' are used in SMART at TREC2).

$\alpha : \beta : \gamma$	$1 : 8 : -8$			$2 : 4 : -1$		
number of terms	100	300	1000	100	300	1000
Ave.Prec	.4475	.4579	.4618	.4280	.4396	.4430

Table 3: Effects of various parameters on Rocchio feedback

Table 4 shows the effects of using our idiom processing ('+idiom') and detecting the co-occurrence of two terms ('2pair') before weight refining.

Run	1000 terms($\alpha:\beta:\gamma = 1:8:-8$)	1000 terms + idiom	1000 terms + idiom + 2pair
Ave.Prec	.4618	.4726	.4771

Table 4: Effects of idiom processing and detection of two-term co-occurrences

In filtering track, the results are slightly better than the average of participants (nttd7bf1) and near the best of participants (nttd7bf2).

Run	Best	> average	Average	< average	Worst
nttd7bf1	6	7	23	7	7

Table 5: Results for nttd7bf1, filtering F1 measure

Run	Best	> average	Average	< average	Worst
nttd7bf2	16	4	22	6	1

Table 6: Results for nttd7bf2, filtering F3 measure

4 Conclusion

We described our system approach and discussed the evaluation results for ad-hoc and filtering in TREC-7. Results in filtering track were quite fine, especially in routing track.

The inductive-learning algorithm is used to detect co-occurrence pairs in the filtering track. This method is only effective when sufficient training documents are used. If only a few training documents are available, using non-judged documents as provisional irrelevant documents might be effective[3, 4].

Our investigation of idiom processing is still in progress, and we have not tried it with any languages other than Japanese. However, languages that have linguistic ancestors in common with English may not be suitable, because they have common lexical borrowings, including idioms.

References

[1] Chris Buckley and Gerard Salton. Optimization of relevance feedback weights. In *SIGIR*, pages 351–357, 1995.

[2] Chris Buckley, Gerard Salton, and James Allan. Automatic routing and ad-hoc retrieval using SMART: TREC2. In *TREC-2*, pages 45–55, 1994.

[3] Hiroyuki Nakajima and Tsuyosi Kitani. Inductive learning using document frequency for relevance feedback (in Japanese). Technical Report FI-45-97, Information Processing Society of Japan(IPSJ), 1997.

[4] Hiroyuki Nakajima, Tsuyosi Kitani, and Mamoru Okada. Improving relevance feedback through recognizing co-occurences of query words (in Japanese). *Transcations of IPSJ*, March 1999. (To be appeared).

[5] Ulrich Pfeifer and Tung Huynh. Freewais-sf, 1994.
ftp://ls6-www.infomatik.uni-dortmund.de/pub/wais/freeWAIS-sf-1.0.tgz.

[6] Porter, M.F. An algorithm for suffix stripping. *Journal of the Society for Information Science*, 3(14):130–137, 1980.

[7] Quinlan, J. R. *C4.5: Programs for machine learning.* Morgan Kaufman, 1993.

[8] Rocchio, J. J. Relevance feedback in information retrieval. In *The SMART Retrieval System*, pages 313–323. Prentice-Hall, 1971.

[9] Amit Singhal. AT&T at TREC-6. In *TREC-6*, pages 215–226, 1998.