NTT DATA: Overview of system approach at TREC-8 ad-hoc and question answering

Toru Takaki
Research and development Headquarters
NTT Data Corporation
Kayabacyo Tower Bldg., 1-21-2, Shinkawa,
Chuo-ku, Tokyo 104-0033 Japan
E-mail: takaki@rd.nttdata.co.jp

1 Introduction

In TREC-8, NTT Data Corporation participated in the ad-hoc task and question answering track. In this paper, we describe our system approach and discuss the results. The summary of each task of our approach is shown below.

Ad-hoc

We submitted five results as official runs. Two kinds of results, long query results (title, description and narrative) and short query results (title and description), were submitted. Another kind of result that applied query expansion technique or not applied was also submitted. In our work at TREC-8, we concentrated our interest on extraction of the query terms. Specifically, we applied a removal technique of negative information in topics and specification of multiword phrases.

Question Answering

The question answering (QA) track is first attempt in TREC. We gave priority to the following verification for the QA track: (1) the effectiveness of technique by surface-text-based information in the text and (2) application of the information extraction technique. In our QA track, the following processing was done: (1) decision of answer form by question analysis, (2) passage scoring and selection for detailed analysis of the answer after initial retrieval, and (3) information extraction that look for words or phrases that match the form of the answer. We submitted two results to the answer categories of different strength respectively. A retrieval technique like ad-hoc is effective in a category answered by 250 bytes or less in our evaluation but the question analysis is important for a stricter category answered by 50 bytes or less.

2 Ad-hoc task

We concentrated our interest on the query term selection from the topics. The terms describing criteria of non-relevance in the topic sentence were not applied as query terms. Multiword phrases and each term composing them are applied as query terms. And, pseudo relevance feedback was done in the query term expansion. This method is similar to Local Context Analysis (LCA) [3].

2.1 Approach

In ad-hoc, we processed the retrieval as follows.

Index

The index was made from stemmed text within <TEXT> and </TEXT> tags in the data set of TREC-8 (in TREC disks 4 and 5: the *Financial Times* 1991-1994, *Federal Register*-1994, *Foreign Broadcast Information Service* and the *LA Times*). In our last TREC-7, four indexes by document source were made, and the relevance ranking processing was done for each index, then those results were merged into one result [1]. However one index was constructed, and the retrieval processing was done in TREC-8.

Search and Relevance ranking

Our ranking processing has 4 steps:

Step 1: Query term selection from topics

(1) Deletion of negative sentences from topics
Sentences discussing criteria of non-relevance in the narratives (such as "Documents that describe... are not relevant.") are removed.

(2) Deletion of stopwords and stemming
The stopwords were deleted by using the list of 550 terms. Moreover, stemming was applied to the terms within the topics.

(3) Extraction of multiword phrases
The multiword phrases were extracted by using a part-of-speech tagger. This procedure was applied to only the title part of topics for all submitted results. Moreover, only two word phrases were extracted and not applied this procedure to the multiword phrases more than three words. Terms other than the multiword phrase were also extracted as query terms.

(4) Weighting for query terms
The word that composed the multiword phrase, were used as query terms. Moreover, each query term was given a weight that decided by which topic category in or whether multiword phrase.

Step 2: Initial retrieval

In TREC-8, we did the relevance ranking by using the BM25 function of Okapi [2]. The function is shown as follows.

$$\sum_{T \in Q} w^{(1)} \frac{(k_1+1)tf}{K+tf} \frac{(k_3+1)qtf}{k3+qtf} \tag{1}$$

where Q is a query, containing terms T,
$w^{(1)}$ is the Robertson/Sparch Jones weight of T in Q,

$$w^{(1)} = \frac{(r + 0.5)/(R - r + 0.5)}{(n - r + 0.5)/(N - n - R + r + 0.5)} \tag{2}$$

N is the number of documents in the collection,
n is the number of documents containing the term,
R is the number of documents known to be relevant to a specific topic,
r is the number of relevant documents containing the term,
K is $k_1((1 - b) + b\dot{d}l/avdl))$,
tf is the frequency of occurrence of the term within a specific topic,
qtf is the frequency of the term within the topic from Q was derived, and
dl and $avdl$ are the document length and average document length.

First, the query terms, which selected with step 1, was input to the system with query word weight, and the initial retrieval result was obtained. The results were submitted before the query expansion (**nttd8al**,**nttd8am**).

Step 3: Query term expansion

A method similar to LCA [3] was adopted as a query term expansion technique. A passage importance score is given to each passage unit and extended terms are selected in LCA. Since our implementation of LCA is not complete, the top n ranked documents of the initial retrieval were used instead of the passage. The data set used for query expansion is the same Disks 4-5 data set as was used for the retrieval data.

Step 4: The second retrieval processing

This retrieval processing was the same as that in step 1 was used. Query terms that were extracted by query expansion were added to the original query terms. In this case, the weights of the query terms are given to the expanded query terms.

2.2 Result and analysis

We submitted five results. Three are by long query and two are by short query. The same ranking method and parameters were used regardless of the retrieval type (long or short). In the three long query results, **nttd8le** is query expanded, **nttd8l** has no query expansion and **nttd8lx** is a hybrid of **nttd8l** and **nttd8le**. In the two short query results, **nttd8me** is query expanded and **nttd8m** has no query expansion. The parameters used for the TREC-8 experiments were as follows. For the BM25 function, k_1=1.0, b=0.5 and k_3=0. The weight of the extracted multiword phrase was 1.5, each word in the extracted multiword phrase was 0.8 and weight of the other words was 1.0. The retrieval result is shown in Table 1.

Run	AveP	Rel_ret	#Q $\geq$ med	P10	P30
nttd8al No expansion (T+D+N)	0.2781	2973	36	0.4880	0.3800
nttd8ale Expanded (T+D+N)	0.2921	3120	40	0.4940	0.3847
nttd8alx hybrid of nttd8al and nttd8ale	0.2817	2986	38	0.4760	0.3840
nttd8am No expansion (T+D)	0.2649	2937	39	0.4600	0.3667
nttd8ame Expanded (T+D)	0.2721	3028	39	0.4900	0.3747

Table 1: Submitted ad-hoc retrieval runs

used topics	AveP		
	Basic query processing (baseline)	Removal of negative information(from N)	Extraction of multiword phrase (from T,D and N)
T	0.2322	No change	0.2386
D	0.2386	No change	0.2322
T+D	0.2714	No change	0.2714
T+D+N	0.2731	0.2820	0.2820

Table 2: Ad-hoc retrieval runs for various processing types

Ad-hoc basic processing

In the basic retrieval processing, we analyzed the results by used part of the topics. The results show that retrieval becomes better as queries get longer (Table 2).

Removal of non-relevant topic sentences

The result of removing a negative sentence and that of the basic retrieval processing are shown in Table 2. This processing, removing the negative, is done only in the narrative part, so the results do not change in the basic retrieval processing, which does not use the narrative part. This processing results in a 3.3% improvement in average precision.

Phrase identification

Table 2 shows the results for multiword phrase processing of two words in all the topic parts. The result did not change too much, although it was successful in the topics of TREC-7.

3 Question Answering Track

This section describes our method adopted for the question answering track and discusses our results.

3.1 Approach

In our QA track, processing was executed according to the following steps:

(1) Decision of answer forms by question analysis,

(2) Selection of candidate documents and parts for detailed analysis by an initial search of the documents, and

(3) Information extraction to output the final results from the candidate parts.

We mainly used adopted method that depended on surface-text-based.

Decision of answer forms by question analysis

Step 1:

Specifies the part of speech in the question by the POS tagger.

Step 2:

The answer forms of each question were decided according to wh-determiner, wh-pronoun, wh-adverb, etc. The correspondence of the part of speech and the answer form was manually made as a table. The number of answer forms was 24. For instance, when the question is "How long .... ?" and the answer form is assumed to be "TIME". For Question 127 "Which city has the oldest relationship as a sister-city with Los Angeles?", three answer forms are sequentially given. The prime candidate of the answer form is "CITY", the second is "LOCATION" , and the third is "PROPER". Here, when the answer form was not able to be specified, an answer form are given as "UNKNOWN", and the subsequent information extraction is not performed.

Initial search

Step 1: Execution of initial search

Our System is based on the BM25 algorithm. The initial search is the same as the one used in our ad-hoc. The query terms were extracted from the question. After the initial search, the top n ranked documents (D_1, D_2, ..., D_n) were to be answer extraction candidate documents (n was assumed to be an evaluation parameter). Some passage parts where the appearance density of the query term was high was extracted from the top n ranked documents of the initial search. These parts were assumed to include an answer. Moreover, a score was given to the extracted answer candidate part. The method of scoring is as follows.

Step 2: Scoring the answer candidate part

The score $s(P_{ij}, Q_k)$ for query term Q_k and each term position P_{ij} in document D_i is given (P_{ij} is the j-th term position from the top of document D_i). When query term Q_k appears at P_{ij}, the IDF measure of Q_k is given to P_{ij}, that is, $s(P_{ij}, Q_k) = IDF(Q_k)$. The score at the circumference term position was related to the distance with term position appearing query term Q_k. The distance with appearing query term Q_k reduces the score (Figure 1). Two kinds of scoring method were executed.

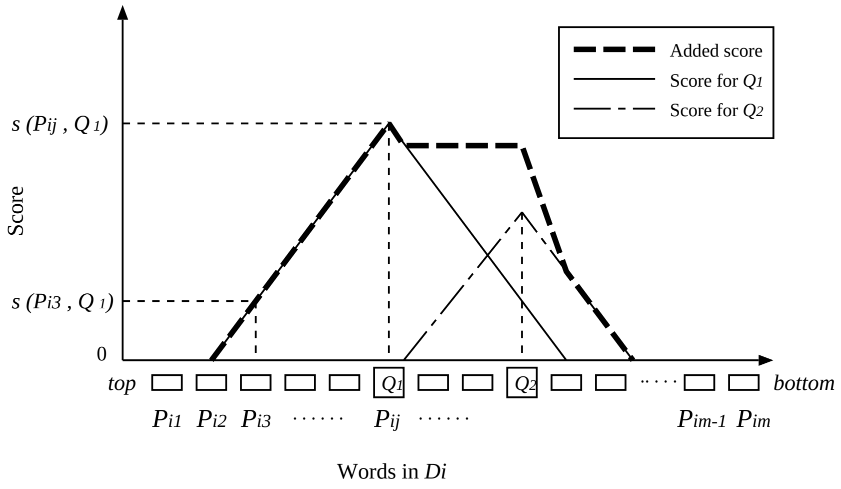

Figure 1: Method of giving term position score.

Methods of giving the term position score

Method X1: The scores are given to the term positions within a fixed number of terms from P_{ij}.

Method X2: The scores are given to the term positions within the range corresponding to the IDF value of the query term Q_k.

Final score $s'(P_{ij})$ at term position P_{ij} was finally assumed to be the sum total of the $s(P_{ij}, Q_k)$ given to each query term. That is, a higher score was given to the part where the appearance density of the query term was high. The consecutive passage parts where the score $s'(P_{ij})$ was more than a set threshold were decided the answer candidate parts. Here, the answer candidate parts are assumed to be C_{ip}. Maximum score $s'(P_{ij})$ in C_p was assumed $sc(C_{ip})$ which the score of the answer candidate parts, that is, $sc(C_{ip}) = max(s'(P_{ij}))$.

Next, the answer text that suits the answer form specified by the query demand is found from the answer candidate parts. When a candidate part C_{ip} included text that matched the answer form, $sc(C_{ip})$ was added as a bonus score.

Information extraction of answer form text

In this approach, information types were given to the each text by the information extraction with the rule-based. Moreover, the name of a country and the city name, etc. was used for information extraction as dictionary information (Table 3), but we did not use corporate name's dictionary and etc., which was not able to be prepared. The number of last-name entries was 88798 but only 1000 general names was used. When an information type that suits the answer type given by the query demand appears, the scores of the candidate part are added. In this case, two kinds of score adding methods were adopted.

Methods of adding score by answer type

Method Y1: The score is added without considering where information that fits the answer type appears.

Dictionary	Number of Entries	Data Source	Examples
Countries	253	ISO 3166 codes	Japan, USA
Cities (airport cities)	1140	www.ufreight.com	Los Angeles, Tokyo
World regions	14	www.yahoo.com	Oceania, Europe, Arctic
US states	50	www.yahoo.com	Maryland, Kentucky, Illinois
Currency names	221	www.bloomberg.com	Euro, European Currency Unit, French Franc
Currency abbreviations	164	www.bloomberg.com	USD, JPY
Dates and times	54	Hand entered	Sunday, Apr, a.m.
Last name	1000	www.census.gov	Smith, Johnson, Williams, Jones

Table 3: Dictionary Features

Runs	Answer length	n (Num. of used initial top ranked documents)	Scoring method X	Scoring method Y
nttd8qs1	50	10	X2	Y2
nttd8qs2	50	10	X2	Y1
nttd8ql1	250	10	X2	Y2
nttd8ql4	250	30	X1(25 words)	Y1

Table 4: Parameters used for the submitted QA runs

Method Y2: The addition degree of the score is proportional to the distance between the term position where the score in answer candidate text C_{ip} is the maximum and the term position of the information that fits the answer type.

Text within a specified number of bytes (50 or 250) is extracted from the peripheral part of text suitable for the answer type, and the final answers are outputted. Our TREC-8's goal was to extract text around the answer as well as the answer.

3.2 Result and analysis

We submitted two results to the category of ”Under 50 bytes” and ”Sentence or under 250 bytes”. **nttd8qs1** and **nttd8qs2** are for the under-50-bytes category, and **nttd8ql1** and **nttd8ql4** are for the under-250-bytes category. The parameters used for each run is shown in Table 4. The result is shown in Table 5. Our result was better than the average of all participants. However, there were a lot of questions in which the correct answer was not able to be included in the top five outputs. Table 6 is the result of classifying by the question to be given the answer form except ”UNKNOWN” and to be given ”UNKNOWN”. Our mean reciprocal rank was much lower in under-50-bytes category when the answer form was not able to be specified. In contrast, the rank was high when it was possible to specify the answer form. However, this difference in rank did not appear in the under-250-bytes category. Therefore, we think a retrieval technique like ad-hoc is effective for under-250-bytes category in our evaluation, and

Run	Mean_	Num. of answers found at rank X						#Q	#Q
	Reciprocal_rank	1st	2nd	3rd	4th	5th	Not found	Best	≥ Med
nttd8qs1	0.273	40	17	9	6	5	121	54	168
nttd8qs2	0.259	37	13	14	7	7	120	49	160
nttd8ql1	0.439	65	32	9	7	6	79	75	183
nttd8ql4	0.371	54	25	10	8	8	93	62	182

Table 5: Submitted QA runs

Classification	#Question	Mean_reciprocal_rank			
		50 bytes		250 bytes	
		nttdqs1	average	**nttdql1**	average
UNKNOWN	48	0.077	0.229	0.520	0.338
Expect UNKNOWN	150	0.335	0.209	0.413	0.330
All	198	0.273	0.214	0.439	0.332

Table 6: Classified analysis by answer form

that the question analysis is important for the stricter under-50-bytes category.

4 Summary

We described our system approach and discussed the results for ad-hoc and question answering in TREC-8. Especially, our results in question answering track were a little fine. Our implementation is not complete with respect to the answer form specific processing and the information extraction processing, so there are a lot of points that should be improved. Moreover, we will examine linguistic techniques for question answering in the future.

References

[1] H. Nakajima, T. Takaki, T. Hirao and A. Kitauchi. NTT DATA at TREC-7: system approach for ad-hoc and filtering. In E. M. Voorhees and D. K. Harman editors, *Proceedings of the 7th Text Retrieval Conference (TREC-7)*, pages 481-489. NIST Special Publication 500-242, 1999.

[2] S.E. Robertson, S. Walker and M. Beaulieu. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive. In E. M. Voorhees and D. K. Harman editors, *Proceedings of the 7th Text Retrieval Conference (TREC-7)*, pages 253-264. NIST Special Publication 500-242, 1999.

[3] J. Xu and W. B.Croft. Query expansion using local and global document analysis. In *Proceedings of the 19th annual international ACM SIGIR conference on research and development in information retrieval*, pages.4-11, Zurich, 1996.