Information Retrieval System for TREC3

Kenji SATOH, Akitoshi OKUMURA, Kiyoshi YAMABANA
Information Technology Research Labs.
NEC Corp.

1. Introduction

This is our first participation with TREC. Our team researches natural language processing, and we have developed English-Japanese and Japanese-English machine translation system. (The code name of machine translation system is VENUS.) We are now researching a new natural language processing environment, including information retrieval and text understanding. (The environment name is VIRTUE: VENUS for Information Retrieval and Text Understanding.)

Last year, our team participated with MUC5, and we got promising results[1]. This year,

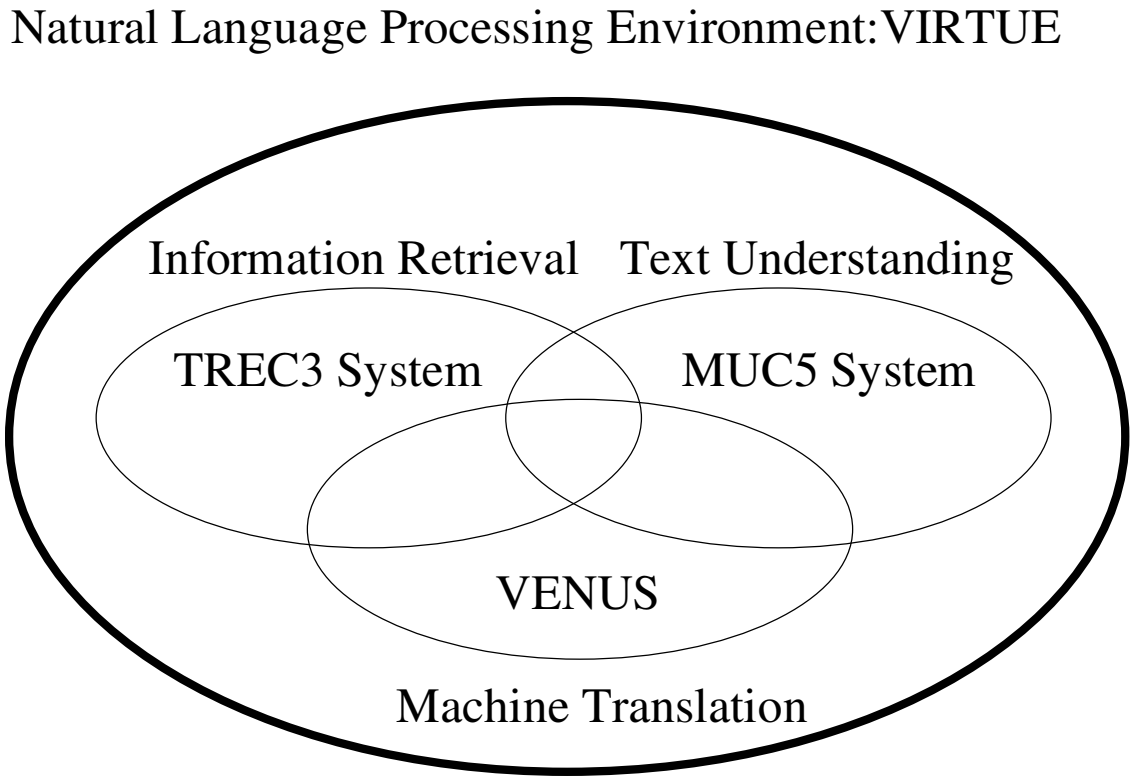

Figure 1. Our Goal

our team has participated with the TREC3 for researching and developing and information retrieval system. Figure 1 shows our goal of researching natural language environment.

Because of our past development of machine translation systems, we have on hand translation system dictionaries (140,000-word), a conjugation table of English words, and morphological and syntax parsers. The purpose of participating with TREC3 focuses on the feasibility of using this data and applying English language analysis technology. We developed a system very quickly using them.

2. System Modules

The overall outline of the system concerns firstly, the generation of an index from NIST document collection as inverted files. We generated inverted files for each CD-ROMs. Query term generation from topic with routing task phase is generated manually, and with ad-hoc task phase is processed automatically.

Query weights are added by weight calculation modules with referring inverted files. Finally, matching between query and document as well as ranking documents used the same program with both tasks.

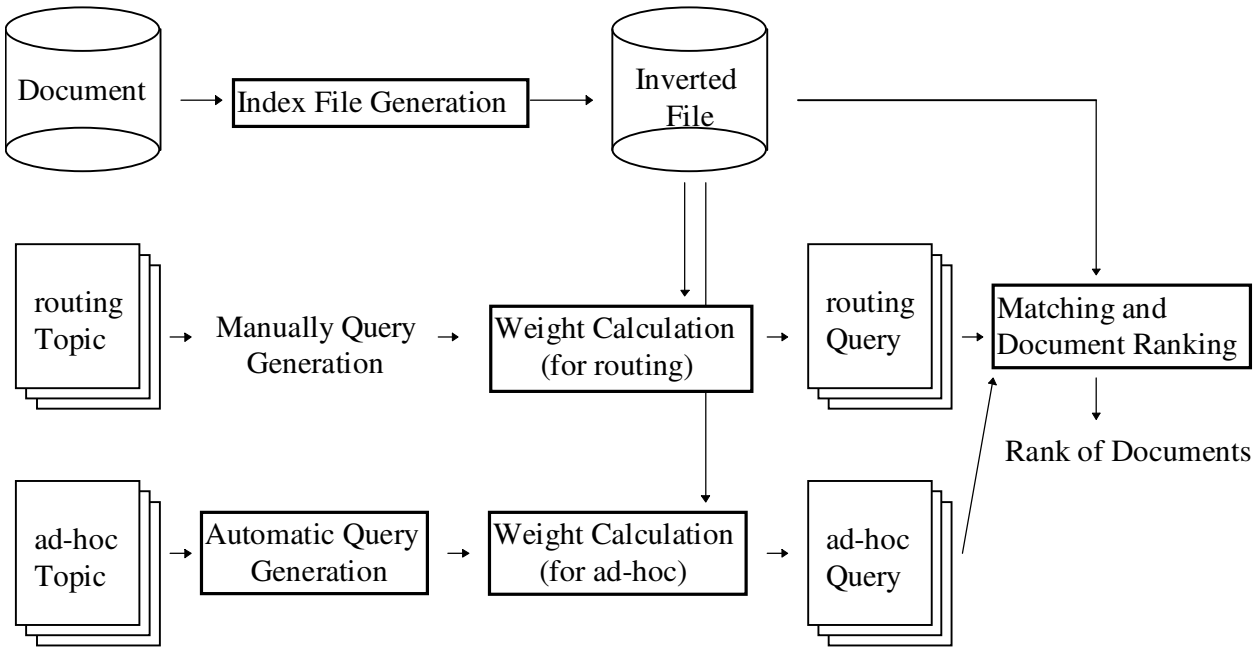

Figure 2. System modules

I will explain this system in order; Index File Generation, Query Generation, and Matching and Document Ranking. Figure 2. shows modules of our system.

3. Index File Generation

The index generation method from a document collection will be explained in this section. The index is created by an inverted file in which a single word (not including spaces) as a key. Words in the conjugated form and/or the plural form may be used as a key.

The reason that general stemming method dose not used is this method can not handle irregular conjugations, our system has an English-language conjugation table, and it is thought that with this table, the conjugated form sufficiently handled during matching phase.

The word recognition of each document is carried out by matching with a 140,000-word English dictionary, the 430 stop words are dropped. Furthermore, symbolic strings (repetition of upper case and hyphens, etc.) are recognized as a single word.

After the word recognition phase is carried out, the DOCNO of documents in which the word has appeared, as well as the frequency and position of the word in each document are saved. Although word weight is not recorded at this point in time, word weight calculates for the words appearing in the query when matching with document.

During indexing generation, listing the number of words in the document is simultaneously generated as a separate file. This information is used in normalizing the word weight during matching.

The size and generating time (with Sun SPARCstation 10) of the inverted files are:

Disk 1 --- 880 MBytes, 93 Hours
Disk 2 --- 610 MBytes, 53 Hours
Disk 3 --- 660 MBytes, 67 Hours

4. Query Generation

A sample of query is shown in Figure 3. The query format is a compound format consisting of the boolean operator and word weight. The one key of the query does not necessarily have to be a singular word; if it is, the word sequence in the text is checked.

As boolean operator, AND condition as simple listing of keys in query, OR condition (enclosed by brackets) in which the appearance of multiple keys regarded as a single key. A key

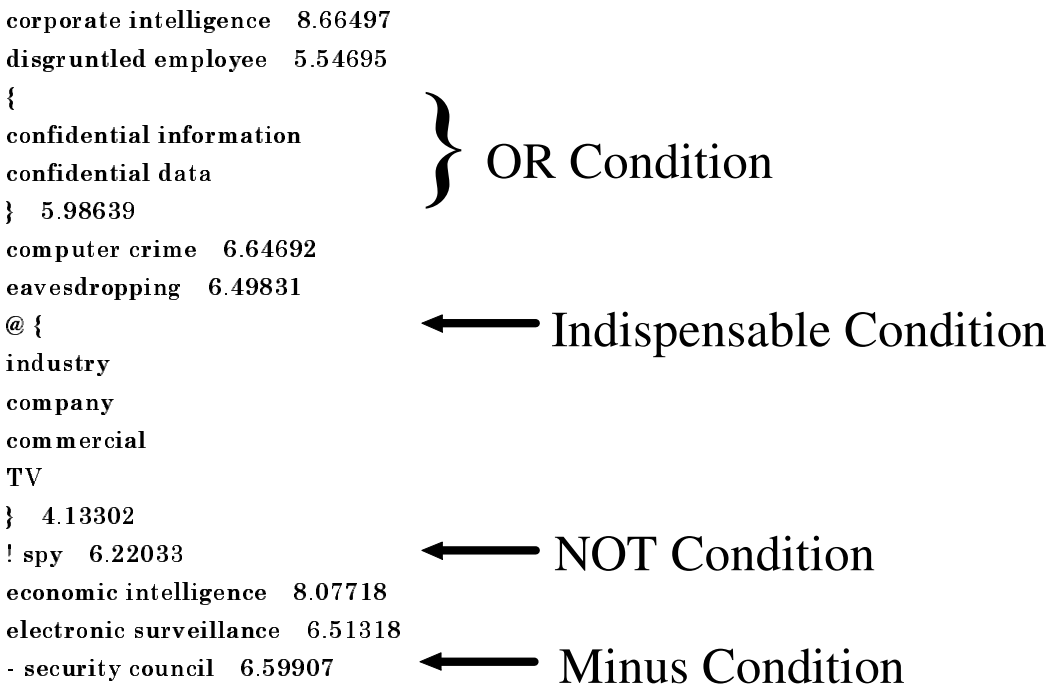

Figure 3. Sample of query

preceded by an exclamation mark indicates a NOT condition, and a key preceded by an @-mark indicates the indispensable condition. A key preceded by a minus-sign makes the word weight value take on a negative.

The query generation with routing task phase is executed manually. At the first stage, initial query matched for training documents. That result is compared with the relevance judgment, upon which the query is modified.

The query generation with ad-hoc task phase processed automatically. For all the files in topic, an English-language analysis is executed from which noun phrases are extracted. In this phase, if there are expressions indicating negation for a noun phrase, the phrase is inserted in a query with a minus operator.

Word weight calculation does not occur in either of the query generation processes. Word weight is assigned by a program after completion of the query keys. A combination method using the operator and word weight will be explained as follows.

5. Matching and Document Ranking

The way in which the query operator and word weight combinations are carried out during matching is as follows. Keys connected by an OR condition calculate the frequency and list up the documents in which a word appears as a single key. A document with the NOT condition key in a matched document on another key is excluded from the ranking calculation. Conversely, a document without the indispensable key conditions in a matched document on another key is excluded from the ranking calculation.

Next, when an inverted file is referenced by a key in the query, the key returns once to its original form by using the conjugation table and it is unfolded into the all the possible conjugated form. If a key 'wrote' exists in the query, it returns to 'write' and it is unfolded into 'write', 'wrote', 'written' and 'writing.'

Also, if the key in the query is consist of multiple words, the unfolding phase and index search are enabled for each word in the key. If a key 'wrote file' exists in the query, it is unfolded into 'write file', 'wrote file', 'written file', 'writing file', 'write files', 'wrote files', 'written files' and 'writing files.' After each word is searched, the sequence check is executed according to their positions in the document which saved in the index file.

The weight calculation during the routing task phase is uses the modified form of traditional relevance weight model[2]:

$$w_i = \log \frac{\left(r_i + 0.5\right)/\left(R - r_i + 0.5\right)}{\left(n_i - r_i + 0.5\right)/\left(N - n_i - R + r_i + 0.5\right)}$$

where

N : the number of documents
n_i : the number of documents containing the term i
R : the number of relevant documents
r_i : the number of relevant documents containing the term i

Also, the weight calculation during the ad-hoc task employs the modified form of the standard deviation of word appearance frequency:

$$w_i = \log\left(\frac{N}{n_i} \sqrt{\frac{1}{L} \sum_{j=1}^{L} \left(\frac{m_{ij}}{M_j} - \frac{n_i}{N} \right)^2} \right)$$

where

L : the number of documents
N : the total number of terms in all documents
n_i : the number of term i in all documents
M_j : the total number of terms in document j
m_{ij} : the number of term i in document j

For ranking, the internal product is taken between the weight of words in a query and word frequency in a document, and the log factor is imported by the results of multiplying each internal product [3].

$$S_i = \frac{1}{\log D_i} \sum_{j=1}^{N} \log\left(\left(d_{ij} + 1 \right) * w_i \right)$$

where

S_i : the score of document i
D_i : the total number of terms in document i
N : the total number of terms
d_{ij} : the number of term j in document i
w_j : the weight of term j

Total time of weight calculation, matching with document and document ranking is about 20 Minutes (average per query.)

6. Discussion

The results of our system are shown in Table 1.

Table 1. Results of our systems

	Routing (manually)	Ad-hoc (automatically)
Average precision	0.2717	0.0624
R-precision	0.3204	0.1170

Because the calculation of the word weight uses the standard deviation of word appearance frequency during ad-hoc task phase, the deviation value is often very large for particularly long documents. This results in such documents being placed in a high ranking. Another problem is that noun phrases which automatically extracted from topic, sometimes retrieves insufficient documents. Table 2 shows the summary of our system.

Concerning points of improvement from now, we are thinking of another weight calculation method rather than the use of deviation. Also, if an adjacent operator were introduced, we believe that even if a topic changes within a particularly long document, more precise matching can be carried out. For the retrieval of more sufficient documents, other sources, such as thesaurus or Word Net, should be used so that the number of keywords is increased.

Table 2. Summary of our system

	routing	ad-hoc
Index File Generation	Conjugated form used as a key. Position of word is saved.	(same as routing)
Query Format	Boolean and weight compound format	(same as routing)
Query Generation		Noun phrase extraction
Term Weighting	(traditional relevance weight model)	Standard deviation
Matching	Keys are unfolded into all conjugated form. Multiple words key is checked the sequence.	(same as routing)
Ranking	(internal product)	(internal product)

7. Conclusion

Because this is the first such presentation of TREC3, we started from estimation of the amount of time and memory necessary for the creation of an inverted file, and unfortunately, did not have enough time to repeatedly matching nor modify of the weight.

But using our original dictionary and conjugation table, as well as adopting natural language processing technology for developing this system, we got promising results when applying these technologies to information retrieval systems, while at the same time, we decreased developing and maintenance costs. (We have developed this system in a month with a 3-person team.)

References

[1] Muraki, K. Doi, S. and Ando, S. “NEC: Description of the VENIEX System as Used for MUC-5”, proceedings of Fifth Message Understanding Conference (MUC-5), pp.147-159, Aug. 1993

[2] Robertson, S.E. and Sparck Jones, K. “Relevance weighting of search terms”, Journal of the American Society for Information Science, 27, pp. 129-146, 1976

[3] William B. Frakes and Ricardo Baeza-Yates edit, “Information Retrieval - Data Structure & Algorithms”