NTCIR-2 Experiments at Matsushita: Monolingual and Cross-Lingual IR Tasks

SATO, Mitsuhiro
msato@trl.mei.co.jp

NOGUCHI, Naohiko
noguchi@trl.mei.co.jp

Multimedia Systems Research Laboratory, Matsushita Electric Industrial Co., Ltd.
4-5-15, Higashi-Shinagawa, Shinagawa-ku, Tokyo 140-8632 JAPAN
+81-3-5460-2744

ABSTRACT

In this paper, we describe our results for NTCIR-2 monolingual and cross-lingual IR tasks utilizing *MEISTER*, a group of software libraries to construct high performance full-text search applications. The results show that the coordination level scoring (*CLS*) which we proposed in NTCIR-1 is again effective for both short queries and long queries in Japanese monolingual task, and show that some modifications of *CLS* works slightly better than the original. In Japanese-English cross-lingual task, we show that utilizing bilingual dictionary with corpus-based term translation slightly improves performance of our approach that we proposed in NTCIR-1.

Keywords

full-text search system, information retrieval model, term extraction, similarity measure, Boolean operator, coordination level scoring, query term translation, cross-language information retrieval

1. INTRODUCTION

For NTCIR-2, we participated in J-J task, J-E task, and J-JE task utilizing again our *MEISTER* system. *MEISTER* is a group of software based on the new type of indexing method called "*maximal word indexing* method" which is especially suitable for text of non-segmented languages like Japanese. It comprises several software modules, for example, an indexing module, a searching and ranking module, a term extraction module, a term similarity calculation module, etc. Its basic functions and features were already described elsewhere in our papers [7][9][12], so we will not go into the details of *MEISTER* system here.

For the experiments on NTCIR-2 tasks, we constructed a full-text search system for all the documents from NTCIR-1 and NTCIR-2 collections, combining several modules of *MEISTER*. We built full-text search systems for Japanese text and English text for J-J task and J-E/J-JE task respectively. The term extraction module was used to extract query terms from each Japanese topic. The searching and ranking module was used to rank the documents applying a set of query terms for each query. We used an EDR-based dictionary as a sole source of linguistic information. We did not use any linguistic processing such as morphological analysis or syntactic analysis.

As for J-J task, we concentrated on evaluating the effectiveness of *CLS* (Coordination Level Scoring). Through our experiments, we tried to see how *CLS* performs for the new set of documents, and whether or not some modifications of *CLS*, which we will propose later in this paper, perform well.

As for J-E task, we experimented the same approach as we took in NTCIR-1. We again tried to see how our corpus-based approach performs for the new set of documents. Then we examined how effective using bilingual dictionary with our corpus-based term translation.

In section 2, we describe our experiments on J-J task. In section 3, we describe our experiments on J-E and J-JE tasks. Section 4 summarizes the paper.

2. EXPERIMENTS ON J-J TASK

2.1 A Brief Description of *CLS*

The similarity measure we called "*CLS*" in this paper can be represented in the formula (1).

$$(1)\quad Sim(d_j, q) = C_1 \sum_{q_i \in Q} w_i \cdot |tf_{ij}| \cdot idf_i + C_2 M_j$$

Here, q is a query, d_j is a document, Q is a set of query terms included in the query q, tf_{ij} is the occurrence frequency of the term q_i in the document d_j ($|tf_{ij}|$ means some normalization is done on tf_{ij}), idf_i is the inverse document frequency of q_i computed by the formula (2), w_i is a given weight of q_i, and C_1, C_2 are constants.

$$(2)\quad idf_i = \log\left(\frac{N}{df_i}\right) + 1$$

(N is the number of the whole documents, df_i is the document frequency of q_i)

The first component of the formula (1) can be thought as the inner product of a document vector and a query vector. The second component M_j is the number of co-occurring terms among Q in the document d_j, which represents a kind of coordination level information. We added M_j here in the formula (1) because of the facts that coordination level information is useful especially for short queries [2][11][14], that most of real user queries are composed of a few words, and that users tend to read just a few top-ranked documents. The default setting of *MEISTER*, which takes these facts seriously, makes C_2 considerably bigger than C_1. That is, C_2 has to satisfy the condition (3).

$$(3)\quad C_2 \geq \max_{d_j \in D}\{C_1 \sum_{q_i \in Q} w_i \cdot |tf_{ij}| \cdot idf_i\}$$

Here, D is the set of the whole documents. The condition (3) means that the result set of documents are ranked by coordination level information first, producing several layers of score, then the documents in each layer are ranked by a kind of *tfidf* measure. We will call this default setting "Coordination Level Scoring", *CLS* for short, and call the setting where $C_2 = 0$ "*naïve tfidf*" for the ease of following discussion.

2.2 Modifying *CLS*

It has been said that the coordination level information is useful for short queries, but is not so useful for long queries [6][14] in general. Our results of NTCIR-1 experiments implied that *CLS* was in fact effective not only for short queries but also for long queries. So our first objective of NTCIR-2 experiments is to see how *CLS* performs for the new document set.

Our second objective is to see how some modifications of *CLS* works for the NTCIR-2 documents. We proposed in the previous report [13] several directions to modify *CLS*. They are:

a) To introduce some weights on coordination level information.

b) To introduce some normalization factor into coordination level information based on the query length.

c) To restrict the domain in which coordination level information is computed.

Our motivation for heavy use of coordination level information in *CLS* is that coordination level information is fairly intuitive and transparent to users when they look at the ranking of the retrieved documents. In practical text retrieval situation where efficient interactions between users and systems are very important, this transparency might be a great help. If we take the approach a) and b), we will go into the world of parameter tuning, therefore the transparency of *CLS* will be blurred. If we take the approach c), the combinatorial nature of *CLS* still remains so that users can understand the resulted ranking more easily. So we take the approach c) for NTCIR-2 experiments.

Considering the reason why *CLS* is said not to be effective for long queries, the possible ways of restricting the domain of *CLS* would be:

c-1) To select only meaningful (pairs of) terms from all the query terms to compute *CLS*.

c-2) To select only important (or useful) parts of the documents to compute *CLS*.

As the number of query terms increases, there would be many possible combinations of unrelated query terms. *CLS* blindly takes those combinations too seriously. The approach c-1) is the way to avoid this situation. But it is hard to select only meaningful (pairs of) terms for *CLS* automatically, so we simply used the fields of each topic to select query terms applying to *CLS*. For example, we only used terms from Title and Concept to compute the second component of the formula (1), C_2M_j, although we used all the terms to compute the first component of the formula (1).

As the length of a document gets longer, there would be many co-occurrences of query terms. Because *CLS* takes every co-occurrence of query terms in a document as equally important, it is possible that term co-occurrences in relatively unimportant portion of a document blur the effects of *CLS*. The approach c-2) is the way to avoid this situation. Again, it is hard to decide automatically which part is important and which part is not, so we simply used the fields of each document, e.g., Title, Keyword and Abstract fields to decide such portions in which CLS is computed

2.3 Query Construction

We first extracted query terms from Title, Description and Narrative field using *MEISTER*'s term extraction facility. *MEISTER* can extract non-dictionary terms using character type information, so both dictionary terms and non-dictionary terms were extracted. In the term extraction process, all the dictionary terms were weighted. We used these weights as query term weights. As for terms extracted from Narrative field, we took top 20 terms by their weights, but non-dictionary terms first. As for terms in the Concept field, we simply used these terms as query terms.

Finally, all the weighted query terms are ***OR***-ed to construct weighted structured query, which would be an input to the searching and ranking processes of *MEISTER*. We did not use any other Boolean operator, ***AND*** or ***NOT***, because these operators are too restrictive in fixing the result set to use appropriately in a fully automatic query construction process. This time, we did not use other *MEISTER*'s special operators like ***ADD*** and ***Synonym*** operator in query construction process.

2.4 Results and Analysis

2.4.1 The effectiveness of CLS

We first conducted an experiment to see how *CLS* performs for the new document set. **Table 2-1** and **Table 2-2** show the results. Here, the meanings of the symbols indicated at the top row of the table are as follow.

D : Query terms are taken only from **D**escription field.

DC : Query terms are taken from **D**escription and **C**oncept field.

DTCN: Query terms are taken from **D**escription, **T**itle, **C**oncept and **N**arrative field. The maximum number of the terms taken from Narrative field is 20.

CLS : Documents are ranked by *CLS*.

(As for the runs which are not indicated by **CLS**, documents are ranked by *naïve tfidf*.)

Table 2-1

	D	**D-CLS**	**DC**	**DC-CLS**
Ave. Prec	0.1252	0.2771	0.1647	0.3270
R-Prec	0.1729	0.2538	0.2039	0.3437
Recall	0.4480	0.4747	0.5486	0.6072

Table 2-2

	DTCN	DTCN-CLS
Ave. Prec	0.1994	0.3312
R-Prec	0.2413	0.3428
Recall	0.5747	0.6358

The results of short query runs are shown in **Table 2-1** where *CLS* drastically improved the effectiveness of each run (**D** and **DC**). **Table 2-2** shows the results for long query runs. Here as well, we can see a drastic improvement in precision and recall by *CLS*. Then we can conclude that *CLS* is consistent in improving effectiveness not only for short query runs but also for long query runs. This is the same conclusion as we got from the experiments in NTCIR-1. In **figure 2-1**, 11 point precision-recall graphs for each run are shown.

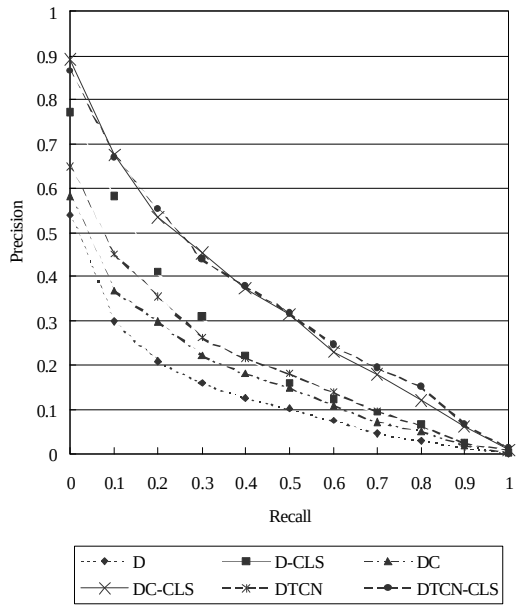

Figure 2-1

Basically, the longer the query is, the better the effectiveness is, for *naïve tfidf*. This is true as well for *CLS*. But comparing **DC-CLS** and **DTCN-CLS**, you can see the difference among them is very slight. We can see that the effects of *CLS* are almost saturated. This fact leads us to make some modification of *CLS* so that we can improve a bit more in the long query runs, that will be the theme of the next section

2.4.2 The results on the modification of the CLS

We tried two approaches described in 2.2 to modify *CLS*. Our first experiment is on restricting the domain of query field. We chose the long query run (**DTCN**) as a baseline and varied the fields from which query terms for computing *CLS* are taken. **Table 2-3** shows the results. The first symbol in the top row of the table represents the fields from which query terms are taken, the second symbol represents the fields from which query terms for computing *CLS* are taken. For example, **DTCN-DT** means that query terms are taken from **D**escription, **T**itle, **C**oncept and **N**arrative field and query terms for computing *CLS* are taken from **D**escription and **T**itle field.

Table 2-3

	DTCN-DTCN	DTCN-DTC	DTCN-DT	DTCN-TC
Ave. Prec	0.3312	0.3305	0.2933	0.3371
R-Prec	0.3428	0.3461	0.3185	0.3514
Recall	0.6358	0.6112	0.5444	0.6238

The precision of **DTCN-TC** is slightly better than **DTCN-DTCN** (the baseline), but other runs are below the baseline. Looking at the content of each field of a topic, the similar expressions are repeatedly used in Title, Description, Narrative and Concept field. This means that if we apply *CLS* for all these fields, there would be too many add-ons in the document score as the coordination level score. Restricting the application domain only to Title and Concept fields might be one way of eliminating this effect. But we are not convinced that such analysis is true merely on this really slight improvement.

Then we conducted the next experiment. This time we chose **DTCN-TC** as a baseline and varied the parts of documents in which *CLS* is computed. **Table 2-4** shows the results. The last symbol in the top row of the table represents the parts of documents in which *CLS* is computed.

TA : *CLS* is computed in **T**itle and **A**bstract fields.

AK : *CLS* is computed in **A**bstract and **K**eyword fields.

A : *CLS* is computed only in **A**bstract field

.

Table 2-4

	DTCN-TC-TA	DTCN-DT-AK	DTCN-TC-A
Ave. Prec	0.3274	0.3295	0.3135
R-Prec	0.3458	0.3487	0.3330
Recall	0.6171	0.6140	0.6100

These runs are slightly worse than the baseline, so this modification did not work well. This means that all Title, Keyword, and Abstract fields are equally important to be used for computing *CLS*. But our prediction is that if we conduct an experiment on other documents like patent documents or full-text of technical papers, the situation will be different. In such cases, we believe that we can get better performances by deliberately selecting the domain of computing *CLS*.

2.4.3 The formal runs

We submitted four formal runs for J-J task. Those are **D-CLS** for description only run (short query run), **DC-CLS** for another short query run (short with concept), **DTCN-CLS**, for long query run, and **DTCN-DTC**, which is a modification of *CLS*. All the runs are described in the previous sections.

3. EXPERIMENTS ON J-E and J-JE TASK

3.1 Corpus-based term translation

For NTCIR-1 CLIR task, we experimented a term translation method using a parallel corpus [13]. For NTCIR-2, we employed the same approach again to check the effectiveness for different set of documents.

Our corpus-based method utilizes the term similarity calculation module of *MEISTER*. The similarity between terms is calculated based on occurrence patterns of each term at a document level. *MEISTER* uses *term vector* to represent the occurrence pattern of each term, and calculates similarities between terms using the vector representation. A *term document matrix*, which comprises the set of all term vectors occurring in the whole documents, is constructed in advance for the better performance of the system.

Using a Japanese-English parallel corpus with document-level alignment, we can translate Japanese terms into English terms. The essence of our approach is:

(a) Construct a term-document matrix from English (target language) documents of the parallel corpus.

(b) Construct an index from Japanese (source language) doc uments of the parallel corpus.

(c) Construct term vectors using the Japanese index each of which corresponds to a Japanese query term.

(d) Calculate similarities between Japanese term vector and each term vector in the English term-document matrix. Then, select *n* English terms that have the highest similarities.

In step (c), the system uses string search facility of *MEISTER*. So, for any arbitrary string, we can get English terms as far as the Japanese string occurs in the corpus.

Figure 3-1 shows the diagram of our corpus-based approach.

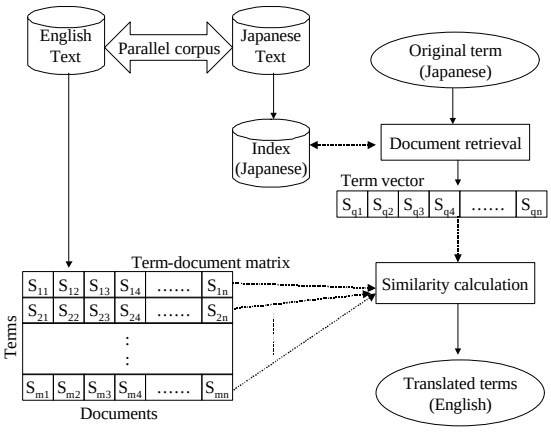

Figure 3-1 Term translation using similarity calculation

To build a parallel corpus, we extracted documents that have both Japanese and English abstracts from NTCIR-1 documents. Then, we made a Japanese index and an English term-document matrix from this corpus. The number of documents in the parallel corpus is 181,489.

3.2 Using a bilingual dictionary

Our corpus-based term translation depends on the quality and quantity of the parallel corpus, and we cannot translate terms that do not occur in the corpus. Using a bilingual dictionary may solve this problem. This time, we tried to construct a bilingual dictionary and use it with our corpus-based term translation method.

First, we constructed the bilingual dictionary simply merging the EDICT [4], the EDR Japanese-English bilingual dictionary of technical terms [5], and some other bilingual resources we already have. At this time, about 91% of Japanese words have less than 3 translation (English) words. We thought these words might be suitable as entries of the bilingual dictionary.

Remaining 9% of words should be disambiguated. Using parallel corpora for disambiguation of translation equivalent is major approach among past CLIR researches [1][3]. We took similar approach. We again utilized the term similarity calculation facility of *MEISTER*. The disambiguation process is:

- Calculate the similarity between a Japanese word and its translation (English word) based on the parallel corpus. We used the same corpus described in 3.1.
- Exclude the translation if its similarity is less than pre-de fined threshold.

Table 3-1 shows the size of our bilingual dictionary. According to our preliminary experiments, using the disambiguated dictionary achieved slightly better performance than using the simply merged dictionary.

Table 3-1

	Words	%
Only 1 translation	129684	78.5
2 translations	27403	16.6
3 translations	8022	4.9
Total	165109	100.0

3.3 Query Construction

Our system constructs English queries automatically by following steps.

(a) Extract query terms from a Japanese description of query request using the same method described in 2.3.

(b) Translate Japanese terms to English terms using the method described in 3.1. Two English terms are selected par a Japanese term.

(c) Look up the bilingual dictionary constructed in 3.2 and obtain the translated English terms. Then, exclude terms that are already selected in step (b).

(d) Terms selected in step (b) and (c) are ***OR***-ed to construct an English query.

We examined following three approaches:

(1) Using only corpus-based term translation. In this case, step (c) was omitted. This is the same approach as we took in NTCIR-1.

(2) Using only bilingual dictionary. In this case, step (b) was omitted.

(3) Using both corpus-based term translation and bilingual dictionary. In this case, above all steps were done.

3.4 Results and Analysis

3.4.1 Effectiveness using the bilingual dictionary

We experimented short and long query runs with three different term translation methods described in 3.3. **Table 3-2** and **Table 3-3** show the results. Here, the meanings of the symbols indicated at the top row of the table are as follow.

D : Japanese terms are taken only from **D**escription field.

DTCN: Japanese terms are taken from **D**escription, **T**itle, **C**oncept and **N**arrative field. The maximum number of the terms taken from Narrative field is 20.

Co : **Co**rpus-based term translation

Di : **Di**ctionary-based term translation

CD : Using both **C**orpus-based and **D**ictionary-based method.

In document ranking, we employed Okapi's BM25 weighting schema [10]. We did not use *CLS* for J-E task since following situations might increase possible combinations of unrelated query terms.

- Mistranslated terms or ambiguous terms may appear.
- Queries in J-E task tend to longer than queries in J-J task. In our experiments, the average of term number was 4.7 in D-CLS of J-J, while 10.7 in D-CD of J-E.

Table 3-2

	D-Co	D-Di	D-CD
Ave. Prec.	0.1290	0.1098	0.1494
R-Prec.	0.1449	0.1388	0.1680
Recall	0.5477	0.3801	0.5727

Table 3-3

	DTNC-Co	DTNC-Di	DTNC-CD
Ave. Prec.	0.1610	0.1160	0.1683
R-Prec.	0.1721	0.1366	0.1846
Recall	0.6626	0.4322	0.6790

Corpus-based term translation (**D-Co** and **DTNC-Co**) achieved higher performance than dictionary-based method in both short and long query runs. Moreover, using dictionary with corpus-based method (**D-CD** and **DTNC-CD**) improved in both precision and recall as we expected. However, the effectiveness in **DTNC-CD** seems to be lower than in **D-CD**. We guess this comes from the lack of technical terms, especially appearing in <NARRATIVE> or <CONCEPT> fields, in our bilingual dictionary. The small difference of performance between **D-Di** and **DTNC-Di** supports this assumption.

3.4.2 The formal runs

We submitted four formal runs for J-E task. Those are **D-Co** and **D-CD** for description only run (short query run), and **DTCN-Co** and **DTCN-CD** for long query run. All the runs are described in the previous sections.

We compared our short query runs with the monolingual (E-E) run. Our monolingual run (**D-EE**) employed very simple query construction: extract words from the <DESCRIPTION> field except stop words, stem words, and concatenate all words by ***OR*** operator. Although we did not submit this run for E-E task, the result was almost same level of median among E-E formal runs (short query) in NTCIR-2.

As shown in **Table 3-4** and **Figure 3-2**, our CLIR achieved 57-66% in precision against the monolingual retrieval.

Table 3-4

	D-EE	D-Co/D-EE	D-CD/D-EE
Ave. Prec.	0.2247	57%	66%
R-Prec.	0.2557	57%	66%
Recall	0.5193	105%	110%

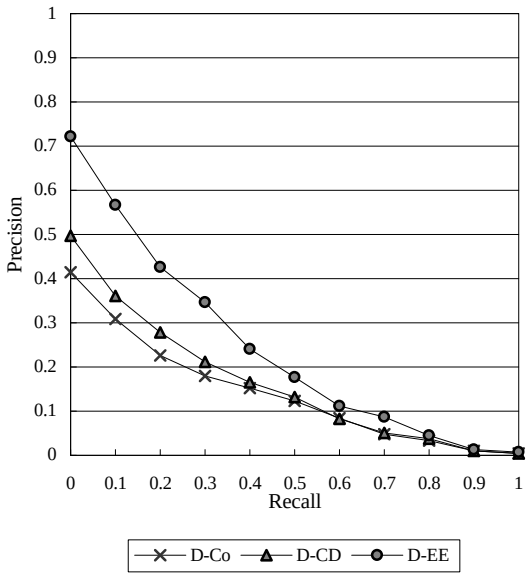

Figure 3-2

We think this result is insufficient. Increasing vocabularies in the bilingual dictionary may improve the accuracy. However, the dictionary including over 165,000 words is not so small. To construct larger bilingual dictionary may be costly.

Constructing larger parallel corpus may be another way of improvement. An advantage of our corpus-based approach is that the parallel corpus utilized in our method needs only document-level alignments. Using similar strategy reported in [8], we may construct large parallel corpus from Web resources.

We also submitted two formal runs for J-JE task. We purely combined our result of J-J run and J-E run, according

to the proportion of Japanese/English documents in the target document set (about 2:1 in NTCIR-2).

D-J-JE : Combined **D-CLS** of J-J task and **D-CD** of J-E task.

DTNC-J-JE : Combined **DTCN-CLS** of J-J task and **DTCN-CD** of J-E task.

Table 3-5 shows the results.

Table 3-5

	D-J-JE	**DTNC-J-JE**
Ave. Prec.	0.1814	0.2501
R-Prec.	0.2335	0.2965
Recall	0.4368	0.5623

4. Conclusion

As for J-J task, we concentrated on evaluating the effectiveness of *CLS* (Coordination Level Scoring). Our results show that *CLS* is in fact effective not only for short query runs but also for long query runs, as we concluded in our previous report of NTCIR-1. Moreover, we tried two modifications of *CLS* either of which is based on the idea that the coordination level information should be evaluated locally. The results show that one of the modifications might work well, but further investigation is needed.

As for J-E task, we evaluated the effectiveness of using a bilingual dictionary in our corpus-based term translation. The use of the bilingual dictionary slightly improved the performance of our corpus-based method, but we couldn't see a drastic improvement. Constructing large-scale dictionary may be costly, so collecting more parallel documents will be effective for our method.

Also, our method currently ignores query contexts, i.e. translated terms are selected without considering entire query. However, we may obtain more adequate translated terms considering query contexts. This is another work we have to do in the future.

5. REFERENCES

[1] Ballesteros, L. and Croft, W.B. Resolving Ambiguity for Cross-Language Retrieval, Proc. of SIGIR98 (1998), 64-71.

[2] Cormack, G.V., et al. Passage-Based Refinement (MultiText Experiments for TREC-6). 6th Text REtrieval Conference (1997), 303-320.

[3] Davis, M.W. and Ogden, W.C. QUILT: Implementing a Large-Scale Cross-Language Text Retrieval System. Proc. of SIGIR97 (1997), 92-97.

[4] EDICT. http://www.csse.monash.edu.au/~jwb/edict.html.

[5] EDR dictionary. http://www.iijnet.or.jp/edr/.

[6] Franz, M. et al. Ad hoc and Multilingual Information Retrieval at IBM. 7th Text REtrieval Conference (1998), 157-168.

[7] Kanno, Y. et al. Maximal-Extension Indexing method for Smart Text Retrieval *MEISTER*: Dictionary/Index Sybsystem (in Japanese). Proc. 55th Annual Convention IPS Japan (1997), 3N-02.

[8] Nie, J-Y. Cross-Language Information Retrieval based on Parallel Texts and Automatic Mining of Parallel Texts from the Web, Proc. of SIGIR99 (1999), 74-81.

[9] Noguchi, N., Kanno, Y., Kurachi, K., and Inaba, M. New Indices for Japanese Text: A New Word-based Index of Non-segmented Text for Fast Full-text-search Systems. Transactions of IPS Japan, Vol.39, No.4 (1998), 1098-1107.

[10] Robertson, S. E., Walker, S., and Beaulieu, M. Okapi at TREC-7: automatic ad hoc, filtering, VLC and interactive. 7th Text REtrieval Conference (1998), 253-264.

[11] Rose, D. E., and Stevens, C. V-Twin: A Lightweight Engine for Interactive Use. 5th Text REtrieval Conference (1996), 279-290.

[12] Sato, M. et al. Maximal-Extension Indexing method for Smart Text Retrieval *MEISTER*: Related Keywords Extraction (in Japanese). Proc. 55th Annual Convention IPS Japan (1997), 3N-04.

[13] Sato, M., Ito, H. and Noguchi, N. NTCIR Experiments at Matsushita: Ad-hoc and CLIR Task. Proceedings of the First NTCIR Workshop on Research in Japanese Text Retrieval and Term Recognition (1999).

[14] Wilkinson, R., Zobel, J., and Sacks-Davis, R. Similarity measures for short queries. 4th Text REtrieval Conference (1995), 277-285.