Internal Use Only

002

TR-IT-0089

Construction and Preliminary Experiments of Document Retrieval System Based on Similarity Between Words

Alexis COLLOMB Kozo OI Eiichiro SUMITA

1995.1

Abstract

In this report, we present an Information Retrieval System combining three types of computation: a semantic distance calculation using a thesaurus, a classical term-weighting method and a physical distance computation intended to reflect how scattered query words can be found in documents. We present our experiments and argue about the quality of the results. Eventually, we introduce new directions and attempt to define heuristic orientations for applying semantics to Information Retrieval.

ATR Interpreting Telecommunications Research Laboratories

Contents

1 Introduction

With the constantly increasing volumes of information circulating around the world, today more than ever before, there is a strong need for intelligent Information Retrieval systems able to select the appropriate data within large databases. This study aimed at designing and implementing a full-text document retrieval system for English. This method combines 3 types of computations :

- semantic distance between words using a thesaurus.
- classical term-weighting.
- physical proximity coefficients for queries representing how scattered query words can be found in documents.

We first had to design the thesaurus. Then, we implemented the preprocessing filters and the processing package. It was done in C on Sun/Sparc workstation.

We begin by situating our work within a general overview of Information Retrieval (IR). After a description of how we designed the thesaurus, we describe precisely the computations performed. We then present our experiments analysing separately the effects of each type of computation.

Information Retrieval evaluation criteria are usually partitioned in two categories: efficiency and effectiveness [15]. The effectiveness of an information system is the ability to furnish information services that the users need. On the other hand, efficiency is a measure of the cost of the time necessary to perform a given set of tasks. Ultimately, the viability of a system depends on both the quality and the cost of the operations. This study, however, only deals with effectiveness.

2 Information Retrieval

2.1 What is Information Retrieval?

Information Retrieval consists in selecting the most appropriate information for a user's need. An ideal IR system is one that retrieves all the relevant documents and only those. To assess the quality of such a system, two criteria are used: precision and recall[15].

Recall is defined as :

$$recall = \frac{number\ of\ relevant\ documents\ retrieved}{total\ number\ of\ relevant\ documents}$$

Precision is defined as :

$$precision = \frac{number\ of\ relevant\ documents\ retrieved}{total\ number\ of\ documents\ retrieved}$$

Combined, those two measurements give an accurate idea of the quality of the retrieval system.

2.2 Overview of Previous Works

In a 1987 review, Belkin and Croft[2] distinguish three main directions in IR :

Relating partial-matching techniques to exact-matching techniques They cite the continuum between the two methods provided by the extended Boolean searching[8][14] and the use of Boolean-derived dependencies in probabilistic searching. They also stress the trend to make operational partial-match techniques that, until then, had always be confined to experimental environments even though they were assumed to give better results than more traditional exact-match techniques.

The combination-paradigm This reflects the fact that no technique is considered to be adequate for all purposes and that either a mix of techniques or a principled choice of techniques is required to improve IR system performance.

The newly increasingly complex representations of the request This emphasizes the strong correlation between the information representation, and the retrieval technique used. It is assumed that by increasing the degree of complexity of the query, it will be possible to apply more retrieval techniques in an endeavour to improve effectiveness. The authors mention the rapidly spreading use of knowledge representation schemes associated with artificial intelligence research for that purpose.

In more recent years, IR research has clearly enhanced the two last points stated by Belkin and Croft. New methods have been implemented combining both statistical and knowledge-based approaches. Refinement of query formulation has been achieved through interactiveness. We would like to cite especially the works of Chen, Basu and Dorbin for *generating, integrating, and activating thesauri for concept-based document retrieval*[3]. It illustrates the two previous points. Also important to support the latter is Salton's, Allan's and Buckley's work for *automatic structuring and retrieval of large text files*[12]. It is organized in two phases. Firstly, a global text similarity is computed by comparing the respective text vectors according to a classical vector-processing method. Text pairs without sufficient global similarity are not considered. Secondly, the system allows the user to refine his query by choosing substructures (such as text sections, paragraphs and sentences) to compute local similarities.

It seems that in the next years, overlapping between IR and artificial intelligence (AI) will be increased. Sparck-Jones[16] distinguishes "strong" form of AI, which involves comprehensive knowledge bases and extensive reasoning capabilities, from the application of AI techniques in an information retrieval context. She claims that the former approach to information access is currently infeasible and potentially even inappropriate, whereas the latter approach may have valuable contributions to make. Croft[4], quoting her, states that there is even little experimental evidence to support the weaker claim. However, he classifies overlapping between IR and AI in three categories and gives examples of improvements for the three of them : expert systems, knowledge representation and natural-language processing. Croft mentions the growing interest for both IR and natural-language processing in statistical analysis of large text databases. The representations produced by natural language-processing techniques can then be combined with the simpler-based representations of typical statistical information retrieval systems. Another promising direction is machine learning categorization where predefined categories are assigned to new documents. Nevertheless, according to Croft, it seems that, even though combining knowledge and statistical approaches seems promiseful, the former still have to clearly show their usefulness.

2.3 What we tried to achieve

Our approach followed the combination-paradigm by combining three types of information :

1. Semantic information : we compute the similarity between words using the distance in a thesaurus classification.

2. Statistical information : we use a classical weighting method [13]

3. Closeness : queries are broken into groups. For each group, we compute a global closeness coefficient that represents the physical distance between words.

Though each type of computation taken separetely might not seem very sophisticated, we expected the combination of the three to produce high-quality results for effectiveness. Even if we tried to make the program suitable for speed achievements, it was not our main concern. In this study, we simply tried to maximize retrieval effectiveness in order to assess the intrinsic possibilities of our method.

3 Thesaurus

This English thesaurus was made using Kadokawa's Japanese thesaurus[11] and EDR's Japanese-English bilingual dictionary.

3.1 Kadokawa's Japanese Thesaurus

This thesaurus is used to calculate the semantic distance between the document and query words. The hierarchy of the thesaurus is in accordance with the thesaurus of everyday Japanese written by Ohno and Hamanishi.

The classification is based on modern semantics and the the two lexicographers' intuition on words in the semantic point of view [18]. It is structured as a decimal classification like in libraries. First the whole set of words is divided into 10 classes: class 0, representing the concept [nature] ; class 1 representing the concept [property&state] ; and so forth up to class 9 representing the concept [articles]. Then each class is subdivided into 10 subclasses, and finally so is each subclass. Consequently, the number of bottom classes is a thousand (10*10*10). Therefore each word has a three-digit code (called a semantic code) of the bottom class to which it belongs.

The size of the thesaurus is about sixty thousand (60,000) words of modern Japanese for everyday use. Words in the thesaurus are labeled with definitions, stylistic labels and examples of usage.

3.2 From Japanese to English using EDR's bilingual database

Electronic Dictionary Research Institute's bilingual database provides both information for English and Japanese. Among all the dictionaries provided by EDR, we used EDR's Concept Dictionary. EDR's Concept Dictionary comprise 400,000 concepts. One of EDR's guidelines was to adopt a general representation applicable to various languages. Therefore, we could expect this semantic transfer from Japanese to English to be quite reliable.

Each entry has different fields of information (HeaDWorD(HDWD) / Concept ID(CID) / Part of Speech(PS) etc.) The procedure used to make the bilingual file was the following :

The DB was run in a CID increasing order. Then for a given English(HDWD/CID/PS), all the CID-matching Japanese HDWDs were attached. Last of all, the Thesaurus Codes(TCs) were assigned using Kadokawa's Thesaurus.

Format and examples follow : `EHDWD CID PS *[JHDWD *(TC)]`

No entry in the bilingual file has simultaneously the same EHDWD, CID and PS. Here are some examples :

lead	3f65cd	EN1		(English Noun (EN))
exile	3f6509	EN1	国外退去処分	(EN with Japanese matching Headword (ENJH))
office	3f66ea	EN1	世話 459a 788	(ENJH with Thesaurus Code (ENJHTC))
refine	3f64e1	EVE		(English Verb (EV))
deduct	3f6991	EVE	差引 差引き	(EV with Japanese matching Headword (EVJH))
stack	3f650f	EVE	集積 228	(EVJH with Thesaurus Code (EVJHTC))

After getting rid of those English headwords that either did not have any Japanese correspondents or that did have some but for which TCs were not assigned because those correspondents were not in Kadokawa's thesaurus, we got:

ENJHTC	21,033
EVJHTC	5,430
EOJHTC	6,701

O stands for all other PS different from Nouns and Verbs.

4 Preprocessing

4.1 Filter_1 : Stopper

- Words belonging to a stoplist and not considered significant are removed[6].
- All characters are converted to lower-case.

4.2 Filter_2 : Formatter

This filter organizes the document so that words are assigned their positions in the documents. The output format is :

Headword	Term ID	Term frequency	Positions

It also removes words beginning with a digit.

4.3 Filter_3 : Morphological Analyser

This filter uses a Wide Coverage Morphological Analyser [10]. Each word (considered an inflected form) is given its possible original root forms. For instance, the input `saw` produces the following:

```
saw N 3sg / saw V INF / see V PAST STR
```

The morphological lexicon handles more than 317,000 forms derived from over 90,000 stems.

There are two reasons why we use this morphological analyzer :

1. It provides stemming for inflected forms (`palying --> play`)
2. It assigns PS.

In the Thesaurus used (see Filter_5), PS are necessary for assigning the right TCs. For instance, for the headword `play` you find :

```
play  1   881   492   898    ( 1 indicates that PS = Noun )
play  2   884   891          ( 2 indicates that PS = Verb )
```

4.4 Filter_4 : Weighting

This filters assigns each word a weight according to its frequency within the document and within the collection.

The weight W_{kD} of a term T_k in a document D is defined as follows:

$$\begin{aligned} W_{kD} &= ntf_{kD} \cdot nidf_k \\ &= \frac{tf_{kD}}{max\_tf_D} \cdot \frac{\log \frac{N}{f_k}}{\log N} \end{aligned}$$

where

tf_{kD} : frequency of a term T_k in a document D.

max_tf_D: maximum tf value for any term in a document D.

f_k: the number of documents in which term T_k occurs.

N: the number of documents in a collection.

The first coefficient represents the given word importance within the considered document, the second coefficient represents the word importance within the whole collection.

4.5 Filter_5 : Assigning Thesaurus Codes

It assigns thesaurus codes to all the documents using the thesaurus previously made. One critical factor is the maximum number of TCs that could be assigned to one word : Max_Num_TC (MNTC). In the first phase of our development, we had MNTC = 105. Not only were the results obtained quite poor but the processing time was important. By only keeping the most significant TCs (assuming that the greater its number of occurences, the more appropriate a TC is), we reduced MNTC to 23. The processing time was divided by 3 and the results were greatly improved. We will only mention the experiments made with the final version of our thesaurus.

5 Processing

The processing is done in 3 phases : loading the documents to memory, loading the queries to memory, performing the similarity calculations using the chain architecture.

5.1 Loading the collection documents

This is done using memory allocations. The documents are chained together. Then, from each documents, stem different PS(Part of Speech) nodes. Each PS node has, in turn, 2 trees stemming from it : one for words with TCs and another for words that do not have TCs.

5.2 Loading the collection queries

This is done using memory allocations. Groups are chained together. From each group stems a chain of query words.

5.3 Similarity Calculations

General Description Whatever the query initial format might be (see next section), it is always eventually broken into groups so that query(Q) can be represented as follows:

$$Q = \underbrace{t_{11}, t_{12}, \cdots, t_{1,N_1}}_{g_1\text{(group1)}} \& \underbrace{t_{21}, \cdots, t_{2,N_2}}_{g_2\text{(group2)}} \& \cdots \& \underbrace{t_{M,1}, \cdots, t_{M,N_M}}_{g_M\text{(groupM)}}$$

where '&' is an AND operation.

You have M groups and each group g_i has N_i words.

The final similarity score between a query Q and a document D is obtained through computations at different levels that are thereafter detailed. The following notations are used:

$Sim(t_{ij}, t_D)$: Maximum similarity between t_{ij} and t_D. If the HW(headword) and the PS(part of speech) of t_{ij} and t_D are same(exact-matching), the similarity is 1.0. Otherwise, the similarity is the maximum similarity in similarities of all combinations of thesaurus code(s) of t_{ij} and thesaurus code(s) of t_D. Similarity between thesaurus codes is determined by the way same as EBMT[18] (See table 1).

t_{ij} : A term in Q.

t_D : A term most similar to t_{ij} in the document.

R: Number of retrieved unique terms most similar to t_{ij} (with $Sim(t_{ij}, t_D) \geq$ **T**(**Threshold**)).

w_k: Weight of retrieved terms most similar to t_{ij}.

CL: Closeness among locations of t_{i*} in the document.

c_1, c_2: Coefficients. (e.g. $c_1 = 2$, $c_2 = 10$)

wd: The minimum distance among locations where t_{i*} occur in the document.

n_i The number of words in the Q_group_i for which there was exact-matching.

N_i The number of words in Q_group_i

The threshold, **T**, is the main parameter in the method. By modifying it, we carried out some experiments.

Q_Word/D_Word Level Similarity between 2 words $Sim(t_{ij}, t_D)$ is computed in the following way :

- If the query word t_{ij} has no TC, then we look for exact-matching in the no_TC_tree stemming from the right PS node. If an exact-match is found, then the score 1.0 is attributed, else it is 0.

- If the query word t_{ij} has TCs, then we look for document words with the same PS in the TC_tree. The semantic similarity is 1.0 for an exact-match, else the thesaurus is used to calculate the semantic distance between the two according to the rules previously described.

Table 1 explains the similarity between two thesaurus codes.

Q_Word/Document Level The previous operation is carried out between all document words and t_{ij}. Then the sum is made according to the sum of weights for

Table 1: Similarity between two thesaurus codes

Condition	Example	Similarity
$CI_1CI_2CI_3 = CD_1CD_2CD_3$	347 , 347	0.75
$CI_1CI_2 = CD_1CD_2, CI_3 \neq CD_3$	347 , 346	0.50
$CI_1 = CD_1, CI_2 \neq CD_2$	347 , 337	0.25
$CI_1 \neq CD_1$	347 , 247	0.0

document words that gave a similarity score greater than T.

$$Sim(t_{ij}, D) = \frac{\sum_{k=1}^{R} w_k \cdot Sim(t_{ij}, t_D)}{R}$$

Q_group/Document Level The sum of t_{ij} scores belonging to a group is made. The closeness —or proximity coefficient— is computed for one group. If we want to remove closeness, we just set this coefficient to 1. This coefficient is computed as

$$CL(g_i, D) = \frac{1}{\frac{wd+1-N_i}{c_2}+1} \cdot \frac{n_i}{N_i}$$

Query/Document Level All group scores are summed to give the final formula

$$Sim(Q, D) = \frac{\sum_{i=1}^{M} \left\{ \left(\sum_{j=1}^{N_i} Sim(t_{ij}, D) \right) \cdot CL(g_i, D) \right\}}{\sum_{i=1}^{M} N_i}$$

6 Experiments

6.1 Description of the 2 Test Collections

IR experiments often use test collections which consist of a document database and a set of queries for the database for which relevance judgments are available. The number of documents in test collections has tended to be small, typically a few hundred to a few thousand documents. Test collections are available on an optical disk[7]. We chose two collections : the CACM and the MED collections[9].

CACM collection Its subject is Computer Science. It consists of 3204 documents and 64 queries. Information on number of words and TC distribution can be found in Table 2 and 3.

MED collection Its subject is Medicine. It consists of 1033 documents and 30 queries. Information on number of words and TC distribution can be found in Table 2 and 3.

Table 2: TC distribution(%) in the thesaurus dictionary, CACM collection and MED collection

TC Class	0	1	2	3	4	5	6	7	8	9
Thesaurus	7.6	17.7	11.8	10.6	11.6	6.8	12.8	7.8	6.2	7.0
CACM	2.7	19.2	15.3	15.0	16.5	2.0	4.7	8.9	12.0	3.5
MED	7.6	18.4	18.4	13.4	14.6	2.6	7.1	8.3	6.9	3.0

Table 3: Average number of words in a query and document and ratio of words with TC and with no TC in collections

	Average number of words		Ratio of words (%)	
Collection	Query	Doc	with TC	with no TC
CACM	16.4	27.6	73.6	26.4
MED	14.3	70.0	64.9	35.1

6.2 Description of different Query Formats

The queries were processed in two formats :

- The original format : simple text.
- The boolean format : the original queries are converted in boolean formats.

6.2.1 The simple-text format

Queries are processed in a natural way. Every sentence makes a group. Sentence delimiters are ".", "?" and "!".

As an example, this is query_35 of CACM collection :

```
Probabilistic algorithms especially those dealing with algebraic
and symbolic manipulation. Some examples:
Rabiin, "Probabilistic algorithm on finite field", SIAM
Waztch, "Probabilistic testing of polynomial identities", SIAM.
```

This query would be preprocessed exactly as documents were, each sentence becoming a separate group in the output.

6.2.2 The boolean format

The plain-boolean format The simple-text queries are converted into boolean queries before preprocessing so that group partitioning is different.

For instance, the same query_35 of CACM collection becomes :

```
#q35= #and( 'probabilistic', 'algorithm',
            #or( 'algebraic', 'symbolic'),
            'manipulation');
```

Here every word between single quotes will constitute a query group. Applying closeness computations —as described above— is not interesting because every group has only one word in it so miminum_closeness is automatically one.

The boolean-for-closeness format This boolean version was made from the previous one to study closeness effect for boolean queries.

Query_35 of CACM becomes :

```
#q35= #and( 'probabilistic algorithm',
            #or( 'algebraic manipulation',
                 'symbolic manipulation'));
```

In this case, every group has two words in it.

Boolean queries were only available for CACM collections. By combining AND & OR operations, each formal query produced a set of query files for which similarity computations were performed. The final score was the maximum of all scores obtained.

Still with same query_35 of CACM, we get —for both plain- and closeness-formats— 2 possible queries. After processing the score retained is the maximum of the two.

Variation of similarity calculation for boolean format The original boolean query Q gives different combinations Q_c as follows:

$$Q = Q_1 \mid Q_2 \mid \cdots \mid Q_c \mid \cdots$$

where '|' is an OR operation.

Similarity between query Q_c and document D is computed as described in the previous section and the maximum becomes the final score

$$Sim(Q,D) = max(Sim(Q_1,D), Sim(Q_2,D), \cdots, Sim(Q_c,D), \cdots)$$

6.3 Description of Result Computations

For assessing the results, we chose precision-recall graphs since using this method seemed to be the most accurate. For a given level of recall (number of relevant documents retrieved / total number of relevant documents in the collection), the precision is computed (number of relevant documents retrieved / total number of documents retrieved). This is done for each query, then an average is made for all the queries. In every case, 100 recall levels were considered (from 1% to 100%).

6.4 Modification of Threshold

The threshold is the main parameter in the method. By modifying it, we can evaluate the method effectiveness from exact-matching (T=1.0) to taking into account all similarity scores (T=0.0). For each collection, we provide the different results without taking closeness into account.

Figures 1, 2 and 3 respectively show results for simple-text format of CACM collection, simple-boolean format of CACM collection and simple-text format of MED collection at T=0.25, 0.5, 0.75 and 1.0.

Those figures show that :

- For all three cases studied, the higher the threshold, the better the results. This means that we get the best results with exact-matching.
- There are only slight differences between the Plain-Boolean and the Plain-Text formats for CACM collection.
- In Plain-Text format, the results for MED are much better than for CACM : precision is nearly doubled for *recall* $<=$ 0.5.

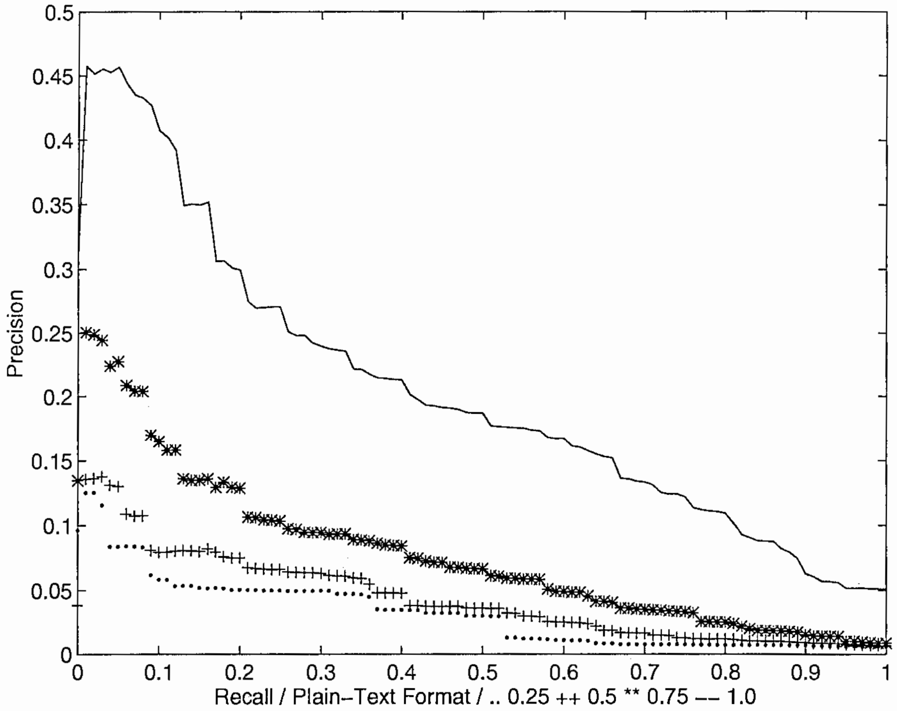

Figure 1: CACM Collection: simple-text format

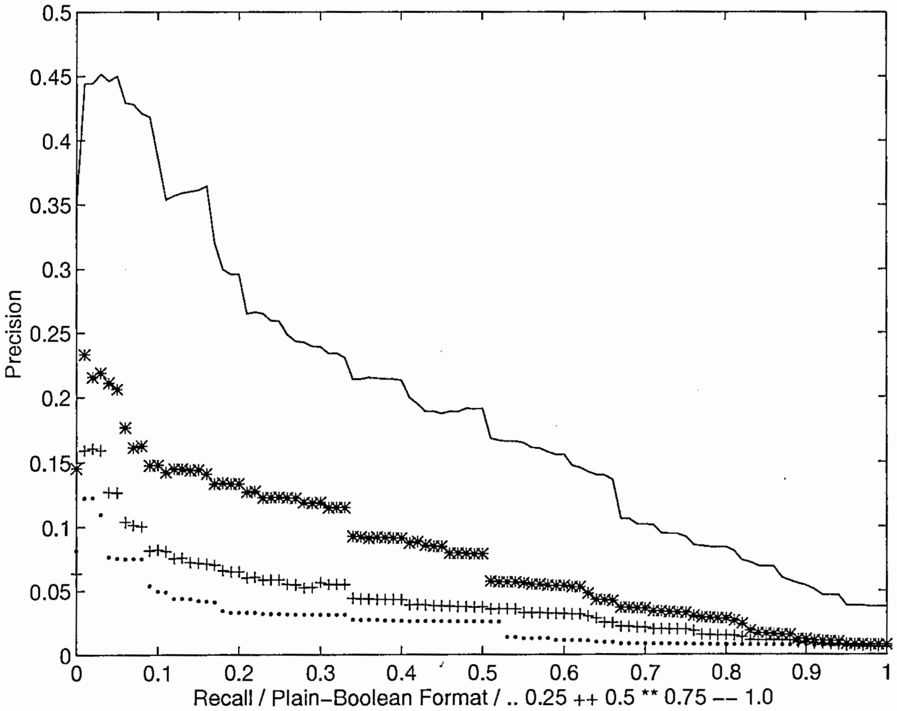

Figure 2: CACM Collection: simple-boolean format

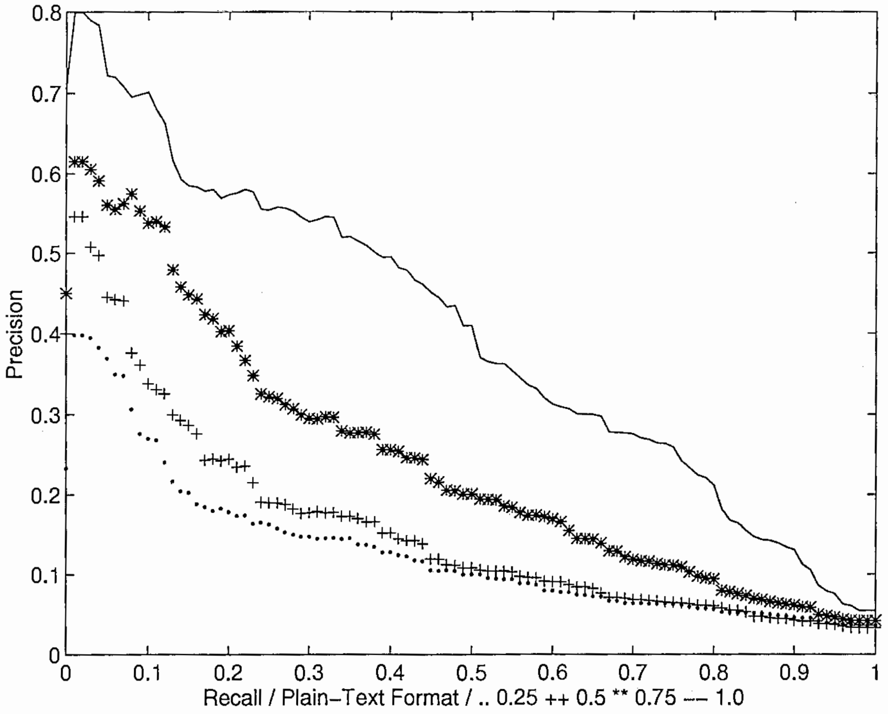

Figure 3: MED Collection: simple-text format

6.5 Adding Closeness

This illustrates the effect of closeness implementation for retrieval effectiveness.

Figure 4 and 5 shows improvements by adding closeness for boolean-for-closeness format of CACM collection at T=0.75 and T=1.0 respectively.

Figure 6 and Figure 7 shows improvements by adding closeness for simple-text format of MED collection at T=0.75 and T=1.0 respectively.

Whether for MED or CACM, those figures show that :

- For both $T = 0.75$ and $T = 1$, the use of closeness provides results that are above all better.
- The use of closeness is much more efficient for $T = 0.75$ (precision can be increased by 0.2) than for $T = 1$.
- There are only small differences between $T = 0.75$ and $T = 1$ when closeness is implemented.

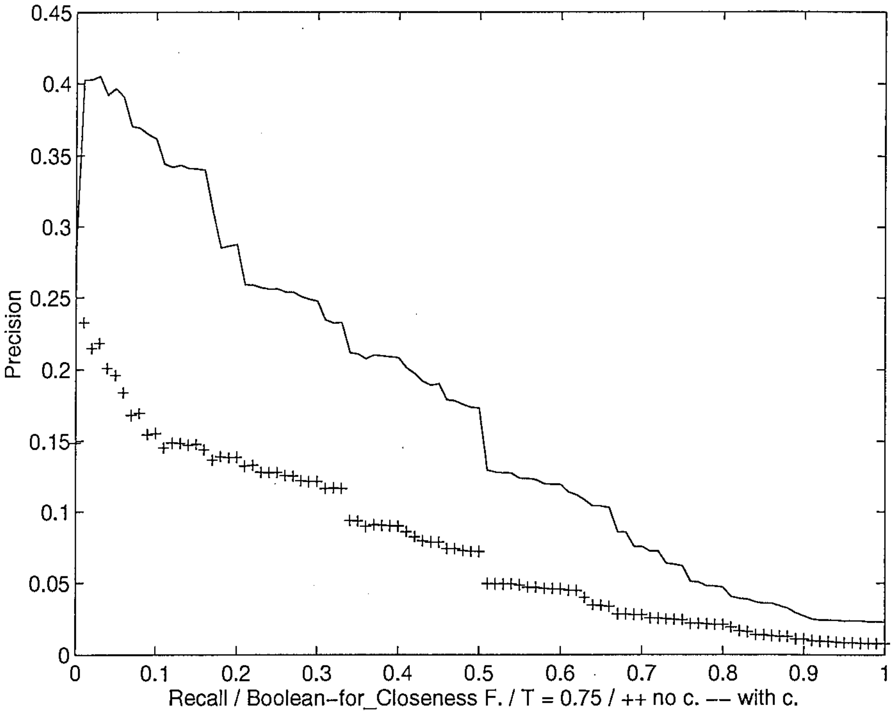

Figure 4: CACM Collection: boolean-for-closeness format = [T=0.75, with closeness / no closeness]

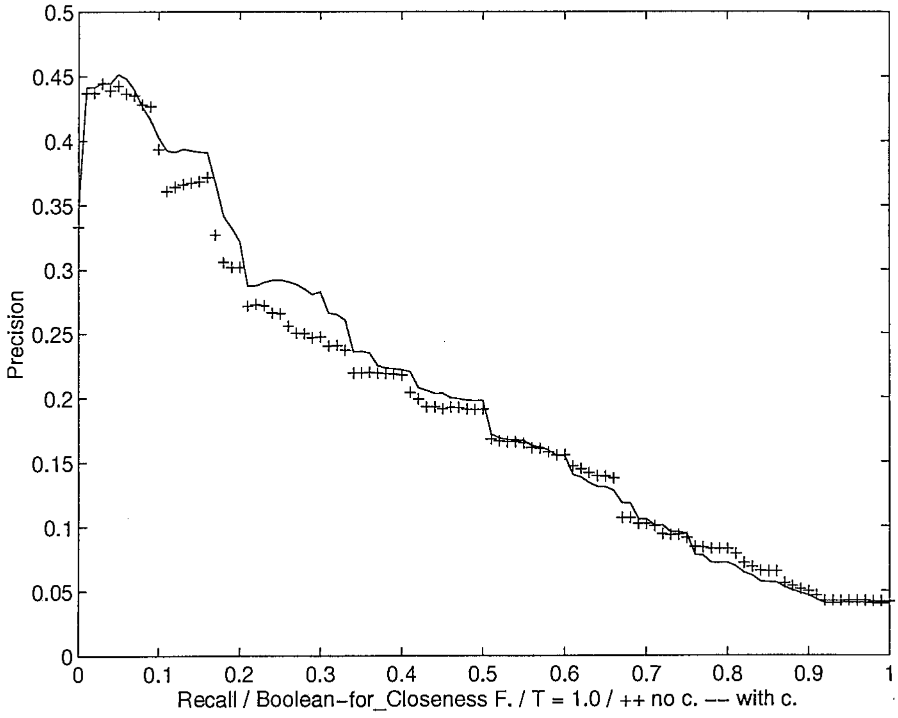

Figure 5: CACM Collection: boolean-for-closeness format = [T=1.0, with closeness / no closeness]

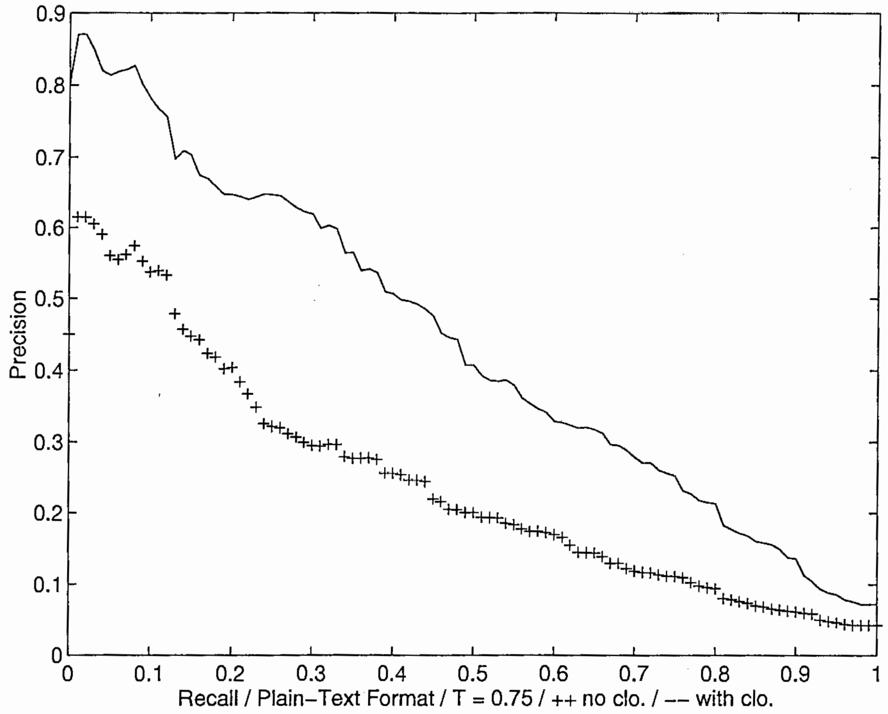

Figure 6: MED Collection: simple-text format = [T=0.75, with closeness / no closeness]

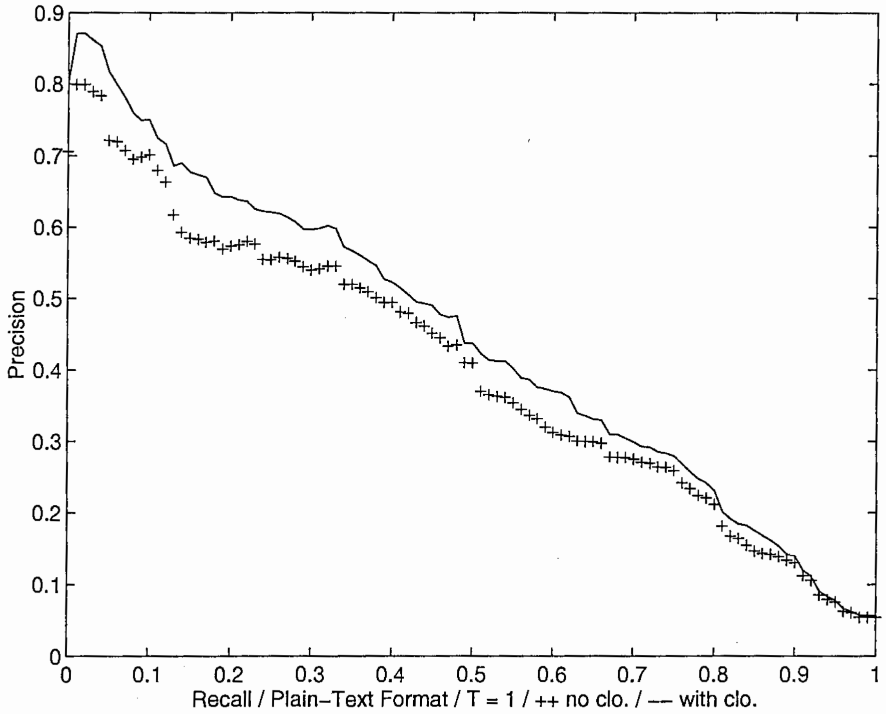

Figure 7: MED Collection: simple-text format = [T=1.0, with closeness / no closeness]

7 Evaluation and Analysis of Results

7.1 Modification of Threshold

7.1.1 Exact-Matching(T=1.0) vs. Thesaurus Use(T=0.75)

Using a thesaurus in order to achieve better retrieval effectiveness was one of our main goals. The previous results clearly show that T=1.0 gave the best results. This leads to the question: why do we get better results with exact-matching than when taking into account semantic distance for not exactly-matched words using the thesaurus?

The idea is very straightforward: the greater the proportion of D_words giving similarity scores above the threshold in relevant over irrelevant cases, the better the results. We will use the concept of **score-ratio** for describing this proportion.

Description of the score-ratio concept Let us define:

$$X_{D,T}(w_{ij}) = \frac{1}{R_{D>=T,T}} \cdot \left(\sum_{k=1}^{R_{D,T}} w_k.T \right)$$

where T is the value of threshold used and D the value of Q_Wd/D_Wd similarity that is of interest.

$$A_{D,T} = \frac{\sum_{i=1}^{M} \left\{ \sum_{j=1}^{N_i} X_{D,T}(w_{ij}) \right\}}{\sum_{i=1}^{M} N_i}$$

$A_{D,T}$ is the average over all words of a query of $X_{D,T}$. The score-ratio $SR_{D,T}$ will be defined as

$$SR_{D,T} = \frac{Average_{rel\_cases}\,(A_{D,T})}{Average_{irrel\_cases}\,(A_{D,T})}$$

and the total score-ratio SR_T as

$$SR_{Total,T} = \sum_{D>=T} SC_{D,T}$$

The case of exact-matching(T=1.0) The final score formula, $Sim(Q,D)$[1], is:

$$Sim(Q,D) = \frac{\sum_{i=1}^{M} \left\{ \sum_{j=1}^{N_i} \left(\frac{1}{R_{ij}} \cdot \sum_{k=1}^{R_{ij}} w_{ijk} \right) \right\}}{\sum_{i=1}^{M} N_i}$$

[1] See subsection 5.3

where R_{ij} is the number of document words matching exactly the query word t_{ij}.

R_{ij} is in that case equal to 0 or 1. With the previous notations

$$Sim(Q, D) = A_{1,1}$$

The case of thesaurus use(T=0.75) In order to understand why the scores are lowered when thesaurus is used(T=0.75) than with simple exact matching, we need to assess separately the respective contributions of those words that gave Sim(t_{ij},t_D)[2] of 0.75 and those words that gave exact-matching.

R(= $R_{1.0} + R_{0.75}$) is the number of words that gave $Sim(t_{ij}, t_D) >= 0.75$.

$R_{0.75}$ represents the number of document words that produced $Sim(t_{ij}, t_D)$ of 0.75; in other words, the number of document terms that shared at least one TC with the query word without matching it exactly.

$$\begin{aligned} Sim(t_{ij}, D) &= \frac{\sum_{k=1}^{R_{1.0}} w_k + \sum_{k=1}^{R_{0.75}} w_k \cdot 0.75}{R_{1.0} + R_{0.75}} \\ &= \frac{\sum_{k=1}^{R_{1.0}} w_k}{R_{1.0} + R_{0.75}} + \frac{\sum_{k=1}^{R_{0.75}} w_k \cdot 0.75}{R_{1.0} + R_{0.75}} \\ &= U + V \end{aligned}$$

And the final similarity formula becomes:

$$Sim(Q, D) = \frac{\sum_{i=1}^{M} \left\{ \sum_{j=1}^{N_i} [U] \right\}}{\sum_{i=1}^{M} N_i} + \frac{\sum_{i=1}^{M} \left\{ \sum_{j=1}^{N_i} [V] \right\}}{\sum_{i=1}^{M} N_i}$$

also simply written with the previous notations

$$Sim(Q, D) = A_{0.75,0.75} + A_{1,0.75}$$

Table 4 and Table 5 show those separate controbutions for CACM and MED.

Table 6 shows the score-ratios for the two collections. If we assume, the greater the score-ratio, the better results, we can justify the figures obtained.

Firstly, within one collection, we see that exact-matching is more selective in terms of relevant over irrelevant cases than TC similarity of 0.75. It also shows that, for exact-matching, this selectiveness is greater for MED than for CACM, and therefore accounts for better results for MED.

[2] See subsection 5.3.

Table 4: Separate Contributions of Exact-Matching and Thesaurus for CACM

CACM	$A_{1,1}$	$A_{0.75,0.75}$	$A_{1,0.75}$
Rel	0.0677	0.0635	0.0550
Irrel	0.0092	0.0517	0.0075

Table 5: Separate Contributions of Exact-Matching and Thesaurus for MED

MED	$A_{1,1}$	$A_{0.75,0.75}$	$A_{1,0.75}$
Rel	0.0576	0.0239	0.0501
Irrel	0.0047	0.0226	0.0035

Table 6: Score-Ratios for MED and CACM

SR	$SR_{Total,1}$	$SR_{0.75,0.75}$	$SR_{1,0.75}$	$SR_{Total,0.75}$
MED	12.3	1.1	14.3	2.8
CACM	7.4	1.2	7.3	2.0

7.1.2 Plain-Boolean vs. Plain-Text Formats

The two formats provide very similar results using CACM as the test collection. In fact, the interest of the Boolean formats is to cluster together in a group those words that usually belonged to a lower-than-sentence syntactic unit. But without implementing closeness, breaking original queries into Boolean requests is practically effectless. The only thing that changes is that Boolean queries can be shorter than their original simple-text counterparts.

7.1.3 MED vs. CACM

A direction of research for explaining further this difference would be to investigate on crossing-over between document terms. It is possible that MED documents are more term-specific than CACM ones, in which case there would be less accidental exact-matching for MED than for CACM and a better score-ratio. The more specific, lexically speaking, documents are, the lesser the chances of seeing a document considered irrelevant having terms in common with relevant documents.

7.2 Adding Closeness Calculation

Using the closeness clearly improves the results. We think that the reason why there are little differences between the cases when T = 0.75 and T = 1 is because our way of computing proximity coefficients is strongly correlated to exact-matching investigation through the $\frac{n_i}{N_i}$ coefficient.

7.3 Discussion

One important point of this study was the attempts to use a thesaurus. We have demonstrated that the thesaurus could not be really helpful as it was. Now we are facing the following question: is it just because our thesaurus is not appropriate or could the same phenomenon take place with even acknowledged English thesaurus?

7.3.1 Inadequacy of Thesaurus Construction

The thesaurus is the core of the method. Therefore, we can expect that the quality of the results will strongly depend on the quality of the thesaurus. We believe now that the way the thesaurus was made was not appropriate.

Kadokawa's thesaurus was made by attaching most Japanese words to concept categories. By using EDR classification that is different from Kadokawa's for transferring Kadokawa's hierarchy from Japanese to English, we assign multiple TCs to English HDWDs. More precisely, every English HDWD is assigned his Japanese correspondents in EDR and then is attached all the TCs each of those Japanese correspondents have in Kadokawa's Thesaurus. Whereas a normal procedure would have been to assign semantic codes from an English Thesaurus, this two-level attribution multiplies the number of TCs and greatly widens —while in the mean time blurring— the semantic attributes of English headwords.

Another problem lies in the use of assigning Japanese concepts to English words. Though the idea of using universal concepts applicable to all languages is appealing, finding such universal characters is practically very difficult[17].

Last, it is likely that the general thesaurus obtained is much too broad for dealing with special collections as MED and CACM repectively concerned with Medicine and Computer Science. For Table 2, by using the cosine correlation metric[2], we found a similarity of 0.9495 between MED and the Thesaurus(THE) used for assigning TCs, 0.9164 between CACM and THE and 0.9746 between MED and CACM. This suggests that the Thesaurus is more appropriate for MED than CACM and goes with the results. However, the three distributions still appear to be quite close and, in our opinion, it seems obvious that the Thesaurus cannot take into account subtle technical differences of the fields covered by MED and CACM. One way of coping with that problem would be by generating automatically thesauri from the the large corpora IR is going to be performed on.

7.3.2 Using Semantics by Treating Words as Independent Entities

By using semantic information in the form of TCs independently for each word, we go against the acknowledged fact that semantics of text is not well represented by surface features such as individual words[5]. As Salton, Allan and Buckley put it referring to Wittgenstein's use theory[12] [19] :

> *The use theory is especially appropriate in IR in which the major concern is not with the intrinsic meaning of the words and text units in isolation but with the global meaning of complete text entities.*

7.3.3 The Ambiguities Introduced by the Morphological Analyzer

The use of the morphological analyzer introduces ambiguities and damages the original semantics of the documents. In fact, the way the input `saw` produces

```
saw N 3sg / saw V INF / see V PAST STR
```

clearly shows that it multiplies documents'number of words —for one entry, you get three in the output— having completely different meanings. If in a query and a document, you have `saw` respectively meaning `to see` and `to saw`, you will get three exact-matches for two words that had no meaning in common.

8 Conclusion and Future Directions

We showed that our method did achieve significant results combining both proximity computations and classical weighting method. However, the use of the thesaurus was disappointing. In our opinion, this can be explained by both the way the thesaurus was constructed and by the intrinsic inadequacy of using thesaurus information by considering words as semantic entities. Furthermore, it is quite clear that the morphological analysis used in the preprocessing is irrelevant, especially in that respect.

Those are the future directions we propose :

- The source of the semantic information should be rethought. We either propose to generate automatically a very specific thesaurus from the large corpora IR is going to be performed on or build a synonym dictionary in the form of a Bayesian network. It is important to assess the difference between a synonym dictionary and a thesaurus [17]. Either way, some learning should be performed meanwhile as to be able to cope with semantic ambiguities and desactivate them using statistics on semantic adjacency constraints (concept A and concept B usually come together in the same document with the probability p). By doing so, research might be undertaken on automatic topic extraction or categorization.

- The morphological analyzer, if not integrated to a parser, should be removed and replaced by a simple stemmer.

- The physical proximity calculation could be refined and still improved. In this study, it turns out to be useful. We feel the need to mention here that those computations can take time and that it might be appropriate, based on recent studies such as Al Haj's[1], to implement these calculations on parallel computers.

We think that combining all semantic, statistical and physical distance methods is very valuable though ambitious. In the future, we suggest those processes had the forms of :

1. Topic extraction for using a semantic similarity metric on documents or part of documents, but not just words.

2. Bayesian networks for inference in case of ambiguities in the above process.

3. Both of the above sources, whether semantic or statistical, should be developped from the database itself on which IR is to be performed.

4. Term-weighting and physical distance calculations.

References

[1] AlHaj, A.M., Sumita, E. and Iida, H.: "Text Retrieval Using Parallel Computers," *Proc. of the 50th Annual Convention of IPSJ*, March 1995.(to be published)

[2] Belkin, N.J. and Croft W.B.: "Retrieval Techniques," *Annual Review of Information Science and Technology (ARIST)*, Vol. 22, 1987.

[3] Chen, H., Lynch, K.J., Basu, K. and Dorbin, T.: "Generating, Integrating and Activating Thesauri for Concept-Based Document Retrieval," *IEEE Expert*, Vol. 8, No. 4, April 1993.

[4] Croft, W.B.: "Approaches to Intelligent Information Retrieval," *Information Processing and Management*, Vol. 23, No 4, 1987, pp. 249-254.

[5] Croft, W.B.: "Knowledge-Based and Statistical Approaches to Text Retrieval," *IEEE Expert*, Vol. 8, No. 4, 1993.

[6] Fox, C.: "Lexical Analysis and Stop Lists," in Frakes, W.B. and Baeza-Yates, R., editors, *Information Retrieval Data Structures & Algorithms*, pp.102–130, Prentice-Hall, ISBN: 0-13-463837-9, 1992.

[7] Fox, E.A., ed.: *Virginia Disk One*, Virginia Polytechnic Institute and State University Press, Blacksburg, 1990.

[8] Fox, E.A., Salton, G. and Wu, H.: "Extended Boolean Information Retrieval," *Communications of the ACM*, Vol. 26, No.11, pp.1022–1036, 1983.

[9] Frakes, W.B.: "Introduction to Information Storage and Retrieval Systems," in Frakes, W.B. and Baeza-Yates, R., editors, *Information Retrieval Data Structures & Algorithms*, pp.1–12, Prentice-Hall, ISBN: 0-13-463837-9, 1992.

[10] Karp, D., Schabes, Y., Zaidel, M. and Egedi, D.: "A Freely Available Wide Coverage Morphological Analyzer for English," *Proc. of COLING'92*, 1992.

[11] Ohno, S. and Hamanishi, M.: *Ruigo-shin-Jiten*, Kadokawa, 1984.

[12] Salton, G., Allan, J. and Buckley, C.: "Automatic Structuring and Retrieval of Large Text Files," *Communications of the ACM*, Vol. 37, No. 2, February 1994.

[13] Salton, G. and Buckley, C.: "Term weighting approaches in automatic text retrieval," *Information Processing and Management*, Vol. 24, No. 5, pp.513-523, 1988.

[14] Salton, G. and Voorhees, E.M.: "Automatic Assignment of Soft Boolean Operators," *Proc. of 8th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ACM, pp.54-69, 1985.

[15] Salton, G. and McGill, M.J.: *Introduction to Modern Information Retrieval*, McGraw-Hill Computer Science Series, 1983.

[16] Sparck Jones K.: "The Role of Artificial Intelligence in Information Retrieval," *J. of the Am. Soc. for Information Science*, Vol. 42, No. 8, pp. 558-565, 1991.

[17] Sparck Jones, K.: *Synonymy and Semantic Classification*, Edinburgh University Press, 1985.

[18] Sumita, E. and Iida, H.: "Example-Based Transfer of Japanese Adnominal Particles into English," *IEICE TRANS. INF. & SYST.*, Vol. E75-D, No.4, July 1992.

[19] Wittgenstein, L. *Philosophical Investigations* Basil Blackwell and Mott Ltd., Oxford, England, 1953.