A Method for Retrieving and Building Structure of NLP Knowledge

Elmarhmoy Ghada, El-Sayed Atlam, Masao Fuketa, Kazuhiro Morita and Jun-ichi Aoe

Dept. of Information Science and Intelligent Systems
University of Tokushima, Tokushima, Japan 770-8506

Summary
Knowledge-based natural language understanding requires the representation of various types of knowledge like linguistic knowledge, conceptual knowledge or knowledge about real world objects. This paper proposes a method of a retrieving and building structure of natural language processing knowledge. We classify all surface case patterns in advance and then consider the typical meaning of noun which has one of these patterns. We present also an efficient data structure by introducing a trie that can define the linkage among leaves. The linkage enables us to share the basic words required for multi-attribute relations. By using this linkage, and a high frequent access between verbs with noun, we could extract an automatic generation of hierarchical relationships. This new method applied to the data extracted from a large tagged corpus (Pan Treebank). From relationships for 11,000 verbs and nouns, it is verified that the presented method is simulated the number and frequent of typical verb, and made a hierarchy group of its noun, and net of linkage group with this high frequent.

Keywords:
Natural Language Processing, trie structure, Hierarchy.

1. Introduction

Natural language processing (NLP) systems utilize a variety of dictionaries. In this paper, two types of information are discussed. One is morphological information about morphemes, or words, and their fundamental attributes such as a part of speech [7], semantic primitive [8][9][24], and so on. A morphological dictionary [16] of a morphological analyzer provides the morphological information with any NLP systems. In general, the space storing morphological information is proportional to the number of words. Another is relational information storing a word pair and the attribute of the relationship such as <co-occurrences>, <selective words>, compound words, idioms and so on. <co-occurrences> defines a pair of words which explain....<selective> ,<compound words>.... Relational information like compound words is formed infinitely, so it takes a lot of spaces if they are registered in the morphological dictionary. Basic knowledge IS-A and PART-OF of artificial intelligence (AI) is also based on word relationships.

A case frame [17][23] is one of important knowledge to solve ambiguity in syntax and semantic analysis [12][13]. Japanese to English, or English to Japanese, machine translation systems [17] need to built translation dictionaries by using the case frame. A question and answering (Q & A) module [14] in the natural language interface (NLI) systems understands user's requests by using case analysis. The case frame belongs to relational information because there are relationships the roles (subject, object, place, etc.) of noun phrases between verbs.

Recently, many types of NLP systems have been developed and there are many individual dictionaries with common knowledge each other, but it is difficult to combine these dictionaries because there is no efficient method for integrating common knowledge. Integrating dictionaries enable us to reduce not only the dictionary space but also heavy costs building and maintaining dictionary, and consequently, it is possible to develop more intelligent NLP systems. From this reason, the aim of this paper is to merge the morphological information and relational information into one data structure and to derive new information from the integrated structure.

In order to store compound words into the compact structure, and Aoe et al. [1][2][3][4] and Morita et al.[18] presented a two-trie structure. Moreover, Morita et al. [19] and Onoo et al. [22][23] presented a link trie. In this paper, the link trie is utilized and a method of storing relational information is presented. Section 2 describes some of relational information to be discussed here as multi-relationships between words. A case frame is defined as the essential knowledge that can connects relational information. In Section 3, the link trie is introduced and a method of integrating morphological and relational information is presented. Section 4 discusses automatic knowledge building from an unknown word and the context and this section divide into three subsection (i) Automatic generation of hierarchical relationships, (ii) Automatic hierarchy from link trie information, and (iii) Extracting new information “Similarity”. In Section 5, it is verified that the presented method is excellent by the simulation results and define new method. In section 6, Conclusion and future work.

2. Multi – Relationships among words

Morphological information for word x is defined as MOR(x) simply, so relational information, call a multi-attribute relation, for a finite of relational attributes will be discussed.

Information about multi-attribute relation can be defined as a triplet (x, y, Alpha), where there is a relationships between x and y, and the attribute is Alpha. In natural language processing there are a variety of attributes we can get, and clearest meaning by using relationships among words as follows.

2.1 Case frame

In order to accomplish better Natural Language Process System (NLPS), it needs to solve some problem, such as word selection to be translated in machine translation systems, constraint of recognized words on voice recognition, disambiguation of natural language analysis, and so on . To cope with this difficulty we have to employ some syntactic and semantic information simultaneously for the analysis of a sentential structure. The best grammatical framework for this application is the case grammar which was first discussed by (C. Fillmore in 1968). A similar idea existed in Japan long before Fillmore, and had been used to some extent. After his paper, there were many improvement and change in theory for purpose of applying the idea actually to the computer analysis of natural language process.

Considering the example of Machine Translation (MT) by [M. Nagao, et. al. [20],[23], the semantic primitive is used to specify what kind of noun can be in what case slot. For example, the verb *eat* demands a noun associated with one of the semantic primitive *animal* as the agent of the verb, and noun of semantic code *eatable material* as an object. This case slot specification is given for each usage of every verb in a dictionary.

We can defined the relations VERB and NOUN (OBJECT, PLACE) relation, as follows:

John live in Japan <*VERB –PLACE*>
John speak English <*SUB. – VERB*>
Cat eat food <*VERB – OBJ.*>
Cat swim in water <*VERB – OBJ.*>
Ahmed treat sickness <*SUB. – VERB*>

2.2 SIMILARITY (Semantic Primitives)

Similarity relation denoted by (x, y, α) means that *x has similar semantic as y*, also gives us more information, because of a word (or a word collection) can occur in any number of synonyms ,with each synonyms reflecting a different sense (meaning) of the word, e.g. <Job, ”Occupation, Business”, SIMILARITY>,i.e. <OBJ.> Similar to <OBJ.>, this mean that word *Job is similar of words Occupation & Business,* which give us more information retrieval about words and make relation group as follows:

Effect : Result, Action, Operation, Impression
Job : Occupation, Business.

2.3 IS – A Relation

IS -- A relation denoted by (x, y, α) means *x is a y*, which give us the relation of words and its differentiates , e.g. <Ahmed, Pilot , IS – A>, < Ahmed, Human, IS -- A> ,< Jhon, player, IS – A > By using case frame < SUB.> IS – A < OBJ.>, we extract also the hierarchy *Pilot IS – A Human*, and Similarity <Pilot, Player, Similarity> ,also we can understand the distinguished between words and make it disambiguation as follows:

Ahmed is a doctor
My father is a teacher

2.4 PART – OF Relation

PART— OF relation denoted by (x, y, α) means *x is a part of y,* which give us more semantic relation and related word, and which often indicate the type of words e.g. <Head, Body, PART --OF>, <SUB.> PART—OF <OBJ.>. From this information we can get more information that Body has a head, Body consist of Head and Other. This means if x PART – OF y, this mean y consist of x, y has x, y make from x, and as shows in this follows examples:

Head is a part of body
Engine is a part of car

2. 5 COMPOUND WORD <Co- Occurrence>

Compound Word relation denoted by <x, y, α) means *x compound with y to give new information* e.g. if we write language is processed by a computer, by using case frame relation we can get <tool> + <Verb> → *computer processing*, and <Subject> + <Verb> → *language processing*, also; Document is classified by a system, from case frame we can extract <OBJ.> + <VERB> → Document classification, and <VERB> + <SUB.>→ classification system. i.e. the clearest meaning and information about word rather than if the word is single, another example as follow:

Information Science
Computer science

3. Link Trie (LT) Function [K. Morita, 18]

3.1 Definition of Word Relationships

Link information about the functions RELATION and ATTRIBUTE is frequently used for retrieval and updating of a key, so it is important to introduce an efficient data structure for those functions as well as a data structure of state transitions. Let (*X*, *Y*, R) be the relation *R* between words *X* and *Y*. In the tries, there are one-to-one correspondence between leaves and keys, so its link trie is defined by connecting leaf *s* for *X* and leaf *t* for *Y*.

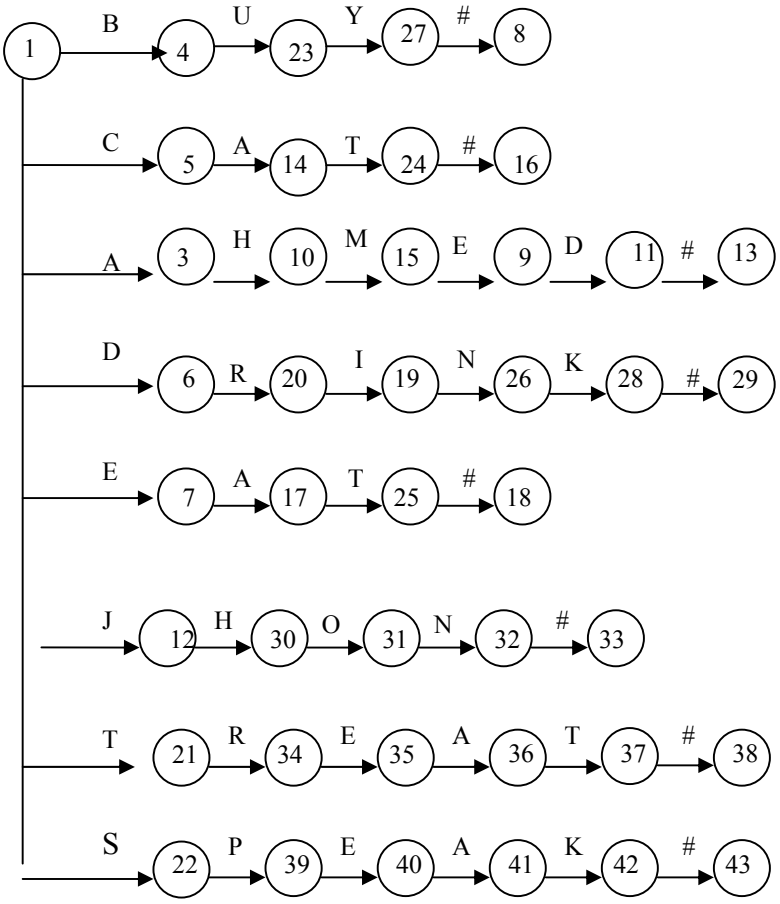

Figure 1. Trie structure

1. Trie structure

In this case, the function RELATION is defined by $t \in$ RELATION (*s*) and the relation is defined by the record R ∈ATTRIBUTE (s, t). A trie including the function LINK and ATTRIBUTE is called a link trie. Table 1 represents link information for Figure 1. We can find the relation between the word "*Ahmed"* as a subject and "*buy"* as a verb by the trie and there is one-to-one corresponding, we find the leaf 13 correspondence key "*Ahmed"* and leaf 8 correspondence key *buy*, and link function is defined by 8 ∈ RELATION(13) and the record <SUB-VERB> ∈ ATTRIBUTE (8, 13). By the same way, we can find the relation between words <VERB -- OBJ.>, <VERB--PLACE>, <PART-- OF>, <COMPOUND WORD>, and <SIMILARITY> as follows:

3.2 Retrieval Algorithm

For the relation (*X*, *Y*, *R*), the retrieval algorithm to be proposed here includes (1): retrieving *Y*, and R from *X*,(2): retrieving R from *X* and *Y*. For LT and for key *X*, the function GET_LEAF (LT, *X*) returns the leaf for *X#* and returns fail if LT has no *X#*. The function GET_LEAF (LT, "Result") returns leaf 77 in Figure 1. For the relation (*X*, *Y*, *R*), the following ALGORITHM retrieves leaves *s* for *X#* and *t* for *Y#* if they are registered in the trie. *s* and *t* can be used to retrieve ATTRIBUTE (*s*, *t*) including relation R. If either *s* or t has not registered in the trie, then ALGORITHM returns $s = t = 0$.

[ALGORITHM1]
begin
s← GET_LEAF (LT, *X*);
t← GET_LEAF (LT, *Y*);
if (*s* = fail or *t* = fail) then return $s = t = 0$;
if ((*t* ∈ RELATION(*s*) and R ∈ ATTRIBUTE (*s*, *t*)) then return *s* and *t*;
end;
(End of Algorithm)

Consider relation ("John", "speak", <SUB. -- VERB>). In this case, the followings are obtained:
s = GET_LEAF (LT, "John") = 33; *t* = GET_LEAF(LT, "speak") = 43
The 43 ∈LINK (33) and <SUB.--VERB> ∈ CONTENTS(33, 43); this relation <SUB.-- VERB> between nouns and verbs, for this reason if we find the same relation between another subjects and verbs we write relation <SUB.-VERB>, for example the words (" *Ahmed" ,"eat")* has the same relation.
Furthermore we find other relations as <VERB--OBJ.>, <VERB--PLACE>, <IS-- A>, <PART-- OF>, <SIMILARITY>, and <COMPOUND WORD>, we can apply the same algorithm. Moreover By using the information in this examples with <SUB. – VERB> relation:

1- Ahmed treat the illness 3 Ahmed cure the sick
2- Ahmed eat food
4- Ahmed speak with the nurse
5- Ahmed drink milk
6- John eat orange
7- John drink tea

8- John speak with his teacher

9- Cat eat food *10- Cat drink water*

and make the trie structure as Figure 1 with linking between leafs in this trie. Moreover,, we can make linking table 1 From this information as follows:

Table 1 Link Trie relation Information

X	s	RELATION(s)	ATTRIBUTE(s, t)
John	33	{2,18,8,29}	ATTRIBUTE(33,2)= <SUB. -- VERB > ATTRIBUTE(33,18)= <SUB. -- VERB> ATTRIBUTE(33,8)= <SUB. -- VERB> ATTRIBUTE(33,29)= <SUB. -- VERB>
Ahmed	13	{2,8,29,18,38}	ATTRIBUTE(13,2)= <SUB. -- VERB> ATTRIBUTE(13,8)= <SUB. -- VERB> ATTRIBUTE(13,18)= <SUB. -- VERB> ATTRIBUTE(13,38)= <SUB. -- VERB> ATTRIBUTE(13,29) = <SUB. -- VERB>
Cat	16	{2,29}	ATTRIBUTE(16,2) = <SUB. -- VERB> ATTRIBUTE(16,29) = <SUB. -- VERB>

and by using this automatic linking information, human can understand from this linking that the things which can eat and drink only and can not speak and buy (i.e. eatable & drinkable only) is Animals, the things that can eat ,buy, drink, and speak and can not treat sickness (i.e. buyable, speakable , eatable, drinkable only) is an antagonize (normal) human, and the man who can eat , drink, buy, speak and have the specialty to treat sickness (treatable) is a doctor. And we can create also this group as Figure 3

From this group we can extract the < IS -- A> hierarchy relation from this group that *Doctor is a human, and human is an animal.*

4.2 Automatic hierarchy from link trie information

If we pick up one information relation from link information Table 2. we can find the hierarchy from this information. Suppose that concept X on the hierarchy structure can be denoted by the concept code STRING (X), and concept Y on the hierarchy structure can be

denoted by the concept code STRING (Y), by using some example and make the trie structure as Figure 1, link trie as in Table 1, and algorithm 2, we can get the hierarchy of one relation <SUB. –VERB> hierarchy in Figure 2, and by the same manner we can apply in all relation. Now By this information we can prove how we can get the hierarchy, For example:

The hierarchical relationship between concepts X and Y can be determined by the comparison of STRING (X1) with STRING (X2) by using the LINK (s) from the table 3, and the following algorithm2. If we find the RELATION (s) of the word X1 including in the RELATION (s) of the word X2, this mean that the word X2 is super-concept of the word X1, and STRING (X2) is hierarchy of STRING (X1).

[ALGORITHM2]

begin
begin
s← GET_LEAF (LT, *X*);
t← GET_LEAF (LT, *Y*);

if (*s* = fail or *t* = fail) then return *s* = *t* = 0;
if ((t ∈ RELATION (*s*) and R ∈ ATTRIBUTE (*s*, *t*)) then return *s* and *t*;
then
begin
if (t ∈ RELATION (s1)) imply t ∈ RELATION (s2)
then
return STRING(X1) ⊃ STRING(X2);
end;
else
return GET_LEAF(LT,X) and GET_LEAF(LT,Y);
end;
end;

(End of Algorithm)

Example:

The word "*John*" with attribute '33' and '2, 18, 8, 29' ∈ RELATION (33), but RELATION (33) ⊂ RELATION (13) which contain '2,8,18,29,38' for the word "Ahmed", where *John is a human & Ahmed is a doctor*, so the STRING (Human) is hierarchy for the STRING (Doctor) and so on. As in Figure 3.

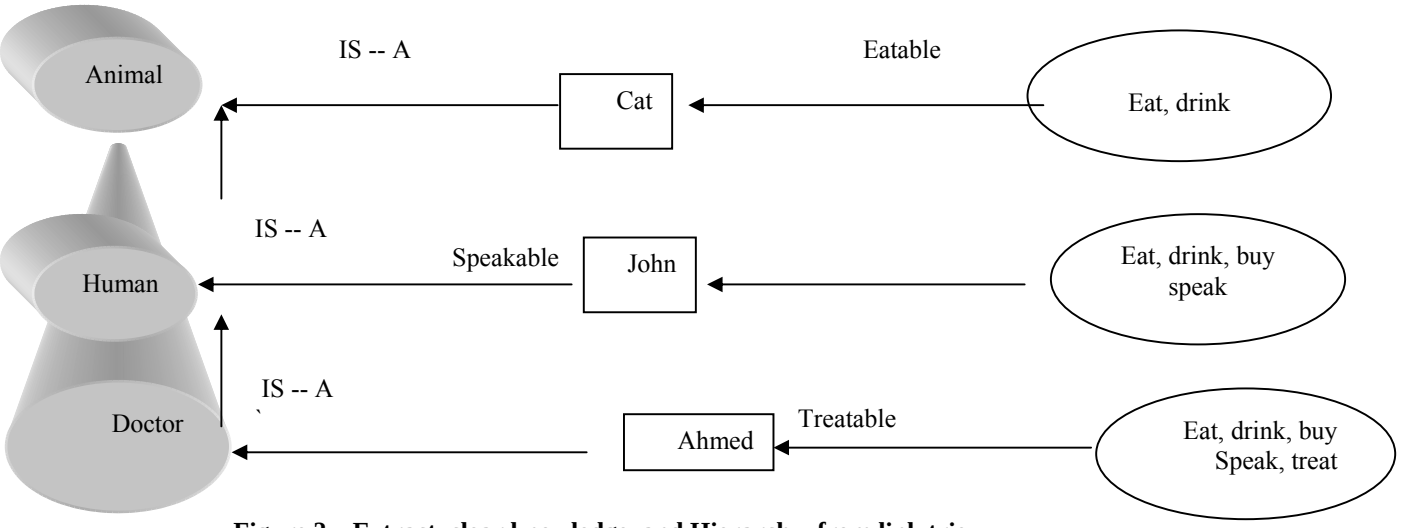

Figure 2 Extract clear knowledge and Hierarchy from link trie

5. Corpus Data & Simulation Results

5.1 Experimental Observation

99,714 sentences from tagged corpus (Pan TreeBank), which has different features, is involved in this experimental.

Data 1:

About 11,970 are used with subject-verb case relation, about 2,514 are used with the relation of verb-object, and 679 are used with verb-place. We could not pick them up enough because of depending on the number of objects and places. For example as in Table 8.

Data 2:

Using case frame and trie structure to introduce many relations among words as in section 2.

Data 3:

Using trie structure and linking trie between leaves for more information as in Figure 2, and Table 3.

Result 1]

By using Data 1,2,3. We can extract automatic generation of hierarchical relationships between words as in Figure 3 and Algorithm 2.

Result 2]

Similarity measurements can be measured between nouns and objects from the link tire information

Result 3].

By collecting this data and making relations between verbs and another kind of keys with link trie, we can show the hierarchy group. For example following Figure shows the linkage between subject and verb.

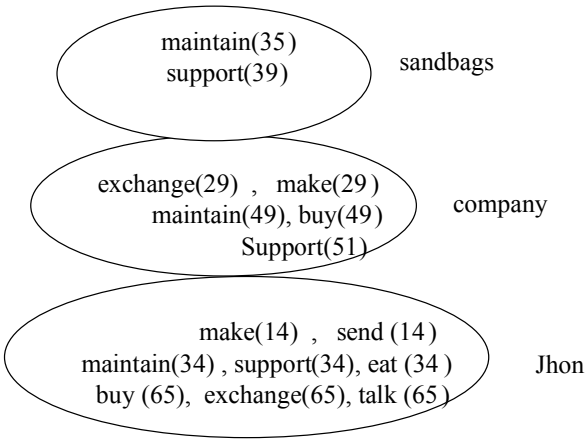

Figure 3. link group between Sbject and verb

By using this high-leaky this mean that ***sandbags*** supportable and maintainable but not buyable and exchangeable, ***company*** is buyable , exchangeable , supportable, maintainable , but not eatable ,talkable, but ***John*** can support, maintain, exchange, buy, eat, talk.

6. Conclusion

This paper proposed a method for acquiring deep cases of nouns from the information obtained by surface case

verbs and their possible substitutions, and showed that based on linkage between leaves, we could generate very basic and important information structures, as linkage groups and a table of high frequent access between verbs with subjects, verbs with objects, and verbs with places, which can be useful for knowledge generation and deep understanding of used database contents. This paper presented also an efficient data structure by introducing a trie that can define the linkage among leaves. The linkage enables us to share the basic words required for multi-attribute relations. By using this linkage, and a high frequent access between verbs with noun, we could extract an automatic generation of hierarchical relationships. The preliminary syntactic analysis can be achieved by many natural languages processing system; we will be able to obtain more precise semantic information from the syntactic resource. So the application of this will be included in our future program.

REFRENCES

[1] A.V. Aho, J. E. Hopcroft, and J. D. Ullman, "Date Structure and Algorithm," Addison-Wesley, Reading, Mass., pp. 163-169, 1983

[2] M. Ai-Suwaiyel and E. Horowitz, "Algorith for Trie Compaction," ACM Trans. IEICE, Vol. J76, D-II, No. 11, pp. 243-263, 1984

[3] J. Aoe, "An Efficient Digital Search Algorithm by Using a Double-array Structure", IEEE Trans. Software Eng., Vol. 15, No. 9, pp. 1066-1077, 1989

[4] J. Aoe, K. Morimoto and T. Sato, "An Efficient Implementation of Trie Structure," Software-Pract. & Expr. Vol. 22, No. 9, pp. 695-721, 1992

[5] J. Aoe, K. Morimoto, M. Shishibori and K. Park, "A Trie Compaction Algorithm for Large Set Keys", IEEE. Trans. on Knowledge and Date Eng., Vol. 8, No. 3, 1996

[6] J. Aoe, String Pattern Matching strategies, IEEE Computer Society Press, Los Alamitos,1994.

[7]E. Brill, "A Case Study in A Part of Speech Tagging", Computational Linguistics, Vol.21, No. 4, pp. 1-37, 1995.

[8] K. Dahlgren, Naive semantics for Natural Language Understanding, Kluwer Academic Publishers Norwell, MA, USA 1988.

[9] W. B. Frakes, Information Retrieval Data Structure &Algorithms, Prentice-Hal, USA,1992.

[10] E. Fredkin, "Trie Memory", Commun. ACM., Vol. 9, No. 2, pp. 490-500, 1960

[11] D. E. Knuth, "The Art of Computer Programming", Vol. 3, Sorting and Search, pp. 481-505, 1973

[12] F. Fukumoto, "Disambiguating preposition phrase attachment using statistical information" , NLPRS., Vol. 34, No. 2, pp. 752-757, 1995.

[13] Y. Jin, and Y. Tackkim, "Noun-sense Disambiguation from the Concept Base in MT" , NLPRS., Vol. 32, No. 2, pp. 357-362, 1995.

[14] J. Kupiec "A Robust Linguistic Approach For Question Answering Using An On-Line Encyclopedia, In proceedings of 16th ACM SIGIR international conference, pp. 181-190, 1993.

[15] H. Li., and N. Abe, "Clustering Words With the MDL Principle", Journal of Natural Language Processing, Vol. 4, No. 2, pp. 71-88, April 1997.

[16] K. Lim and M. Song, "Morphological Analysis with Adjacency Attributes and Word Dictionary", In proceedings of the international conference on computer processing of oriental language, pp. 263- 268, 1994.

[17] D. W. Loveland, Natural Language Parsing system, 1987

[18] K. Morita, H. Mochizuki, Y. Yamakawa and J. Aoe "An Efficient Retrieval Algorithm of Collocational Information Using Trie Structures" (in Japanese), Transactions of the IPSJ, Vol. 39, No. 9, pp. 2563-2571, 1998

[19] K. Morita, El-Sayed Atlam, M. Fuketa, K. Tsuda and J. Aoe, "Fast and compact updating algorithms of a double-array structure", Information Sciences, Vol.159, No.1-2, pp.53-67, 2004.

[20] M. Nagao, J. Tsujii, and J. Nakamura, "Machine Translation from Japanese to English", Vol. 74, No. 7, pp. 993-1012, 1986

[21] M. Oono, M. Fuketa, K. Morita, S. Kashiji and J. Aoe, "An Improvement Key Deletion Method for Double-Array Structure using Single-Nodes", Information Processing & Management, Vol.40, No.1, pp.47-63, 2004.

[22] M. Oono, El-Sayed Atlam, M. Fuketa, K. Morita and J. Aoe, "A Fast and Compact Elimination Method of Empty Elements from a Double-Array Structure", Software Practice and Experience, Vol.33, No.13, pp.1229-1249, 2003.

[23] A. Oishi, and Y. Matsumoto, " A Method for Deep Case Acquisition Based on surface Case Pattern Analysis", NLPRS., Vol. 34, No.2, pp. 678-684, 1995.

[24] T. A. Standish, Data Structure Techniques, 1981

[25] R. E. Tarjan and A. C. Yao, "Sorting a Sparse Table", Commun. ACM., Vol. 22, No. 11, pp. 606-611, 1979.

[26] T.Takenobu and I.Makoto, "Text Categorization Based on Weight Inverse Document Frequency", SIG-IPSJ, pp. 33-39, 1994.

[27] A. Utsumi, K. Hori, and S. Ohsuga "An Affective-Similarity-Based Method for Comprehending Attribution Metaphors", Journal of Natural Language Processing, Vol. 5, No. 3, pp. 3-30 July 1998.