Comparison between Proximity Operation and Dependency Operation in Japanese Full-Text Retrieval

Yasuaki Hyoudo, Kazuhiko Niimi, Takashi Ikeda
Dept. Information Science
Gifu University
www.ikd.info.gifu-u.ac.jp/{~hyodo,~kazuhiko,~ikeda}

Abstract In this paper we propose a full-text retrieval for Japanese document using dependency relation between words and evaluate it comparing to proximity operation. The proximity relation has been used as an approximation to syntactic or semantic relation because syntactic or semantic analysis with high accuracy has still been a high-cost task for a computer. We have developed the method of skeletal syntactic analysis for Japanese and apply it to the full-text retrieval. The figures of the index size, response time and accuracy of retrieval of our experiment show the effectiveness of the use of dependency relation.

1 Introduction

In this paper we propose a full-text retrieval for Japanese document using dependency relation between words and evaluate it comparing to proximity operation. The proximity relation has been used as an approximation to syntactic or semantic relation because syntactic or semantic analysis with high accuracy has still been a high-cost task for a computer.

Japanese is an agglutinative language and its basic linguistic unit is a Bunsetsu which is composed of a content word and succeeding function words. Japanese sentence is composed of a sequence of Bunsetsu. Modifier-modifiee relations(i.e. dependency relations) in Bunsetsu produce a basic semantics of a sentence. In information retrieval, modifier-modifiee relation, instead of a single word will allow additional specification and focusing of the concept to provide better precision and reduce the user's overhead of retrieving non-relevant items.

Japanese syntactic analysis is composed of a Bunsetsu analysis and a dependency analysys between Bunsetsu. In our paper[2], we have proposed a skeletal syntactic analysis for Japanese. The skeletal syntactic analysis is a technique to analyze skeletal structure of Japanese using only surface level information of a sentence. A skeletal structure is not necessarily a completely analyzed syntactic structure but a structure that may contain ambiguous or incompletely analyzed parts.

We apply our skeletal analysis to Japanese patent documents and build a full-text retrieval system in which a user can specify modifier-modifiee relations. The experiment shows the usefulness of dependency relations. Precision and recall rate is about 92% and 96% respectively, whereas in the case of using proximity relation it is about 84% and 81% (in the case of within 2 distance). Response time is even shorter than in the case of using proximity relation, index size is only 12% bigger.

2 Japanese Full-text Retrieval Using Dependency Relation

Figure 1 shows the outline of our retrieval system. The index is constructed on a patricia tree and is composed of word entries with pointers to the inversion lists on a secondary storage. An inversion list contains, along with the sentence position in the documents, the word position in the sentence and the position of its modifiee. For example, the word "A" is located at the fourth position and modifies the sixth word in the sentence "01001", the inversion list for "A" will include "01001/4/6". Inversion lists of the words in a query are read from secondary storage and expanded into hierarchical bit-tables.

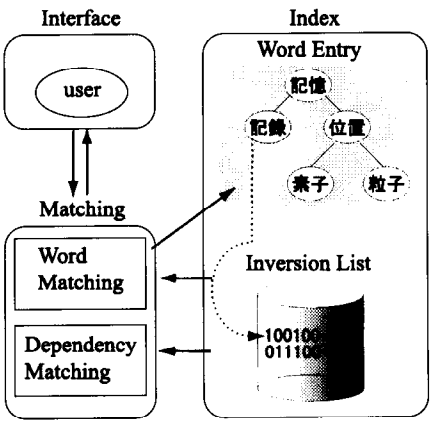

Figure 1: The Retrieval System

The retrieval is accomplished in two matching steps, word matching and dependency matching. Word matching is done very quickly by using hierarchical bit-tables[1]. Among the sentence collection of word matching, the system further proceed to investigate whether the required dependency relations can be found in it or not. The positions of modifiee candidates and that of modifier candidates are expanded into bit-tables respectively. The intersection of these two bit-tables are examined [Figure 2]. Dependency matching through bit-tables enables simple handling of multiple modifiees for a single modifier,

which sometimes occur in our skeletal analysis because it allows ambiguity.

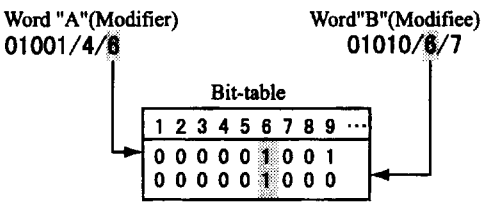

Figure 2: Dependency Matching

3 Performance Tests

For the evaluation, we have built a prototype system of full-text retrieval for Japanese patent documents. We discuss in this section the details of the index size, response time and accuracy of retrieval of our system comparing to proximity operation.

3.1 Index Size and Response Time

Table 1 shows the comparison of dependency operation to proximity operation in terms of average of the response time and the index size. (The system has been built on SunSparc 20 workstation with 64MB of main memory.)

Table 1: Average Response Time and Index Size

	Average response time	Index size
Boolean operation	19.10(ms)	3937(KB)
Proximity operation	22.56(ms)	5182(KB)
Dependency operation	21.45(ms)	6594(KB)

As shown in Table 1, dependency operation takes only 12% more than boolean operation in average response time, which is even less than proximity operation. The storage space required by the index for dependency operation is about 1.26 times as large as that for proximity operation and 1.67 times as large as that for boolean operation.

3.2 Accuracy of Retrieval

Table 2 shows the comparison between precisions and recalls for ten queries on patent documents.

Here precision and recall means the following.
(d.r. : dependency relation)

$$Precision = \frac{retrieved\ documents\ with\ the\ required\ d.r.}{documents\ retrieved}$$

$$Recall = \frac{retrieved\ documents\ with\ the\ required\ d.r.}{possible\ documents\ with\ the\ required\ d.r.}$$

As shown in Table 2, precision rates and recall rates for proximity operation are in trade-off relation. Precision and recall rate for dependency operation are 92% and 96% respectively. There is no proximity operation which exceed this figure at the same time. Table 3 is an example which shows the retrieval with dependency is superior to that of proximity.

Table 2: Accuracy of Retrieval

	Precision(%)	Recall(%)
Dependency operation	92.11	96.01
Proximity operation (within 1 distance)	96.12	74.25
Proximity operation (within 2 distance)	84.06	81.55
Proximity operation (within 3 distance)	75.92	87.28
Proximity operation (within 4 distance)	69.94	90.52
Proximity operation (within 5 distance)	65.56	93.51

Table 3: Comparison of the Retrieval
[(表面) → (形成):form something on a surface]

Sentence in the Document	Retrieved? P2	P3	D	Should be Retrieved
surface form film (表面)に(形成される)(皮膜)である […is a film formed on the surface]	Y	Y	Y	Yes
surface many uneven form glass plate (表面)に(多数)の(凹凸)が(形成された)(ガラス板) […a glass plate with many uneven surface]	N	Y	Y	Yes
surface oxidation latent image form (表面)を(酸化し)て(潜像)が(形成される) […the surface is oxidation and the latent image is formed]	N	Y	N	No
surface nailhead pattern simple low price form (表面)に(釘頭状)の(模様)を(簡単に)(安価に)(形成する) […nailhead pattern is simply formed on the surface with low price]	N	N	Y	Yes

P2:Proximity Operation within 2 distance
P3:Proximity Operation within 3 distance
D:Dependency Operation

4 Conclusion

We have developed a skeletal syntactic analysis method for Japanese and apply it to full-text retrieval system which allows a dependency operation. Our experiment on Japanese patent documents shows the superiority of dependency operation compared to proximity operation. Precision and recall are 92% and 96% respectively, which could not be attained by proximity operation. Response time is slightly less than proximity operation and index size is only 12% bigger than that of proximity relation.

References

[1] M. Fuketa, S. Mizobuchi, M. Shishibori, and J. Aoe. An efficient algorithm of retrieving example sentences from huge text data bases. *Information Processing Society of Japan*, 38(10):2004–2013, 1997.

[2] Y. Hyodo and T. Ikeda. Skeletal syntactic analysis of a long Japanese sentence based on surface information and N-neighborhood blocking technique. *Information Processing Society of Japan*, 36(9):2091–2101, 1995.