A Japanese KWIC Indexing System

SYUNSUKE UEMURA*

0. *Preface*

This report describes an experiment on Japanese KWIC indexing system. Major basic characteristics of the research are as follows.

(1) The system is perfectly Japanese language oriented. It can process *Kanzi, hiragana, katakana* and so on as its data. A kanzi-teletypewriter is used as an off-line IO device for a computer.

(2) A simple method of automatic segmentation has been developed, programmed and proved to be practical.

(3) Some topics are investigated concerning index format of Japanese sentences (vertical or horizontal line outputting, and their conversion from each other).

1. *Japanese character string*

Many kind of characters are used in Japanese sentences, *kanzi* (Chinese characters), *hiragana, katakana,* and even alphabets.

There is no custom of segmentation between words.

These special features cause much trouble in processing Japanese by computers. It is necessary to have an appropriate IO device and also to develope method of automatic segmentation (if not by hands).

2. *Segmentation for KWIC index*

To make Japanese KWIC index, we have to detect the positions of keywords (or phrases). Though this is a kind of segmentation the process here is rather simple. We don't have to segment Japanese sentences into separate words, but only the detection of character position from which the keyword begins (we call it 'keyword position') will be requested for indexing.

3. *Automatic segmentation*

The method we have developed here for automatic segmentation is based upon the analyses of character classes and transition from one class to another. The rules are roughly described as follows.

Suppose we have a Japanese title,

国産の COBOL コンパイラ研究

We can easily obtain a string of character classes from it, *kanzi, kanzi,*

This paper first appeared in Japanese in Joho-Shori (the Journal of the Information Processing Society of Japan), Vol. 10, No. 5 (1970), pp. 270-278.

* Software Division, Electrotechnical Laboratory, MITI, Tokyo.

hiragana, alphabet, . . . , *kanzi*.

(a) The head position of *kanzi* string has high possibility to be a keyword position.

国産の COBOL コンパイラ研究
↑ ↑

A long *kanzi* string usually consists of the series of 2 *kanzi* phrases.

情報|処理|研究

There are also some groups of *kanzi* that are used as prefixes or suffixes.

超|高速，計算|機

Any long *kanzi* strings can be divided into meaningful units by the rule above, and they show how we should segment *kanzi* strings (i.e. keyword positions).

超|高速|電子|計算|機

(b) Also the head of alphabetic string has high possibility to be a keyword position, especially when they consist of more than 1 character.

国産の COBOL コンパイラ研究
↑

(c) The head of *katakana* string is sure to be a keyword position in usual Japanese sentences.

国産の COBOL コンパイラ研究
↑

(d) The transition point to *hiragana* from others (*kanzi, katakana,* ……) has to be carefully examined, because it has high possibility to be *zyosi* or *zyodosi* which should not be a keyword. Japanese titles of 148 scientific reports were investigated and the number of combination of actually used *zyosi* and *zyodosi* was proved to be less than 30. Some of them are:

に，による，と，としての，の，のための，が，を，および

The rest has some posibility to be keywords.

These rules require only small tables for segmentation. Processing speed is expected to be much shorter in comparison with longest-matching method, and the experiment proved it.

4. *Page formatting*

Since this indexing system is for Japanese, not only usual horizontal output form but also vertical line outputting has been designed and programmed. Transformation from outputting in lateral lines into vertical style (or vice versa) requires some character relacements. '(' or ')' in horizontal style should be '⏜' or '⏝' in vertical style, and so on. Fig. 1 (shown next) is an example of vertical style KWIC index.

5. *Sorting*

Sorting of key phrases is based upon kanzi-teletypewriter code of each

Fig. 1. A Sample Output.

character, which usually reflects its '*on-yomi*' order of *kanzi*. As is shown in Fig. 1, the result is almost natural but strange at some points.

6. *Experiments*

A program was written using COBOL and experimental KWIC index was obtained successfully. A sample page is shown in Fig. 1.

Input paper tape is prepared manually by kanzi-teletypewriter and computer output is punched on paper tape. Kanzi-teletypewriter prints it with average speed 2 characters/second.

Only 2 errors out of 1031 KWIC lines was found in the result of our automatic segmentation method. The indexing required 3.5 min for the computer (FACOM 230-50).

7. *Conclusion*

The proposed segmentation method is considered to be simple and practical. So far as the titles of technical papers concern, the method has enough accuracy for KWIC indexing. Further investigation for automatic arrangement (sorting) of Japanese words will be necessary to get more conventional indices.