PRACTICAL TECHNIQUES OF CHINESE AUTOMATIC WORD SEGMENTATION IN AN APPLIED SYSTEM - CASS

Chun-yu KIT

Department of Applied Linguistics
City Polytechnic of Hong Kong
Fax:7888706 Phone:7888489 E-mail:ALCYKIT@CPHKVX.BITNET

ABSTRACT

It has become obvious in recent years that automatic segmentation of Chinese character string into words is a key issue in Chinese Information Processing. It is regarded as another bottleneck besides Chinese Character Coding to be resolved. Therefore, it is now a common research topic in China (Mainland, Taiwan and Hong Kong) and overseas wherever Chinese computing is undertaken. The main purpose of this paper is to demonstrate general principles of an applied system of Chinese automatic word segmentation through introducing the design and implementation of the CASS system. The overall architecture of the system, segmenting algorithm, dictionary construction, recognition and handling of ambiguous segments of character string will all be discussed.

I. Introduction

Chinese automatic word segmentation, also known as automatic word identification, is a crucial issue to overcome in Chinese Information Processing. It aims at recognizing words, including idioms and terminology, in written Chinese text. Without this essential step, Chinese computing can not proceed to further processing, e.g. parsing, after the Chinese characters have been input into computer, because it is on the word level that the computing of natural language is done.

Since this issue arose, it has been attacked from many necessary sides. At the technical side, there are a number of difficult problems to overcome in implementing segmentation methods and handling ambiguous segments of character string. An ambiguous segment here refers to a part of a character string which can be divided into at least two different groups of words. Techniques for building segmenting system can be classified into two different levels: mechanical segmentation and knowledge processing. The former carries out basic segmentation methods and necessary support, dictionary construction for example. The latter, which employed many practical AI techniques applicable to Chinese computing, checks and corrects mistaken segmentations. Making use of knowledge of language has become more and more important in improving the accuracy of word segmentation.

These two kinds of practical techniques are urgently needed in CIP today. The CASS system is presented in this paper as an illustration of how these techniques may be employed. The abbreviation CASS stands for Chinese Automatic Segmenting System.

II. Objectives and Overall Architecture of CASS

CASS system was implemented as the central part of a large-scale state project in China to build up several Chinese word banks in the domain of economics. Word selection is based, in part, on the usage frequency in a corpus of economic literatures composed of more than ten million Chinese characters. Word segmentation and frequency calculation was assigned to the CASS system.

CASS was carried out using VS FORTRAN on an IBM-4361 (1988, Beijing, China). It provides two execution modes. One, the batch processing, is the usual way to process text data in large amounts. The other is usually used to deal with small texts in interaction with a human user, in order to improve the accuracy. The interactive mode is also used to obtain special experimental data about Chinese word segmentation.

By presetting parameters, the user can also maks choice on use of interaction and knowledge processing. CASS brings out all machanical methods of word segmentation simultaneously. Users can choose any of them. It is probably one of the most distinguishing features of the system and may be regarded as a new progress in word segmentation of Chinese that all segmenting methods (according to ASM(d,a,m) model, see next sections), no matter forward scanning or backward scanning, are carried out with a dictionary arranged in normal sequence.

The dictionary used by CASS is organized within a special structure, known as first-character-index structure. It is stored as a VSAM KKDS file to improve access speed and decrease the memory and disk space used. With its support, mechanical segmentation in CASS reaches an average speed of 200 characters per second in test runs.

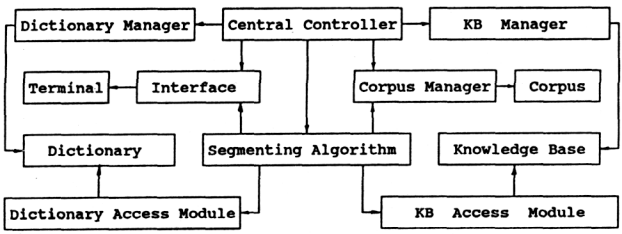

Figure 1

CASS system is composed, shown as Figure 1, of five functional components: 1.The Central Controller that coordinates operations of the whole system; 2.The Segmenting Program, which contains three individual programs: Segmenting Algorithm, Dictionary Access Module and Knowledge Base Access Module; 3. Utility programs, including Dictionary Manager, KB Manager, Corpus Manager and a small interface used in interaction between CASS and its users; 4. Dictionary; 5. Knowledge base for word segmentation.

The processing flow of CASS is shown in figure 2, in which A and B indicate interactive processing and batch processing modes respectively, and 1 2 3 and 4 indicate typical flows for the four different segmenting modes that users can choose one each time. Segmenting method and output options are also selected by users at presetting. If the word frequency calculation program needs

to be activated to update the data in dictionary, a password must be supplied, otherwise the output of segmentation will be displayed on the screen or stored in sequential files, depending on indication of previous settings.

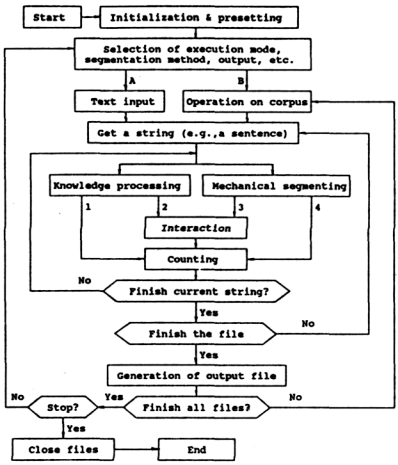

Figure 2

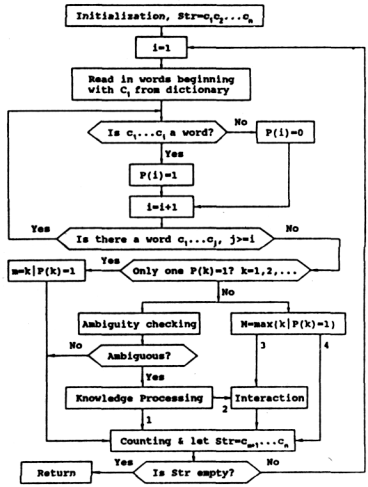

Figure 3

III. Selection & Implementation of Segmentation Methods in CASS

Since the issue of word segmentation arose in CIP, a number of segmentation methods[5] [7] have been proposed. In fact, all methods of Chinese automatic word segmentation are essentially based on character string matching. A method of segmentation is therefore determined, as we found, by three factors: first, the direction of scanning, which may be forward or backward; second, the change of the number of characters in the matched string through each round of matches, i.e., adding or omitting characters one by one to get through the matching process; third, the selection of result from maximum match or minimum match. A round of matches means all matches carried out in identifying a word. While the word is determined, another round of matches begins if the segmentation is still going on. By the way, of maximum match and minimum match, the latter is not really suitable to contemporary Chinese, because almost every character in the written form of this language makes a word or a single-used morpheme.

Consequently, a systematic structure model ASM(d,a,m), [7] [8] in which ASM refers to automatic segmenting methods, has been suggested to classify, and present as well, all methods of Chinese automatic word segmentation. The three parameters are defined as follows:

dϵD={+1,-1}, +1 refers to forward scanning,
-1 the backward;
aϵA={+1,-1}, +1 means adding charactesr in a round of matches,
-1 omitting;
mϵM={+1,-1}, +1 is maximum match,
-1 the minimum.

Therefore, the Maximum Match Method, commonly known as the MM method, which has the properties of forward scanning, character omitting and maximum match, can be represented within this model as MM=ASM(+1,-1,+1). Similarly, the Backward (or Reverse) Maximum Match Method can be described as BMM=ASM(-1,-1,+1). Based on this model, we have brought forward other two new methods ASM(+1,+1,+1) and ASM(-1,+1,+1), which prove to be most beneficial to the whole process of word segmentation.

Among maximum match methods, those with character omission, where characters in the matched string are cut away one by one from one end till a word is found, can not accumulate information about the structure of the output string. So they do not provide data for the follow-up testing and correction of incorrect segmentations. In contrast, methods with character addition, where the number of characters in the matched string is increased one by one, allow information about the structure of the output string to be retained and later used to test and correct mistaken segmentations in knowledge processing.

Today, technical development of Chinese word segmentation focuses more and more on increasing segmentation accuracy, and it depends almost wholly upon knowledge processing. In view of this consideration, we chose ASM(+1,+1,+1) as the kernel method for segmenting algorithm in CASS system. Other methods with maximum match can be carried out using this algorithm. Its flow chart is shown in Figure 3.

IV. Construction and Features of the Dictionary

A machine dictionary for automatic word segmentation is usually organized into the first-character-index structure, which is generally composed of two component parts, the index and the data. The index can be wholely put into memory, but the data area that stores almost all words and regularly-used phrases of a language is so huge that it must be located on a mass storage device, e.g. hard disk, preferably within the format of a direct access file, like a relative record file, to allow a very fast access speed.

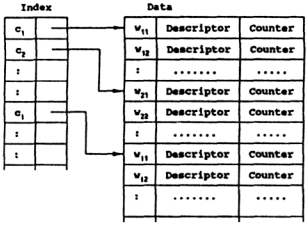

Figure 4

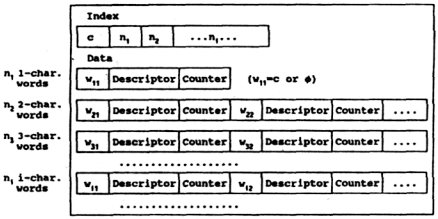

Figure 5

The first-character-index structure is shown as Figure 4. Each record in the index consists of a character, i.e. the first-character, and one (or more than one) pointer which indicates the location in the data area of words beginning with that character. As for how many words are put into a record in the data area and how words with the same beginning character but different number of characters, i.e. word length, are arranged, what can be done on these aspects depends absolutely upon the development environment. But two critical factor, i.e. space requirement and access

time, both must be minimized. That ought to be a basic principle to be observed in dictionary design.

The dictionary in CASS also employed first-character-index structure, constructed within a VSAM KSDS file which has a variable record length. First-character is chosen as the key value of a record for retrieval. The content of a record includes the index data and all words beginning with the same character. Words are arranged according to word length and ascending sequence of EBCDIC code, the internal code of IBM main frame for Chinese characters. The main function of the index is not to indicate where words are placed in the data area, but to help read/write data from/to a record in as fewer times of access as possible. Structure of the dictionary is shown in Figure 5, each record is made of an index field, which contains a first-character and a series of numbers of words with different length, and data fields, each of which holds a word (minus first-character), an counter storing frequency and a descriptor with items of descriptive information.

A dictionary structured like the above one has a number of properties very beneficial to automatic word segmentation:

1. Powerful support to segmenting algorithm to reach a rapid speed. The fact that a record can be read into the memory with a single read operation means that only one dictionary access is necessary to recognize a word.

2. Minimum use of memory and disk space. The memory only holds index data pertaining to one first-character at a time, instead of data of the whole index area. In the file storage, words with equal length take equal space:

words of 1 character each take 2 bytes,
words of 2 characters each take 4 bytes,
.....
words of i characters each take 2i bytes, etc.

The space complexity remains as low as possible.

3. Easy and simple access. You can get a whole record you need into the memory by direct access, and then divide it into single words according to the index data:

n_1 word(s) of 1 character,
n_2 word(s) of 2 characters,
.....
n_i word(s) of i characters, etc.

And generation of the whole dictionary is also very simple.

4. Perfect adaptability. The dictionary does not set any limits on word length. And all operations on words of different length are uniform. This is a feature very important for a specialized word bank which collects words and terms pertaining to a specific domain. Some terms of this kind may consist of as many as two or three dozen Chinese characters. If the ability of the dictionary is not adequate to perform manipulations on them, the performance of the whole system would suffer significantly. With ideal adaptability of this dictionary, CASS is able to handle words of arbitrary length.

At present, there are 136,000 words, including near 20,000 terms from fields of economics, collected in the dictionary. It is believed by the author to be the largest Chinese dictionary in the world.

The dictionary is managed by the dictionary manager, which sorts words into a particular sequence, generates index and data, adds/deletes words and transfers words to the main dictionary from a temporary one, etc. The dictionary manager takes about ten minutes generating the whole dictionary. The speed is rather satisfactory.

There is another kind of structure for building a machine dictionary for word segmentation of Chinese, which has proved through experiments in developing the CASS to be worthy of serious recommendation. In this structure, the first-character-index is also employed, but words with the same first character are organized differently. To illustrate, consider the following words beginning with character " 或 ":

或	或然	或然率	或然判斷
或然性	或然性推理	或體	或許
或則	或者		

These words can be represented effectively as a tree structure, coded in the following way:

或 [然 [率 , 判 (斷) , 性 [推 (理)]] , 體 , 許 , 則 , 者]

Among delimiters used above, '[' and ',' indicate that 'it make a word from the beginning to here'. This structure requires even less storage, since common initial characters are not duplicated, though each word needs at least one delimiter. Most important, this structure allows a great simplification of segmenting algorithm.

V. Recognition and Handling of Ambiguities

It has been well known that ambiguities in Chinese automatic word segmentation can be divided into two types:

1. Embracing: in a sequence of Chinese characters $S=a_1...a_i b_1...b_j$, if $a_1...a_i$, $b_1...b_j$ and S each make a word, then S is known as an embracing ambiguous segment. The segment S that itself is a word embraces some other words into which it can be directly divided.

2. Overlapping: in a sequence of Chinese characters $S=a_1...a_i b_1...b_j c_1...c_k$, if section $a_1...a_i b_1...b_j$ and $b_1...b_j c_1...c_k$ each make a word, then S is defined as an overlapping ambiguous segment, and $b_1...b_j$ is known as the overlap. Usually, the overlap has a length of 1 character.

In CASS, there is a unique program, named Ambiguity Finder (AF), which is designed to detect ambiguous segments. It is, in fact, the Segmenting Program (SP) running with a different control strategy. While working, AF calls the SP to gain a group of return values. According to these values, AF may call SP again with other initial values to obtain further information, or make adjustments if the relevant information is already adequate. SP is called by the procedure Seg(Str,P) in CASS. Str is the character string to be segmented, supposed $Str=C_1C_2...C_i...C_n$, and P is an array carrying the return values. An element of P can only take a value between 1(True) and 0(Faulse). P(i)=1 (i=1,2,...m, soppose the maximum word length with current first character is m) means that the character string $C_1C_2...C_i$ makes a word. Otherwise, there is no such word in dictionary. For instance, while Str=" 結 合 成 分 子 ", working process of AF is as follows:

1st call of $Seg(Str_1,P)$: Str_1=" 結 合 成 分 子 "

Return values

P(1)=1, P(2)=1,
P(3)=0, ..., P(m)=0

Thus, we have

$j_1=max(i|P(i)=1,i=1,2,...m)=2$
$k_1=max(i|P(i)=1,i=1,2,...j_1-1)=1$

2nd call of $Seg(Str_2,P)$: $Str_2=Str_1(k_1+1:n)$= " 合 成 分 子 "

Return values

$P(1)=1,\ P(2)=1,$
$P(3)=0,\ \ldots,\ P(m)=0$

Thus, we have

$j_2=\max\{i \mid P(i)=1, i=1,2,\ldots m\}=2$
$k_2=\max\{i \mid P(i)=1, i=1,2,\ldots j_2-1\}=1$

Adjustment:

$\because j_2>j_1-k_1 \ \therefore$ " 結 合 成 " is an overlapping ambiguous segment.

3rd call of Seg(Str_3,P): $Str_3=Str_2(k_2+1:n)=$ " 成 分 子 "

Return values

$P(1)=1,\ P(2)=1,$
$P(3)=0,\ \ldots,\ P(m)=0$

Thus, we have

$j_3=\max\{i \mid P(i)=1, i=1,2,\ldots m\}=2$
$k_3=\max\{i \mid P(i)=1, i=1,2,\ldots j_3-1\}=1$

Adjustment:

$\because j_3>j_2-k_2 \ \therefore$ " 合 成 分 " is an overlapping ambiguous segment.

What the above j and k with subscripts exactly mean can be seen clearly in Figure 6. It is obvious that calling SP strategically in this way can detect all overlapping ambiguous segments of character string with any overlaps.

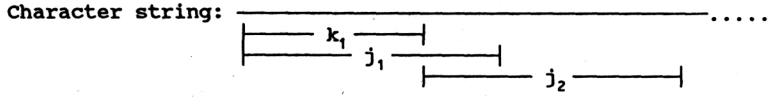

Figure 6

And besides, the AF can advoid meaningless manipulation on an unambiguous segment which contains an overlap. For instance, the string " 或 然 性 推 理 由 于 ... ", which is really a segment of this kind, is often distinguished by some other ambiguity checking as an overlapping ambiguous segment.[3] [4] But as a contrast, AF leads to the right end:

1st call of Seg(Str_1,P): $Str_1=$ " 或 然 性 推 理 由 于 ... "

Return values

$P(1)=1,\ P(2)=1,\ P(3)=1,\ P(5)=1$
$P(4)=0,\ P(6)=0,\ \ldots,\ P(m)=0$

Thus, we have

$j_1=\max\{i \mid P(i)=1, i=1,2,\ldots m\}=5$
$k_1=\max\{i \mid P(i)=1, i=1,2,\ldots j_1-1\}=3$

2nd call of Seg(Str_2,P): $Str_2=Str_1(k_1+1:n)=$ " 推 理 由 于 ... "

Return values

$P(1)=1,\ P(2)=1,$
$P(3)=0,\ \ldots,\ P(m)=0$

Thus, we have

$j_2=\max\{i \mid P(i)=1, i=1,2,\ldots m\}=2$
$k_2=\max\{i \mid P(i)=1, i=1,2,\ldots j_2-1\}=1$

Adjustment:

$\because j_2=j_1-k_1 \ \therefore$ " 或 然 性 推 理 " is an embracing ambiguous segment, not an overlapping ambiguous segment.

Thus, according to above return values and adjustment, it becomes so clear and easy to get the correct result.

CASS provides three different ways to handle the identified ambiguities: human-machine interaction, knowledge processing and

mechanical segmentation. The knowledge base to be used in knowledge processing is now being developed, and it is expected to be introduced in another paper before long. With mechanical segmentation, we can assign overlaps of overlapping ambiguous segments to their preceding sections or the following sections in order to achieve a better result. It was reported that assigning overlaps to their following sections would gain a higher accuracy in word segmentation of contemporary written Chinese.[2][4]

VI. Realization of Multiple Segmentation Methods

It can be seen that different methods of segmentation with maximum match reach the same result in dealing with embracing ambiguities. They take the longest character string in successful match with the dictionary as the output and ignore others. But in handling overlapping ambiguities, they differ from each other in assigning overlaps. Methods scanning forward, e.g. the MM method, will connect an overlap to its preceding section to form a word. For example, character string $a_1 \ldots a_i b_1 \ldots b_j$ from the overlapping segment S cited above will be taken as the output. However, methods scanning backward will assign an overlap to its following section and thus obtain a word in form of $b_1 \ldots b_j c_1 \ldots c_k$.

Up to now, we can see that, apart from their difference in influence upon segmenting speed and in keeping information about the internal structure of output string, methods of segmentation distinguish from each other characteristically in their treatment of overlaps while handling overlapping ambiguities. Different treatments of overlaps will yield results of different segmenting methods. CASS integrates multiple methods of segmentation into one system using a dictionary in a normal ascending order by allowing selections of resolving overlapping ambiguities.

These methods of segmentation are also realized by calling SP with the procedure Seg(Str,P) in CASS, working together with AF. After calling Seg(Str,P) one or several times, AF determines whether or not an overlapping ambiguous segment is met. If yes, the overlap can also be decided. And then you can make your preferable assignment of it to obtain the result of any segmenting method you like. In CASS, users can preset initial values to control the system to output the result of MM method or BMM method, or others. Of course you can also obtain results of all segmentation methods simultaneously. This is of very important significance in comparing properties of different segmenting methods in the research of Chinese automatic word segmentation.

So it is clear that CASS integrates all practical methods of word segmentation. It is also believed to be a first example in Chines Information Processing.

VII. Conclusion

We have spent almost two years in developing CASS. Today, it is ready for practical use, though the knowledge base is still weak and being expanded. According to the statistical data gained in test runs, the error rate for segmentation is 1/300, which means that only one incorrect segmentation occurs while scanning through 300 Chinese characters. That indicates the rate of correct word identification reaches as high as 98 per cent.

We also found in the test runs that the vast majority of segmentation errors could be corrected if some simple relevant rules were used. We therefore believe, and also expect, that at

least half of mistaken segmentations can be handled properly once the knowledge base is fully implemented. Our target next stage is to lower the error rate to 1/800 or less by knowledge processing, though we will face and overcome a number of difficulties.

Finally, from CASS system we can see the follows: i) Segmentation methods with the property of character addition in matching process are most beneficial to the detection of ambiguities and correction of mistaken segmentation with knowledge processing; ii) Differences between segmentation methods lies in their different ways of handling the overlaps of overlapping ambiguous segments; iii) An applied system of Chinese automatic word segmentation is basically composed of a few functional modules, e.g. a segmenting program, a dictionary, some utility programs, a knowledge base, etc., though some of them may not be physically independent.

Acknowledgements

The author would like to thank Prof. LIU Yuan and LIANG Nan-yuan of Department of Computer Science, Beijing University of Aeronautics and Astronautics, for their giving constructive advice. The author also greatly appreciate the aid from Miss Laura Proctor of Department of Applied Linguistics, City Polytechnic of Hong Kong.

Main References

[1] LIU, Y. & LIANG, N.Y., OM Method of Automatic Word Segmentation, Chinese Information Processing, 1985, No.2.
[2] LIU, Y. & LIANG, N.Y., Basic Engineering for Chinese Processing - Modern Chinese Words Frequency Count,Journal of Chinese Information Processing, 1986, No.1.
[3] LIANG, N.Y.& LIU, Y., Automatic Word Segmentation of Written Chines, Chinese Information Processing, 1986, No.1.
[4] LIANG, N.Y., Chinese Automatic Word Segmentation System CDWS, Journal of Chinese Information Processing, 1987, No.2.
[5] Liang, N.Y., A Survey of Automatic Segmentation of Written Chinese, Computer Application and Software, 1987, No.3.
[6] FAN, C.K. & TSAI, W.H., Automatic Word Identification in Chinese Sentences by the Relaxation Technique, Computer Processing of Chinese & Oriented Languages, 1988, V.4, No.1.
[7] JIE(=KIT), C.Y., LIU, Y.& LIANG, N.Y., On Methods of Chinese Automatic Word Segmentation, Journal of Chinese Information Processing, 1989, No.1.
[8] JIE(=KIT), C.Y., A Systematic Structure Model for Methods of Chinese Automatic Word Segmentation and Their Evaluation, Proceedings of Chinese Computing Conference'89, Singapore.