1998 International Computer Symposium
Workshop on Software Engineering and Database Systems
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Signature-Based Chinese Character Retrieval Schemes Using DAYI Input Method

Chin-Chen Chang and Kuo-Lung Hong

Department of Computer Science and Information Engineering,
National Chung Cheng Univerisity, Chiayi, Taiwan 621, R.O.C.
Email:{ccc,jackhong}@cs.ccu.edu.tw

ABSTRACT

In this paper, we propose two new Chinese character retrieval schemes using the DAYI input method. Our schemes are based on signature files, which have always been used for text retrieval. These schemes are economical in storage space use and are very efficient when dealing with large databases. Moreover, the implementation of wild-character function becomes easy and quick. The problem one of our schemes has to face is that it produces false drops. That is, some DAYI codes indicating Chinese characters may in fact come out to find no characters to them at all. We argue that this is not a big fault in real applications. On the other hand, we also propose an alternative method to retrieve Chinese characters, which efficiently solves the false drop problem but takes more space and access time. In this paper, we make an experiment for it, a comparison between traditional method and the methods we propose has been made to show the superiority of our schemes.

1. Introduction

The research and development of Chinese language computing capabilities have been explored for many years. The main difficulty of this problem is to access Chinese characters from its vocabulary rapidly and efficiently. Several methods [1-3,6,7] for Chinese character retrieval have been contrived. Most of them can be classified into two major categories – input by phonetic spellings and input by ideographs. Being one of the most popular methods by ideographs, DAYI is a breakthrough in the field of Chinese character input methods for the reason that it is not only the simplest input method but a fast one of its kind.

In the DAYI system, each Chinese character is synthesized by combining the DAYI roots from one to four. There are forty clusters in the DAYI root sets. Each cluster, associated with a keystroke on the keyboard, is listed in Table 1. As we know, an input method itself and its implementation are two totally different issues. We may have the best-input method but have the lousiest implementation. Therefore, the main consideration for an implementation is the efficiency to access a Chinese character from its vocabulary.

We propose a scheme here to retrieve Chinese characters using the DAYI input method, which is based on a signature file. As we will see, it is economical in using the storage space and is efficient in dealing with large databases. The disadvantage of this scheme is that it produces false drops. This phenomenon comes from the use of signature files. It is the cost we pay to make the signature file efficient. However, having the idea of the DAYI input method in mind, we argue that it is not a big fault when false drop rate is low because there are also collisions in the original DAYI input method. The increase of collisions is very little when the false drop probability is small. Even

cluster root	(1) 力	(2) 方	(3) 竹	(4) 金	(5) 言	(6) 牛	(7) 目	(8) 四	(9) 王	(10) 車	(11) 田	(12) 八	(13) 止	(14) 虫	(15) 人	(16) 馬	(17) 七	(18) 日	(19) 一	(20) 土
Keystroke	,	.	/	0	1	2	3	4	5	6	7	8	9	;	A	B	C	D	E	F
Cluster root	(21) 手	(22) 鳥	(23) 木	(24) 月	(25) 之	(26) 女	(27) 雨	(28) 刀	(29) 口	(30) 耳	(31) 石	(32) 工	(33) 十	(34) 小	(35) 廾	(36) 大	(37) 山	(38) 水	(39) 火	(40) 心
Keystroke	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

Table 1. Forty DAYI root clusters associated with the keystrokes on the keyboard

when additional collisions occur, the users may not seem to feel too much more uncomfortable about it. Therefore, considering the trade-off between the efficiency of accessing time and the false drops, we think that our scheme has really pushed the DAYI input method a step ahead.

There is another advantage when we use the signature file technique to implement DAYI input method. That is, the implementation of wild-character function becomes simple and quick. The wild character is a special character that can substitute any keystroke of DAYI codes. Whenever the user can not spell the whole DAYI code, she/he can use the wild character to get the correct Chinese character that she/he wants. So the wild-character function is powerful in real applications. In the next section, we shall introduce how our scheme implements the wild-character function.

In the remainder of this paper, we shall review the signature file technique based on the superimposed coding method in Section 2. In Section 3, we shall propose the Chinese character retrieval scheme based on a signature file and explore its storage structure. In Section 4, an alternative solution without any false drop is proposed. In Section 5, we shall provide an experiment to analyze the performance of various algorithms. Finally, in the last section, we shall make a conclusion.

2. Preliminaries on the Signature File Technique

The signature file has been proven to be one of the most efficient access methods for text retrieval. Since it can deal with unformatted data, many application domains have shown their interest in signature file techniques, e.g., office information systems, statistical and logic databases. It acts as a search filter to reduce the search space in the primary database and is therefore very suitable for large database [5].

In general, the signature file access method can be taken into consideration for partial match retrieval whenever the object is characterized by a set of keywords. In the signature file access method, each object is associated with an object signature produced from the transformation of the associated keywords of an object. The transformation of a keyword is called a word signature, and an object signature is combined with word signatures. A collection of object signatures is called a signature file. The query described by a set of specified keywords is also transformed to be a query signature with the same method as the object signature generation. The object whose signature seems to be qualified but actually unqualified is called a false drop. A high false drop rate causes unnecessary disk access and lowers the overall system performance [8].

A great amount of effort has been devoted to the study of optimal algorithms to generate signatures from objects as well as to the estimation of false drop probability [4]. The basic types of signature extraction methods include Word Signature, Superimposed Coding, Bit-block Compression, and Run Length Compression [10]. Among them, Superimposed Coding is the most popular one, and that is also why it is used in this paper. In Superimposed Coding, each keyword yields a bit pattern, namely, a word signature, of size F where M bits have value "1" while the others have value "0". These bit patterns are OR-ed together to form the object signature. The number of 1s in a signature S is called the signature weight and is denoted as w(S). If an object signature contains 1s in the same bit positions as the query signature does, then the object signature qualifies the query signature. Let S_i denote the i-th record signature and S_Q the query signature. Formally, the set of these qualified signatures is defined as $\{S_i|(S_i \cap S_Q)=S_Q\}$. The time required to compare two signatures is very short, especially for query signatures with low weights.

Even though the false drop rate is an important measure for the comparison of different signature extraction methods, the performance of signature filters depends mainly on the I/O cost. However, the efficiency of filtering is determined by the storage structures that support the filtering process. Quite some heavy work has been done to improve the speed of signature search so far to organize the signature file structure. The simplest signature method stores the signatures one by one and sequentially. However, searching the signature file itself may be slow. This is because the size of the signature file is proportional to that of the database.

One approach, called the Bit-Slice representation, improves query response time by reducing the number of bits that have to be retrieved from the file of the signatures at query time. This storage technique has been used in [9]. The main idea of this structure is to store the columns, rather than the rows of the signature table, together and provide direct access to the columns. The advantage of such an arrangement is that, when answering a query, only w(Q) columns must be accessed. However, increasing query weights requires additional costs. The maintenance of bit slices is extremely time consuming. For this reason, the Bit-Slice method is only suitable for stable files and queries with low weights.

3. A Signature-based Scheme for the DAYI Input Method.

3.1 The Basic Idea

In the previous section, we have introduced the signature file technique. We know that signature files are very efficient in space use and access time for text retrieval. The retrieval of Chinese characters using DAYI input method is also a kind of text retrieval, and this is the basic idea when we use the signature file technique to implement the DAYI input method. There are forty clusters in the DAYI root sets, where each cluster is a representative associated with a keystroke. Each Chinese input code for Chinese characters is combining the keystrokes from one to four. We regard such a keystroke as a keyword of a signature file. The object signature of each Chinese character is produced from the transformation of its corresponding keywords. We call this scheme the *DAYI Signature Superimposed Coding* (DSSC).

With this approach, there are several important things we have to consider. First, because there is no order among the keywords in the signature file, this approach will regard the same keywords differently ordered as the same Chinese input code. Therefore, many DAYI codes originally indicating some Chinese Characters may in fact find no characters to them. That is, the false drop rate is much higher. For example, the DAYI code of "鷹" is "CH" rather than "HC". However, when we use the signature file technique to encode these two input codes, they will both produce the same result. In order to solve this problem, we encode these forty keystrokes representing root clusters with their corresponding positions in the DAYI code. For instance, After encoding, the new keywords of the Chinese character "鷹" are "C_1","H_2". Thus, the new keywords of the input code "HC" will be "H_1","C_2". Hence, we can distinguish these input codes with different keywords and make the false drop rate lower.

Except for the ordering problem, there is another issue having to do with a higher false drop rate, too. This problem can be illustrated by the following example. The character "日", whose DAYI code is "D", will be encoded with the keyword "D_1". The character "旦", whose DAYI code is "DE", will be encoded with the keywords "D_1","E_2". When the query input is "日", the characters "日" and "旦" will both be qualified objects. Therefore, it produces a false drop. In order to solve this problem, we divide the DAYI input codes into four types according to their lengths and denote them as "$\$_1$","$\$_2$","$\$_3$" and "$\$_4$". When producing an object signature for a Chinese character, we regard its corresponding type as a keyword and add it to the keywords of the object. As the previous example, the keywords of the character "日" will now be encoded with "D_1" and "$\$_1$", and the keywords of the character "旦" will be encoded with "D_1","E_2" and "$\$_2$". Hence, when the query input is "日", the Chinese character "旦" will no more be the qualified object.

As mentioned before, The number of the keyword types in our scheme is 40*4+4=164. This amount is much fewer than that of an ordinary signature file. For this reason, when producing the word signature of a keyword, it is no longer necessary to use hash functions. All we need is to map the keyword to a certain number when producing a word signature. This approach will reduce the probability of false drops. The Algorithm, illustrated in the following, will describe how a keyword is transferred to a word signature.

Algorithm: Mapping a given keyword to a width-F, weight-M word signature

```
A1 : W = 0, SIG[I] = '0', for 1 ≤ I ≤ F;
A2 : Directly map the keyword to a certain number , then
     assigned to KEY.
A3 : Resume random number generator with KEY
A4 : I = random() mod F
A5: if SIG[I]='1' goto A4
A6: SIG[I] = '1' , W = W + 1;
A7: if W< M goto A4
A8 : Return SIG[I] , for 1 ≤  I ≤ F;
```

3.2 Storage Structure

The storage structure of signature files that our scheme adopts is a Bit-Slice-like method. The Bit-slice signature files are not subject to modification, but the searching is indeed faster. Since the size of the DAYI input file is fixed and only retrieval operations are allowed, it is very suitable to use Bit-Slice storage structure to speed up the access time of the Chinese character retrieval. Moreover, the whole size of the signature file is not very huge, so we can put it all into the main memory in real applications. The storage structure of the signature file in the main memory is different from that in the disk. For the sake of speed improvement, namely to reduce the bit's operation, we adopt the transposed structure of Bit-Slice methods. That is, we transpose the bits of each column and store them using the consecutive bytes. In the implementation, we can find the corresponding columns of the 1's bit of query signature. Putting AND operations into them, we can find the qualified object signature and their corresponding indices for the retrieval of Chinese characters.

3.3 The Wild-character Function

One of the most important features of signature files is that you can find the desired object even when you only know part of the keywords for the object signature. Because of this feature, the implementation of the wild-character function becomes natural and simple. Moreover, since only part of the keywords is being searched for, the weight of the query signature will be lower. The wild-character function does not cost more to make itself more efficient so long as our scheme's storage structure is in the form of a Bit-Slice-like method. The following example will illustrate how our scheme employs the wild-character function.

For example, The DAYI code of Chinese character “龍” is ”KJ9X”. After encoding, the keywords become “K_1”,”J_2”,”9_3”,”X_4” and ”$\$_4$”. If we only know the head and the tail keystrokes of DAYI code of “龍”, we can input “K??X” when we query it. Then our scheme will generate the keywords “K_1”,”X_4” and “$\$_4$”. Hence, the Chinese character “龍” will be the qualified character, and we can retrieve it. If the number of the qualified characters is not one only, we can regard it as the collision problem existing in all Chinese input methods. At this time, all the qualified characters will be shown in the bottom window of the display monitor, and the user can make a choice by pressing a corresponding indication key to specify the desired character.

4. An Alternative Solution with No False Drop

Obviously, the scheme proposed in Section 3 is unlikely to prevent the information loss when large amounts of distinct word signatures are superimposed. That is, the occurrence of false drops is inevitable. However, decreasing the number of 1s when superimposing signatures can alleviate this information loss problem. The best result can be achieved by preparing the corresponding word signature of each keyword before the construction of the signature file. That is, each bit position of a signature can be assigned 0/1 according to the principle of minimal overlap. The goal of obtaining no false drops will be perfectly reached through the assignment of the object signature's length, namely, the possible number of DAYI keywords, to be 164 bits and through setting only one bit for each keyword to “1” (i.e., M=1). We call this approach the *DAYI Signature Bit-Block method* (DSBB). Although this method takes more space and needs more access time, as the experiment in the next section will show, it is still faster than the traditional method.

5. Our Experiment

The experiment described in this section is a real implementation of the DAYI Chinese input method. We choose 5846 frequently used Chinese words for this experiment. For the sake of comparison, we also re-implement the DAYI Chinese Input Method. Since the system of our experiment has to employ the wild-character function, the algorithm that the traditional method uses is the sequential search approach. Before comparing the performance of them, we need to describe their space cost. Because the traditional method is implemented using sequential search, it needs not any additional index overhead; each Chinese character occupies 6 bytes (4 bytes for DAYI code and 2 bytes for the Chinese character itself). Similarly, the DAYI Signature Superimposed Coding (DSSC) that we propose now uses 32 bits to store each object signature. Therefore, the space it takes is the same as that of the traditional method. However, the DAYI Signature Bit Block (DSBB) uses 164 bits to store each object signature. The space it takes is 5 times more than the previous two methods.

In order to compare the performance of these methods, we divide the experiment into two parts. In the first part, we vary the number of the access words, which are generated at random, from 100 to 5000. Figure 1 shows that DSCC has the best performance, although it seems not fair because DSCC has false drops. However, the excellent performance of DSSC is very impressive. Moreover, in the same situation yet with no false drops, Figure 1 also shows that DSBB is 10 times faster than the traditional method. This result is also very noticeable.

The second part of the experiment fixes the access words at 5000 and varies the size of the database from 100 to 5856. The result is shown in Figure 2, and it is almost the same as the first part of the experiment. The difference of the two is that the later shows that the growth rates of DSCC and DSBB are much slower than that of the traditional method when the size of the database increases. That is, the larger the database is, the more efficient our schemes are.

Unit (sec) **number of database words : 5856**

Number of access words	100	500	1000	2000	3000	4000	5000
Traditional method	2.8	13.18	26.97	52.62	78.82	105.63	133.64
DSSC	0.11	0.55	0.99	1.65	2.47	3.3	4.06
DSBB	0.27	1.54	2.91	5.65	8.57	11.54	14.39

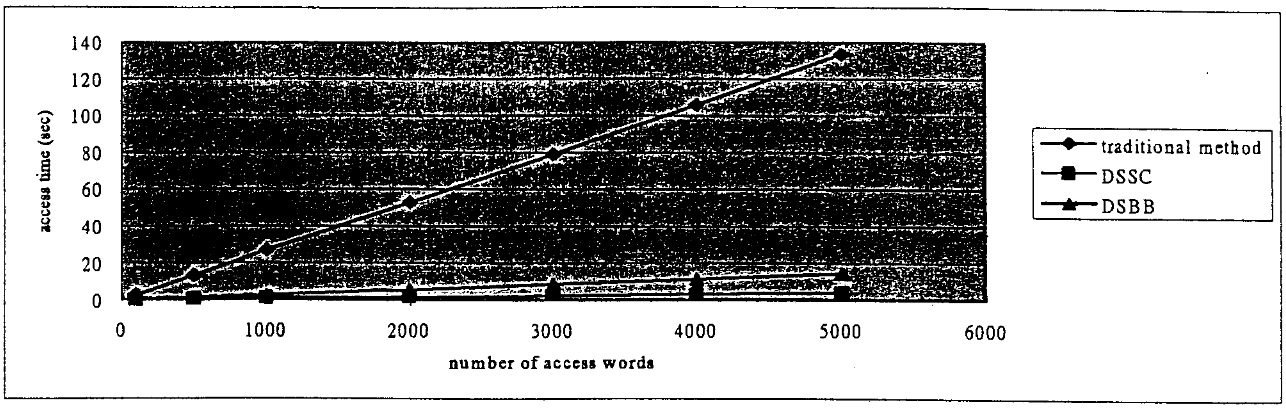

Figure 1

Unit (sec) **number of access words : 5000**

Number of database words	100	500	1000	2000	3000	4000	5000	5856
Traditional method	2.14	10.71	21.75	44.93	67.06	89.2	112.38	133.64
DSSC	0.55	0.72	1.04	1.7	2.25	2.91	3.52	4.06
DSBB	11.86	12.09	12.3	12.74	13.18	13.68	14.12	14.39

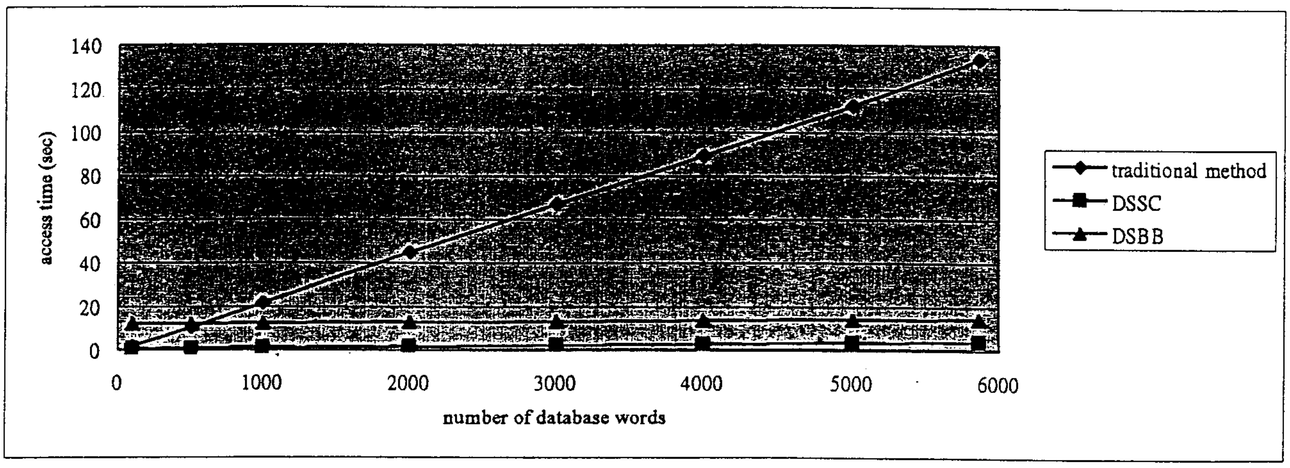

Figure 2

6. Conclusions

In this paper, we have proposed two Chinese character retrieval schemes to modify the DAYI input method. These schemes are both based on signature files. The first scheme, called *DAYI Signature Superimposed Coding*, has the best performance, but it has false drops. The second scheme, called *DAYI Signature Bit Block*, has no false drops, but it takes up more space and access time than the former scheme. Either of these two schemes can be chosen depending on whether the user minds false drops or not. We have also provided an experiment to compare these two newly schemes with the traditional method. The results show that our schemes are indeed efficient in accessing a specified character.

7. References

[1]. Chen, J.N. and Chang, C.C., (1992) "A

Science and Engineering, Vol. 8, 1992, pp. 487-507.

[2]. Chang, C.C. and Lee, C.F., (1997) "Retrieving Chinese Characters Using DAYI Chinese Input Method," *Proceeding of 17th International Conference on Computer Processing of Oriental Languages*, Hong Kong, Apr. 1997, pp. 387-391.

[3]. Chang, C.C. and Lin, D.C., (1994) "Utilizing the Concept of Longest Common Subsequence to Retrieve Similar Chinese Characters," *Computer Processing of Chinese and Oriental Languages*, Vol. 8, No. 2, Dec. 1994, pp. 177-191.

[4]. Faloutsos, C. and Christodoulakis, S., (1984) "Signature Files: An Access Method for Documents and Its Analytical Performance Evaluation," *ACM Transactions on Office Information System*, Vol. 2, No.4, 1984, pp. 267-288.

[5]. Faloutsos, C., " Signature-based text retrieval methods: a survey," *Data Engineering Bulletin, IEEE Computer Society*, Vol. 13, No. 1, Mar., 1990, pp.25-32.

[6]. Liu, C.C., Chang, C.C. and Buehrer, J., (1992) "Encoding and Accessing Chinese Words Using Mandarin Phonetic Spellings," *Computer Processing of Chinese and Oriental Language*, Vol. 6, No. 2, Dec. 1992, pp. 195-204.

[7]. Sung, S.Y., (1989) "Chinese Words Accessing Based on Phonetic Input," *Proceedings of International Conference on Chinese and Oriented-Language Computing*, Aug. 1989, pp. 388-392.

[8]. Shan, M.K. and Lee, S.Y., (1998) "Placement of Partitioned Signature Files and Its Performance Analysis," *Information Science: An International Journal*, Vol. 204, No. 3, 1998.

[9]. Wong, H. K. T., Liu H., Olken, F., Rotem, D., and Wong, L., (1985) "Bit Transposed Files," *Proceedings of the Eleventh International Conference on Very Large Data Bases*, Stockholm, Sweden, Aug. 1985, pp. 448-457.

[10]. Zezula, P., Rabitti, F. and Tiberio, P., (1991) "Dynamic Partitioning of Signature Files," *ACM Transactions on Information System*, Vol. 9, No.4, 1991, pp. 336-369.

[11]. 大易輸入法, 王贊傑編著, 六版, 1995, 太易資訊股份有限公司.