Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 6 (5): 1066-1070, 2006
ISSN 1812-5654

A Two-Layer Dictionary Organization for Quickly Retrieval

[1]Jau-Ji Shen and [2]Chun-Ta Li
[1]Department of Information Management,
National Formosa University, 64 Wen-Hwa Road, Huwei, Yunlin, Taiwan 632, R.O.C.
[2]Department of Computer Science, National Chung Hsing University,
250 Kuo Kuang Road, Taichung, Taiwan 402, R.O.C.

Abstract: Present goal in this study was to develop a dictionary searching technique designed specifically for World Wide Web, which is called "Two-layer dictionary retrieval". Two-layer dictionary retrieval, through "signature extraction process" which extracts the "signature" from every vocabulary in the dictionary, generates the dictionary signature file and distributes to every user. The user want to looks up a vocabulary in the dictionary, through the same process that extracts the signature from which the vocabulary is desired, such that the user can search this vocabulary on the downloaded dictionary signature file. Our technique is not only able to minimize the searching area in the dictionary, but is also able to minimize the space required to establish an index so as to greatly improve the efficiency of dictionary retrieval over the Internet. According to the experimental results, our method is five times better than the other methods available in terms of efficiency.

Key words: Dictionary organization, false drop, indexing, internet, spelling check

INTRODUCTION

Nowadays more and more data files have been quickly created each day since 1995, the year in which the World Wide Web has begun to advance so rapidly. To quickly retrieve the desired information for users from such enormous data files is an important issue of dictionary retrieval.

There are currently a number of dictionary searching techniques that have been proposed (Angell *et al*., 1983; Burkhard and Keller, 1973; Doster, 1977; Hall and Dowling, 1980; Kohonen, 1987; Morgan, 1970; Owolabi, 1996; Owolabi and McGregor, 1988; Salton and Wong, 1978; Szanzer, 1969; Szanzer, 1973), which primarily engage themselves in dictionary organization. That is, they place their strategy upon organization for the index establishment. For example, the length of characters contained in a vocabulary is taken as a basis for classification (Morgan, 1970; Szanzer, 1969), or the first letter in a vocabulary is taken as a classification (Hall and Dowling, 1980; Szanzer, 1973). The above two classification methods, after undergoing the experiment (Owolabi, 1996) that O. Owolabi pointed out, result in an uneven distribution that impedes the searching efficiency. Thus O. Owolabi combined the two methods in which he first classifies the primary index by the length of characters in a vocabulary, then moves on to the secondary index and classifies by the first letter from the length of characters in the vocabulary. This way he can amend the uneven distribution, but this way brings up another issue leading to larger space required for the indexing. For example, assume in the primary index containing the length of characters from 2 to 50 and there are a total of 49 classifications, if he wants to classify each vocabulary by its first letter in the subordinate index and each length of characters contains 26 classifications (from A to Z), there will be a total of 1274 classifications needed to establish the secondary index and thus the resulting large file will make it difficult to be transferred over the Internet. Therefore our goal in this study is to make possible the even distribution in the dictionary and minimize the additional space required to establish the indexes. Furthermore, our technique is not only able to greatly improve the efficiency of dictionary retrieval over the Internet, but also to be applied on the spelling check that it can quickly retrieve stored lexicon to spell check the vocabulary does really correctness in the dictionary.

The concept of two-layer dictionary retrieval: Present study proposes a two-layer dictionary organization

Corresponding Author: Jau-Ji Shen, Department of Information Management, National Formosa University,
64 Wen-Hwa Road, Huwei, Yunlin, Taiwan 632, R.O.C.
Tel: 886-5-631-5197 Fax: 886-5-636-4127

Table 1: Statistical table of the selected medical dictionary containing the number of occurrences for variant characters

Sort	Character	Number of occurrences	Sort	Character	Number of occurrences	Sort	Character	Number of occurrences
①	e	202760	⑰	y	45348	㉝	3	386
②	a	191554	⑱	g	39335	㉞	4	329
③	I	182916	⑲	b	28847	㉟	5	289
④	o	159617	⑳	f	25856	㊱	(	252
⑤	r	149568	㉑	v	19116	㊲	)	252
⑥	s	138910	㉒	x	9334	㊳	6	224
⑦	t	136899	㉓	-	9232	㊴	7	141
⑧	l	117922	㉔	,	6594	㊵	9	95
⑨	Space	103379	㉕	w	6268	㊶	8	90
⑩	c	102408	㉖	z	4267	㊷	/	61
⑪	u	71942	㉗	'	4228	㊸	:	51
⑫	p	70857	㉘	j	2498	㊹	.	31
⑬	m	70257	㉙	q	2070	㊺	+	24
⑭	h	56494	㉚	1	749	㊻	&	8
⑮	d	56454	㉛	2	489	㊼	-	3

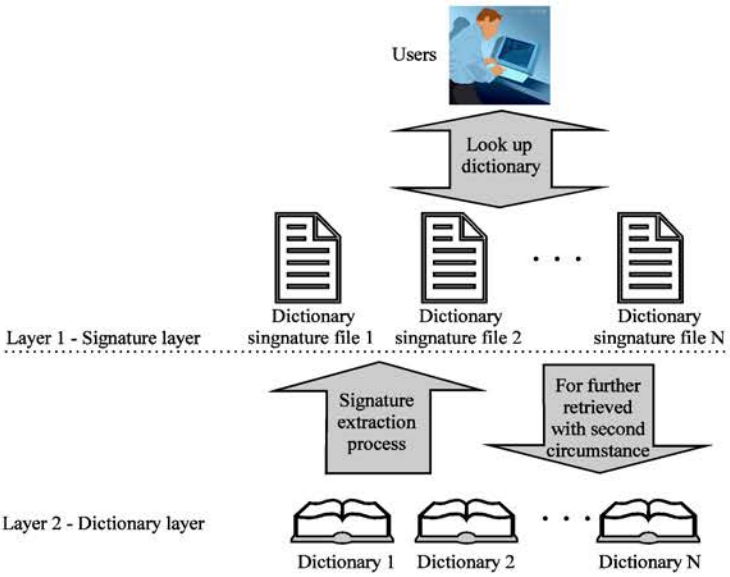

Fig. 1: The architecture of two-layer dictionary retrieval

method also called two-layer dictionary retrieval, in which its architecture is shown in Fig. 1.

The architecture in Fig. 1. can be applied to any current types of dictionaries such as Medical Dictionary, Webster Dictionary, etc. During the pre-process for each dictionary, after the signature extraction process that extracts a dictionary signature file in the first step, the dictionary signature file can now be downloaded depending on user's specific needs. When the user wants to look up the vocabulary, he will directly make his initial retrieval on the specific dictionary signature file at layer 1. Since the dictionary signature file after undergoing the signature extraction process is considered small-sized, this will greatly reduce the cost for sending the information back to the user over the Internet.

There are only two circumstances of retrievals. First circumstance is that the signature for which the vocabulary is to be retrieved after the same signature extraction process does not exist in the dictionary signature file. At this time, the user will receive a definite reply that the vocabulary to be retrieved does not exist in the original dictionary. Second circumstance is that the signature for which the vocabulary is to be retrieved after the signature extraction process does exist in the dictionary signature file, implying that the vocabulary to be retrieved may exist in the original dictionary. It will then remotely log onto the dictionary database at the back end for further retrieval at layer 2. Because the dictionary is pre-classified by the signature, that the searching can be achieved on a few specific areas in the dictionary without going through the entire dictionary. It is this way that makes our dictionary retrieval better prepared and applied to current networks.

The signature extraction process

The encoding table of dictionary: From the architecture of two-layer dictionary retrieval mentioned in the last Section, each vocabulary in the dictionary must be

Table 2: The encoding table of dictionary

Group code	Included character	Sorting condition	Total number of occurrences
00	"e", "n", "l", "d", "y", "k", ",", "2", "3", "0", "9", "_",	1 8 9 16 17 24 25 32 33 40 41 48	568,604
01	"a", "t", "Space", "h", "g", "-", "w", "1", "4", "7", "8", "&",	2 7 10 15 18 23 26 31 34 39 42 47	544,478
10	"I", "s", "c", "m", "b", "x", "z", "g", "5", "6", "/", "+",	3 6 11 14 19 22 27 30 35 38 43 46	539,607
11	"o", "r", "u", "p", "T", "v", "'", "j", "(", ")", ":", "."	4 5 12 13 20 21 28 29 36 37 44 45	504,268

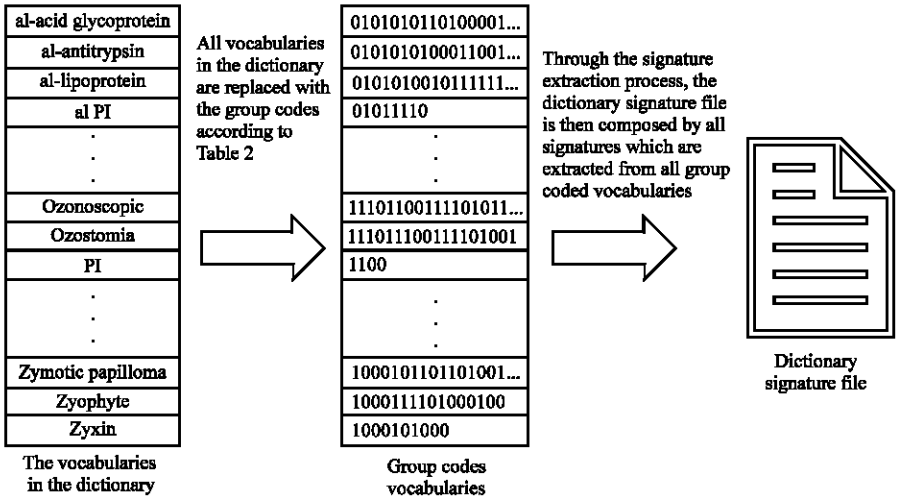

Fig. 2: An example to illustration the process of replacing vocabularies with the group codes according to Table 2

replaced by the group code in the encoding table of dictionary. To ensure that every vocabulary can be evenly distributed onto each group code, the number of occurrences for each character must be all recorded so every vocabulary can be evenly distributed onto each group code in the encoding table of dictionary. Table 1 is a statistical table for a certain medical dictionary containing the variant characters and their number of occurrences that is placed in a decreasing order. For the reason mentioned above, the upper and lower frequency characters are deemed as same code.

In the encoding table of dictionary of this paper, there are four group codes pre-determined which are 00, 01, 10 and 11, respectively. Next, according the sequence shown in Table 1, Characters are distributed onto each group code. The results are shown as Table 2 in the following. From Table 2, we know the difference is not obvious in total number of occurrences; in other words, the distribution of the characters onto each group code is quite even. How to replace the vocabulary in the dictionary with the group code is shown in Example 1. All of the aforementioned process introduced in this section as shown in Fig. 2.

Example 1: If "east" is a vocabulary in the dictionary. By replacing "east" with the group codes according to Table 2, we get "e"-00, "a"-01, "s"-10 and "t"-01. The group coded "east" is 00011001.

Algorithm Signature Extraction:
Input: TX, C
Output: The signature K of TX
Step 1: Calculate the length of binary code of TX as |TX|
Step 2: d = (C–(|TX| mod C)) mod C
  if (d<>0)
    TX = TX + the string of d zeroes // "+" represents the concatenation of sting //
Step 3: Divide TX evenly into TX1, TX2, ...,and TXc
Step 4: P = |TX| / C
  FOR I = 1 **TO** C
    FOR t = 0 **TO** 1
      FOR j = 1 **TO** P
        r_j = (t) XOR (j-th bit of TX_i)
      Next j
      Compose the total bit string R_t as $r_1r_2...r_p$
    Next t
    Compose R_0 and R_1 to check the number of 1's in these two strings. If R_0 has least number of 1's then set K_i = 0 else set K_i = 1
  Next I

Step 5: Output the signature of TX as bit string K as $K_1K_2...K_c$

Fig. 3: The progress of signature extraction algorithm

Dictionary signature file: After all vocabularies in the dictionary are replaced with the group codes in the encoding table of dictionary and in order to reduce the space required for the dictionary signature files, we must move on to extract and partition these group coded vocabularies in terms of their signatures. Present design starts from partitioning the vocabularies and the number of partitions is C. Insufficient number of partitions lead to

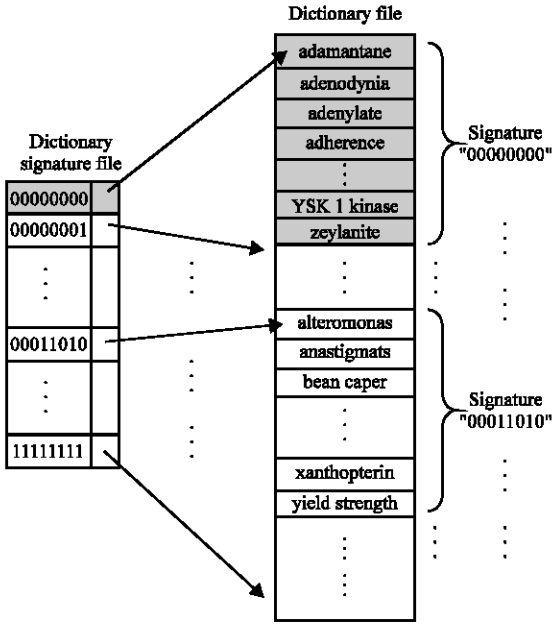

Fig. 4: Establishment of indexing for C = 8

lower distinguishability; extravagant number of partition take up too much space and thus lose the meaning of signature extraction process. Since the length of group codes of every vocabulary is different, the length of each partitioned sub-string from every vocabulary is not the same. A signature extraction algorithm is designed as below to extract the signature from each partitioned sub-string. The signature extraction version of dictionary is then composed by all signatures which are extracted from all group coded vocabularies. Our paper designs the signature extraction process as the following algorithm shown in Fig. 3. Example 2. is a simple illustration that explains the progress of the signature extraction algorithm.

Example 2: Let TX = "0010011001101101 01", set the number of partitions C = 4.

Step 1: |TX| = 18.

Step 2: d = (4 – (|TX| mod 4)) mod 4 = 2,
TX = "0010011001101101 01" + "00"
= "0010011001101101 0100".

Step 3: TX is evenly partitioned into four sub-strings, which are respectively: TX_1 = "00100", TX_2 = "11001", TX_3 = "10110" and TX_4 = "10100".

Step 4: Determine the number of occurrences of 0 and 1 separately for the four sub-strings in TX and take the one with more occurrences as signature code for each sub-string. If the number of occurrences of 0 and 1 are the same, 1 is preferred. The result is K = " 0 1 1 0 ".

Step 5: Signature of TX is "0 1 1 0".

Figure 4 shows how a dictionary signature file is generated from the incorporation of signature of every vocabulary extracted from the signature extraction process and organizes these signatures containing the vocabulary to establish the index. When the signature for which the vocabulary the user wants to look up exists in the dictionary signature file, the user can retrieve from the established index according to the retrieved signatures; in this case, the overall dictionary searching efficiency is improved.

RESULTS AND DISCUSSION

The experimental data is collected from a medical dictionary that contains 142,709 vocabularies, 2,156,957 characters and the average length of characters of every vocabulary is 16. Determination of number of partitions, C, can be made by the size of dictionary or the average length of vocabulary considering the size of which the dictionary signature file is to be sent to the user over the Internet is appropriate. Here C ranges from 8 to 16 and the observation of various partitions which result in different outcomes is shown as Table 3 in the following. We randomly pick 3000 vocabularies for our experiment where 2000 vocabularies do exist and the other 1000 vocabularies do not exist in the dictionary, then we take the average value for these two circumstances of retrievals.

From Table 3, when C gets bigger, not only will the practical range to be searched be much less with bigger number of signatures indices but will the time be much less also and the search time is not even 1 millisecond. Yet this brings up another disadvantage-the memory space occupied is increased. How to determine the size of C value for the best coordination between the number of signature indices and memory space occupied is the critical part to which we should pay special attention in our method. From these experimental results, we obtain the best value for C is 14 (shadow part) considering that this value results in little memory space occupied under the minimum number of words searched and the number of words searched only take up 0.09% of overall vocabulary in the dictionary as well as that the search time is below 1 millisecond. Another reason for picking C = 14 is due to that 60% of the average length of vocabulary in the dictionary is between the length of characters being 7 and 16. When the number of partitions is set as 14, the partition point for the vocabulary exactly locates in the space between the characters to extract exactly the signature in each character so as to enhance the representation to which the index it belongs and the overall search efficiency is increased hence.

In addition, Table 4 is a comparison of our method with previously mentioned methods. These methods

Table 3: The impact of size of C value chosen

C	No. of signature indices	Memory space occupied (bits)	Average No. of words searched	Average search time (10^{-3} sec)
8	256	2048	1212	4.63
9	512	4608	728	3.72
10	1024	10240	610	2.31
11	1928	21208	466	1.48
12	3881	46572	304	1.42
13	5769	74997	259	1.25
14	11345	158830	124	0.78
15	15364	230460	116	0.76
16	25103	401648	107	0.67

Table 4: Summary comparison of four methods

Method	Average dictionary size searched (# of vocabularies)	Dictionary search (%)	Average search time (10^{-3} sec)	False drop probability (%)
Partition by string length	2159	1.51%	14.679	33.33
Partition by first letter	6089	4.27%	39.691	33.33
Tow-level indexing	647	0.45%	3.656	32.49
Two-level dictionary retrieval (C = 14)	124	0.09%	0.782	15.42

include Partition by String Length, Partition by First Letter, Two-Level Indexing and Two-Layer Dictionary Retrieval (C = 14). Partition by String Length and Partition by First Letter are commonly seen dictionary classification methods. Two-Level Indexing by Owolabi (1996) shows the best efficiency. That's why we choose these three methods as subjects of comparison with our two-layer dictionary retrieval. In the comparison, we compare the average number of words searched, average search time and even the false drop probability for the four methods under the circumstance that the vocabulary to be retrieved does not exist in the dictionary to see whether it's effective in filtering the dictionary and reducing the search range. The formula for false drop probability is shown as follows:

$$\text{False drop probability (\%)} = \frac{\text{Number of false drops}}{\text{Number of total executions}} * 100\%$$

From Table 4 it may concluded that present method is far better than other three dictionary search methods considering the average range to be searched in the dictionary is less than 0.1% and average search time is not even 1 m sec. As for false drop probability, our method shows a lower value than others. Even when a false drop circumstance occurs, the range to be searched is greatly decreased because of better filtering capability so as to reduce the search time over the Internet.

CONCLUSIONS

In this study, we proposed a dictionary retrieval system appropriate for current distributive networks-Two-Layer Dictionary Retrieval. Through the signature extraction concept, the dictionary retrieval over the Internet will not need as much space for the indexing so the file to be transferred over the Internet is small, reduce the false drop probability and even reduce the range to be searched in the dictionary by means of dictionary filtering capability for much better overall efficiency of dictionary retrieval in terms of Internet application.

REFERENCES

Angell, R.C., G.E. Freund and P. Willett, 1983. Automatic spelling correction using a trigram similarity measure. Inform. Process. Manage., 19: 255-261.

Burkhard, W.A. and R.M. Keller, 1973. Some approaches to best-match file searching. Comm. ACM, 16: 230-236.

Doster, W., 1977. Contextual postprocessing system for cooperation with a multiple-choice character-recognition. IEEE Trans. Comp., 26: 1090-1101.

Hall, P.A.V. and G.R. Dowling, 1980. Approximate string matching. ACM Comp. Sur., 12: 381-402.

Kohonen, T., 1987. Content Addressable Memories. Springer-Verlag, Berlin.

Morgan, H.L., 1970. Spelling correction in system programs. Comm. ACM, 13: 90-94.

Owolabi, O., 1996. Dictionary organizations for efficient similarity retrieval. J. Sys. Software, 34: 127-132.

Owolabi, O. and D.R. McGregor, 1988. Fast approximate string matching. Software Prac. Exp., 18: 387-393.

Salton, G. and A. Wong, 1978. Generation and search of clustered files. ACM Trans. Databases Sys., 3: 321-346.

Szanzer, A J., 1969. Error-correcting methods in natural language processing. Inform. Process. 68, IFIP, pp: 1412-1416.

Szanzer, A.J., 1973. Bracketing techniques in elastic matching. Comp. J., 16: 132-134.