Effective text compression with simultaneous digram and trigram encoding

Janusz L. Wiśniewski *
Computing Centre, Nicholas Copernicus University, 87-100 Torun, Poland

Received 18 June 1986
Revised 6 January 1987

A new approach to data compression based on digram encoding is presented. Statistical analysis of the linguistic property of digram co-occurrence in textual material leads to a new, more effective simultaneous digram and trigram encoding technique. A combined list of digrams and trigrams eligible for encoding is constructed by an original scheme that replaces some of the frequently co-occurring encodable digrams with so-called master trigrams. Using Shannon's second order Markoff process approximation to English, conditions which the master trigrams should satisfy are formulated and then tested on four representative textual files. The performance of the proposed technique is experimentally examined and compared with the performance of the conventional digram encoding. It is found that the simultaneous digram and trigram encoding produces better results with regard to the measured compression ratios and use of processing time for encoding.

Introduction

Many data processing applications involve storage of natural language text files. This is the case, for example, in large automated library systems and information retrieval systems where the files are generally large and complex. The number of such applications is increasing rapidly and, at the same time, due to proliferation of computer communication networks they involve not only storage, but also massive transfer of data over long-distance teleprocessing links. To reduce the data storage requirements and the data transfer costs, there is a need to reduce the intrinsic redundancy in the data representation. A technique for the reduction of this redundancy is described as data compression.

Some of the most common textual data compression techniques described in the literature include statistical block encoding schemes such as Huffman codes [1] or arithmetic coding [2], which typically encode a block of input character strings as a variable-length string of bits determined by stochastic characteristics of the source data; differential or front-end key encoding methods (e.g. [3,4]), which usually compress a textual file by recording only the difference between adjacent records; and textual substitution or macro encoding schemes (e.g. [5,6]), which factor out multiple occurrences of character strings in a file, replacing the repeated elements with pointers to the character strings, which are stored in a special dictionary. The original file is recovered by substituting dictionary strings for pointers. In addition, many other less common methods of data compression have been presented in the literature. A detailed and complete overview of these methods has recently been presented by Severance in [7].

This paper is devoted to a special type of macro encoding described in the literature [8] as digram encoding. The scheme involved in this technique scans the entire text to be compressed, seeking common patterns of two characters (digrams) occurring frequently and substitutes an unused IBM EBCDIC or ASCII code-word for the located digram. The digrams eligible for replacement (encodable digrams) with character-length code-words are selected, in most of the applications, on the basis of their frequency of occurrence in a given text. Small tables containing the selected digrams are stored in the main memory, and then used for rapid and simple encoding/decoding based on indexing operations.

The digram encoding is an easily implemented compression technique that is able to guarantee a 43–48% storage space reduction [9,10]. Those values compare favourably with the values achieved with the technique of variable length Huffman coding which claimed to be the most effective, in terms of the reported compression ratio data [7]. The latter is, however, much more time-consuming due to the cumbersome bit manipulations involved in compression/decompression procedures. Furthermore, fixed-length codes eliminate

* Currently with Beilstein-Institut, Varrentrappstrasse 40–42, 6000 Frankfurt am Main 90, Fed. Rep. Germany.

North-Holland
Journal of Information Science 13 (1987) 159–164

the desynchronization problems [11], inherently associated with Huffman codes that may cause the propagation of an error during transmission of the compressed data over unreliable communication links.

The approach presented in this paper seeks improvement of the effectiveness of the digram encoding technique by inclusion of a small number of trigrams (strings of three consecutive characters) into the list of encodable digrams and simultaneous encoding of digrams and trigrams from the list. The encodable digrams and trigrams are selected by a new method that relates the performance of compression to linguistic characteristics of textual material. The established set of encodable digrams and trigrams, when used for encoding, proved to be more effective, in terms of the measured compression ratios, than the conventional sets, which contained only digrams or only trigrams as the encodable units. Experimental data showed a relative increase up to 18.8% in the value of the compression ratio with respect to the standard digram encoding technique.

Digram encoding

A set of digrams eligible for encoding is usually selected on the basis of frequency of occurrence. The list of all different digrams occurring in a sample is sorted in descending order of frequency of occurrence; and n top elements, where n denotes the number of all available unused EBCDIC code-words, are selected to be the set of encodable digrams. This method of selection, although the simplest is by no means the most efficient. Since the encodable digrams have a tendency to cluster, it becomes very common that a given encodable digram immediately precedes or immediately follows in the text another encodable digram.

This overlap of encodable digrams reduces compression performance. In the extreme case of a complete overlap of encodable digrams (i.e., the initial character of each occurrence of an encodable digram, except the first, is, at the same time, the second character of an occurrence of another encodable digram) then only half of all encodable digrams will actually be encoded.

The extent of this overlap and its effect on the performance of the digram encoding technique has been studied in detail by Bookstein and Fouty [12]. In particular, they have formulated the expression that relates the probability P that a given digram appearing in a textual sample is encoded with the linguistic property of digram co-occurrence. The probability is calculated as

$$P = E/(1 + E'), \tag{1}$$

where E denotes the probability that a given digram is encodable, while E' denotes the probability that a given digram is encodable, conditional upon its being followed in the sample by an encodable digram. Since P approximates the compression ratio for sufficiently large samples, the expression for P may be used for estimating the effectiveness of digram encoding.

As is seen from (1), the set of encodable digrams should be selected not only on the basis of frequency of occurrence (E in (1)), but also by taking co-occurrence data into consideration (E' in (1)). An encodable digram may be profitably traded for a less frequently occurring digram, only if the degree of overlap is sufficiently reduced.

Essential for further reasoning is the manner in which the probability E' has been calculated. Let K denote the total number of different characters appearing in a sample, and let f_{ij}, for $i, j = 1, 2, \ldots, K$, denote a frequency of occurrence of the ith encodable digram immediately before the jth encodable digram. In practice, f_{ij} is equal to a frequency of occurrence of a trigram such that its two first characters are identical with the ith digram and, simultaneously, the second and the third character are identical with the character pair in the jth digram. Any trigram possessing this property will subsequently be referred to as a *clustering trigram*. For example, the frequency of occurrence of the digram TH immediately before HE is equal to the frequency of occurrence of the clustering trigram THE. Using the above notation, the conditional probability E' may be expressed as

$$E' = \sum_{i=1}^{K} \sum_{j=1}^{K} f_{ij}. \tag{2}$$

On the other hand, the digram frequencies f_{ij} in (2) may be equally well replaced with the frequencies of the clustering trigrams.

Careful analysis of mutual relations between the encodable digram frequency distribution and the clustering trigram frequency distribution has led to a new, more effective compression tech-

nique based on simultaneous digram and trigram encoding.

Rationale

Although the technique presented here refers to n-gram encoding, in this particular case for $n = 2$ and 3, it should not be identified with the n-gram encoding encountered in the variety generator approach for fixed-length representation of variable-length character strings. The latter approach is very well established and recently has been reviewed in detail by Cooper and Lynch in [5].

The variety generation approach involves recognition of the fact that the maximum compression with fixed-length codes will be achieved only if the encodable n-grams are equifrequent. The equifrequency demand results in selection of n-grams whose lengths may differ substantially and one has to deal with the non-trivial problem of matching the coding method to the n-gram generation method [13]. On the other hand, the technique presented in this paper belongs to the group of methods, related to Huffman coding, that compensate for redundancy due to the varying frequencies of encodable n-grams by substitution of variable-length codes. In fact, the technique of simultaneous digram and trigram encoding is a form of variable-length coding with digrams and trigrams as the encodable units: the more common digrams and trigrams are represented by means of 8-bit bytes, while less frequent ones require 16 or 24 bits.

It is well established that one can increase compression by increasing the length of n-grams with non-uniform frequency distribution [14]. However, the problem is much more complex than simple replacement of shorter n-grams by longer ones. Due to the correlated occurrence probabilities of multi-character strings in a natural language text, such a replacement does not have to result necessarily in better compression [15]. This paper fully recognizes that fact and offers a novel method for the selection of trigrams which, by replacing digrams, should provide more effective text compression.

A 55 273 character English language text file containing abstracts of publications on library science was analysed. The total number of different characters appearing in the sample was 57. Hence there were 199 unused EBCDIC code-words left for digram encoding. The set of 199 encodable digrams was selected strictly on the basis of frequency of occurrence by taking the 199 top items from the sorted frequency list of all digrams occurring in the sample. The frequency sublist of the 199 digrams eligible for encoding was compared with the frequency list of all possible clustering trigrams appearing in the sample. Out of 3713 different trigrams occurring in the sample, although 187 193 ($57 \times 57 \times 57$) were possible, 76 proved to be the clustering trigrams. Furthermore, it was determined that 21 clustering trigrams had the important property of having their frequency of occurrence f_T not smaller than $\frac{2}{3}$ of the frequency of occurrence f_D of their front digram, i.e., the digram composed of the first and the second character in its corresponding clustering trigram. For example, the measured frequency of occurrence of the clustering trigram ING was $f_T = 0.048$, which, at the frequency of occurrence of the front encodable digram IN equal to 0.069 ($f_D = 0.069$), satisfied the condition that $f_T > \frac{2}{3} f_D$.

Since the two digrams in any clustering trigram overlap, it becomes obvious that only the first encountered digram, i.e., the front encodable digram (IN in the example) may possibly be encoded. The other digram (NG in the example), although encodable, will not be encoded, for its first character has been encoded already as the second character of the preceding digram. If, additionally, some or all clustering trigrams have the property that $f_T > \frac{2}{3} f_D$, then it is justifiable to assume that the resulting compression will be greater if, instead of encoding only two characters in the front digrams, three-character strings in the trigrams will be encoded with single-byte codes. The assumption would be perfectly true only if the clustering trigrams were occurring independently and did not overlap with preceding and/or following digrams. If the preceding and/or the following digrams happen to be encodable ones, then the overlap may substantially reduce the compression performance and, depending on the degree of this overlap, the expected improvement may not be observed. The extent of this overlap and its effect on the compression performance may, however, be easily estimated by applying the model of Bookstein and Fouty.

Using (1), the probability P was calculated for the frequency list of 199 selected digrams. Then,

for each of the 21 clustering trigrams satisfying the condition that $f_T > \frac{2}{3}f_D$, the same equation was used to calculate the probability P' that one of the two following events takes place: a digram appearing in the sample is encoded or one determined clustering trigram appearing in the sample is encoded. All the 21 values of P' were then compared with P and all those clustering trigrams were selected, for which their values of P' exceeded the value of P. The clustering trigrams selected in this manner, which are referred to in this paper as the *master trigrams*, were then incorporated in the list of encodable digrams. The combined list of the encodable digrams and the master trigrams formed the final encoding tables used during the compression procedure.

In the case of the sample examined, 17 out of the previously selected clustering trigrams were found to satisfy both conditions: $f_T > \frac{2}{3}f_D$ and $P' > P$, and thus they constituted the set of master trigrams for the sample.

The approach adopted in the scheme presented assumes that only if $P' > P$ and $f_T > \frac{2}{3}f_D$ are simultaneously satisfied, one may obtain increased compression by simply encoding the master trigrams instead of some overlapping encodable digrams. In general, however, the assumption is valid, on condition that master trigrams themselves are randomly and independently distributed in a sample, i.e., the probability that a master trigram is encoded, conditional upon its being followed by another master trigram is negligibly low. Considering Shannon's second order Markoff process approximation to English [16], one may assert that stochastic influence of digram co-occurrence does not extend beyond the immediately neighbouring digrams, and hence the condition of non-overlapping master trigrams should be, at least theoretically, fulfilled. Recently, it has also been confirmed experimentally by measurements on representative samples of English language text [17].

The final set of 199 encodable units for the sample under examination was established by removing the 17 least frequent digrams from the list of originally encodable digrams and replacing them with the selected master trigrams. The unused EBCDIC codewords were then assigned uniquely to the 182 remaining encodable digrams and the 17 master trigrams.

The same selection scheme was repeated for three other different textual files and compression performance of the technique was determined in the experiment. As expected, all four files under examination yielded higher compression ratios than in the case of the conventional encoding of only digrams or only trigrams.

Experimental results

It was desired to obtain results in a form that would allow the compression ratios obtained after simultaneous digram and trigram encoding to be compared to the values measured after only digram or only trigram encoding. In the standard digram or trigram encoding, encodable elements are selected strictly on the basis of their frequency of occurrence in a sample and are considered to occur independently of one another. Hence, no linguistic property of digram/trigram co-occurrence is taken into account.

The compression ratio, designated by R, is defined as the fraction of storage space demanded by an uncompressed sample that is saved by using one of the techniques. It is calculated as $R = (N_0 - N_1)/N_0$, where N_0 and N_1 denote the number of characters in the uncompressed and the compressed version of a sample, respectively.

The experiment was carried out on a variety of four English language textual file samples:

Sample No. 1: 27 014 characters – literary English textual file;

Sample No. 2: 84 123 characters – fragment of the PKL Periodical Tape containing titles of periodicals available through the Paul Klapper Library at Queens College, City University of New York;

Sample No. 3: 55 273 characters – fragment of the PKL Abstract Tape containing abstracts of publications on library science available at the same library;

Sample No. 4: 38 705 characters WPASCAL source file.

The results of the experiment are listed in Table 1.

It may be observed that the new technique of simultaneous digram and trigram encoding yielded, depending on the sample, a compression ratio improvement ranging between 2.9 and 18.8%. The smallest value of 2.9% was obtained for Sample No. 4 at a very high absolute value of the initial compression ratio for digram encoding equal to 0.460 (inherently, compression with digram en-

Table 1
Comparative statistics for the simultaneous digram and trigram encoding technique

Position	Description/Sample No.	No. 1	No. 2	No. 3	No. 4
1.	Length of sample (char)	27014	84123	55273	38705
2.	No. of unused EBCDIC codes	197	195	199	200
3.	No. of master trigrams	20	23	17	21
4.	No. of encodable digrams (Pos. 2 – Pos. 3)	177	172	182	179
5.	R (digram encoding)	0.443	0.351	0.403	0.460
6.	R (trigram encoding)	0.375	0.323	0.313	0.389
7.	R (simultaneous digram and (trigram encoding)	0.469	0.417	0.446	0.474
8.	Relative gain [(Pos. 7 – Pos. 5)/Pos. 5] × 100	5.9%	18.8%	10.7%	2.9%

coding is limited to 50%). This small relative gain increased the resulting compression ratio for the simultaneous digram and trigram encoding technique up to a very high value of 0.474. On the other hand, the highest relative gain of 18.8% for Sample No. 2 was obtained at the smallest value of the initial compression ratio for digram encoding equal to 0.351.

Considering the experimental data from Table 1, it is difficult to observe any regularity in the relationships between the measured value of the relative gain in compression and the number of selected master trigrams. The relative gain was calculated with respect to the values of compression ratios yielded by the standard digram encoding. It may, however, be easily checked that the gain would be much higher if the calculations were referred to compression ratios yielded by the trigram encoding technique (position 6 in Table 1).

Simultaneous digram and trigram encoding

Since the technique of simultaneous digram and trigram encoding is about as simple to implement as the standard digram encoding, it may be easily implemented in any application for which the latter proves to be feasible. The process of encoding, based on a left-to-right longest match algorithm, is conducted via rapid indexing operations on two small tables stored in the main memory. The first table contains the master trigrams, while the second contains the remaining encodable digrams. Elements of both tables are indexed with the assigned unused EBCDIC codes.

Comparative statistics of costs in CPU time due to compression/decompression for digram vs. simultaneous digram and trigram encoding were not collected. However, it may be easily observed that the latter technique should provide encoding of a sample in a shorter period of time.

Since, by definition, the numbers of encodable elements that are indexed are identical for both techniques (equal to the number of available unused EBCDIC code-words), the probability of a 'hit' during checking, initially, a three-character string against the table of master trigrams and, eventually, a pair of character against the table of encodable digrams is equal to the probability of the alternative that the three-character string is a master trigram or that the pair of characters is encoded. The probability of these alternative events was determined to be higher than the probability that a given digram is encoded in the standard digram encoding technique. Since the last probability is identical with the probability of a 'hit' during checking a pair of characters against the table of encodable digrams, one may conclude that the technique of simultaneous digram and trigram encoding requires less CPU time for pattern matching. Furthermore, each successful 'hit' in the table of master trigrams results in reducing the length of a still uncompressed portion of a sample by additional characters in comparison with a successful 'hit' in the table of encodable digrams used in the conventional digram encoding technique.

Comments

The extension of the selection method to the general case, in which n-grams of sizes greater

than three are included, is possible. However, it becomes very difficult to model the amount of compression achieved, as greater numbers of inter-relations among digrams must be accounted for in formulating conditions that the conditional probabilities must obey. On the other hand, this additional complexity in both selection scheme and coding technique does not, in most cases, result in increased compression ratios [19].

If the technique presented is to prove effective, the statistical analysis of occurrence of digrams and, additionally, trigrams in a sample must be carried out. The experiment showed that such an analysis is costly in terms of CPU time expenditure when compared with the actual encoding. The additional storage space reduction achieved through this technique was obtained at increased CPU utilization due to selection of the master trigrams. The selection has to be conducted at least once in order to generate coding tables used during compression/decompression procedures. Since digram and trigram frequencies remain relatively stable in expanding natural language databases [10,18], the coding tables generated for a particular database will be sufficiently consistent over a substantial period of time and the statistical analyses of occurrence of digrams and trigrams will not have to be repeated frequently. Thus the additional storage savings offered by the technique should soon amortize the increased development costs.

Acknowledgements

The author is grateful to the Department of Computer Science at Queens College, City University of New York where this research was done. Also acknowledged is the financial support provided by the Kosciuszko Foundation in New York.

References

[1] G.V. Cormack, Data compression on a database system, *Comm. ACM 28* (12) (1985) 1336–1342.

[2] J. Rissanen and G.G. Langdon, Arithmetic coding, *IBM J. Res. Develop. 23* (2) (1979) 149–162.

[3] T.Y. Young and P.S. Liu, Overhead storage consideration and a multilinear method for data file compression, *IEEE Trans. Soft. Eng. 6* (1980) 340–347.

[4] J.M. Bray, V.P. Nelson, P.A.D. deMaine and J.D. Irwin, Data compression techniques ease storage problems, *Comp. Design 24* (14) (1985) 102–106.

[5] D. Cooper and M.F. Lynch, Text compression using variable-to-fixed-length encodings, *J. ASIS 33* (1982) 18–31.

[6] J.A. Storer and T.G. Szymanski, Data compression via textual substitution, *J. ACM 29* (4) (1982) 928–951.

[7] D.G. Severance, A practitioner's guide to data base compression, *Inform. Systems 8* (1) (1983) 51–62.

[8] M. Snyderman and B. Hunt, The myriad virtues of text compaction, *Datamation 16* (16) (1970) 36–40.

[9] W.S. Schieber and G.W. Thomas, An algorithm for compaction of alphanumeric data, *J. Lib. Automation 4* (1971) 198–206.

[10] J.L. Wiśniewski, On some data compression techniques based on the codable elements selection from sources with known statistical characteristics, Ph.D. Thesis, Warsaw Technical University, 1985.

[11] B. Rudner, Construction of minimum-redundancy codes with an optimum synchronizing property, *IEEE Trans. Inform. Theory 17* (1971) 478–487.

[12] A. Bookstein and G. Fouty, A mathematical model for estimating the effectiveness of bigram coding, *Inform. Process. Manag. 12* (1976) 111–116.

[13] E.J. Schuegraf and H.S. Heaps, A comparison of algorithms for data base compression by use of fragments as language elements, *Inform. Stor. Retr. 10* (1974) 309–319.

[14] V.R. Walker, Compaction of names by *x*-grams, *Proc. ASIS 6* (1969) 129–135.

[15] M. Guazzo, A general minimum-redundancy source-coding algorithms, *IEEE Trans. Inform. Theory 26* (1980) 15–25.

[16] C.E. Shannon, A mathematical theory of communication, *Bell System Techn. J. 27* (1948) 379–423, 623–656.

[17] J.L. Wiśniewski, Compression of index term dictionary in an inverted-file-orientated data base: some effective algorithms, *Inform. Process. Manag. 6* (1986) 493–501.

[18] M.F. Lynch, J.H. Petrie and M.J. Snell, Analysis of microstructure of titles in the INSPEC data base, *Inform. Stor. Retr. 9* (1973) 331–337.

[19] E.S. Schwartz, Dictionary for minimum redundancy encoding, *J. ACM 10* (1963) 413–439.