ISSN 1990-4789, Journal of Applied and Industrial Mathematics, 2008, Vol. 2, No. 4, pp. 477–483. © Pleiades Publishing, Ltd., 2008.
Original Russian Text © M.P. Bakulina, 2007, published in Diskretnyi Analiz i Issledovanie Operatsii, Ser. 2, 2007, Vol. 14, No. 2, pp. 3–13.

Application of the Zipf Law to Text Compression

M. P. Bakulina*

Institute of Computational Mathematics and Mathematical Geophysics, pr. Akad. Lavrent'eva 6, Novosibirsk, 6300990 Russia

Received May 17, 2007; in final form, October 29, 2007

Abstract—Under consideration is one of the important problems of information theory, the problem of the compression of data, in particular texts in natural languages, preserving the possibility of their unique restoration (decoding). A way of solution is proposed: to construct the codes that are based on the Zipf law. In distinction from the general methods, a construction of this kind uses the information about the statistical structure of the source of messages. The algorithms of two-pass and single-pass encoding schemes are adduced and the compression effectiveness is estimated.

DOI: 10.1134/S1990478908040042

INTRODUCTION

The data compression is one of the important problems of information theory. This is connected with the fact that the problem of compact representation of data which preserving the possibility of their unique recovery (decoding) arises in practice quite often. For instance, some preliminary compression allows us to significantly improve the characteristics of the communication system. The compression is used in storing information, its transmission via satellites, and in other applications. Moreover, the urgency of the compression problem increases due to the growth of the amount of transmitted and stored information.

Presently, many algorithms for compact data representation are known and find practical application; and they give better compression than the standard archivers. In addition to the algorithms using the letter-by-letter encoding (for example, the well-known Huffman code [4]), there are also known the approaches of the word-based data compression [5, 10, 11]. In these methods, for a given message (for example, a text in a natural language), a frequency dictionary is compiled which is a list of all words used in the text together with the frequency of their appearance. Thereafter, in the message encoding process, there is assigned a code to each word and the higher the word frequency is the shorter the code is. One of the basic approaches to the problem of efficient data encoding, in particular, addressing the text encoding in a natural language, may be the construction of codes which is based on the general laws of linguistics, for example, on the Zipf law [13]. Apart from the universal methods, this law uses the information about the statistical structure of the message source. In the present article, some new algorithms for the text encoding are suggested basing on the Zipf law. These methods have a high degree of compression and a high speed of encoding. The degree (coefficient) of the text compression is defined here as the ratio of the volume of the encoded text to its original volume. In Section 2, under study is the two-pass scheme of compression based on the two ways of encoding the words (uniform and nonuniform). Also, some estimate is established for the quotient of the compression degrees obtained under these methods of coding.

Together with theoretical consideration, the experiments were carried out in which some texts in natural languages were used with various volumes of dictionaries and their compression degrees were computed. Comparison showed that the experimental results are in good agreement with the results obtained analytically. In Section 3, there are considered the algorithms of a single-pass compression scheme with known (unknown) dictionary volume. In Section 4, the best known effective methods of coding are presented, and also there is given comparison of the results of the compression performed by those methods with the results of the Zipf-law-based algorithms.

*E-mail: marina@rav.sscc.ru

1. THE ZIPF LAW

This law was established by G. Zipf on the basis of analysis of the texts in various languages and consists of the following: Let there be a text in a natural language, and let L be the volume of this text dictionary. Let us arrange the words in the dictionary according to diminishing the frequency of their use and enumerate them from 1 to L. Then, the dependence between the frequency and the sequential number of the word can be described as

$$f_i = \frac{k}{i^\alpha},$$

where f_i is the frequency of the ith word, k is a quantity depending on the dictionary volume, and α is a constant. This dependence was called the *Zipf law*. Later, the law was confirmed by many researchers for all European languages [1]. In particular, B. Mandelbrot [8] showed that the Zipf law would hold if the blank space between the words is considered as a random symbol. He also obtained this distribution using the assumption that the evolutionary process of the selection of the word length can be described as a random walk [9].

Let us divide both sides of this equality by N, where N is the total number of words in the text. Then the Zipf law can be written as

$$p_i = \frac{c}{i^\alpha}, \qquad c = k/N,$$

where p_i is the probability of the ith word. It is known that, for the words of a natural language, $\alpha \approx 1$ and the Zipf law is given by the formula

$$p_i = c/i,$$

which we will use in the sequel. Moreover, since $\ln p_i = \ln c - \ln i$, the dependence between the frequency and the sequential number of the word in the dictionary is very close to a linear one in the logarithmic coordinates; i.e., it can be represented by a straight line in the graph with the axes $\ln i$ and $\ln p_i$.

2. CODING WORDS, UNIFORMLY AND NONUNIFORMLY WITH RESPECT TO OUTPUT

Let D be the volume of the dictionary of a text in a natural language. We arrange all words according to diminishing of their frequencies and consider some coding the words that uses the Zipf law and is uniform (nonuniform) with respect to output. An algorithm of encoding such that, at first, the frequency dictionary is compiled and, then, encoding of the words of the dictionary is performed based on the available statistics, will be called a *two-pass compression scheme*. Under the Shannon coding, a code word of the length $\lceil -\log p_i \rceil$ corresponds to each word, where p_i is the probability of the ith word. Under the uniform coding, the length of the code word equals $\log D$.

Thus, the length of the text obtained under the Shannon coding is equal to

$$B_1 = N \sum_{i=1}^{D} p_i \lceil -\log p_i \rceil, \tag{1}$$

where N is the total number of words in the text. Under the uniform coding, the length of the text is equal to

$$B_2 = N \cdot \lceil \log D \rceil. \tag{2}$$

Our task is to figure out whether it is possible to achieve a substantial advantage in the compression with nonuniform coding in comparison with the uniform one. We define the coefficient (degree) of the text compression as the ratio of the volumes (in bytes) of the text obtained by coding and the original text.

Theorem. *Let there be a text with a dictionary of volume D; and, in the text, let the distribution of words with respect to frequency be a subject of the Zipf law. Let n_1 and n_2 be, correspondingly, the compression coefficients obtained under the coding of the words nonuniform and uniform with respect to output. Then*

$$\frac{n_1}{n_2} \to \frac{1}{2} \quad \text{as} \quad D \to \infty.$$

Proof. Let us estimate the volume B_1 of the text obtained under the nonuniform coding. Using (1) and taking into account that

$$-\log p_i \leqslant \lceil -\log p_i \rceil < -\log p_i + 1,$$

we infer

$$N \sum_{i=1}^{D} p_i(-\log p_i) \leqslant B_1 < N \sum_{i=1}^{D} p_i(-\log p_i + 1). \tag{3}$$

According to the Zipf law,

$$p_i = \frac{k}{i}, \tag{4}$$

where k is a constant depending on D. For evaluation of this constant and for the further transformations, we will use the next available estimates of finite sums (see [3]):

$$\ln(D+1) < \sum_{i=1}^{D} \frac{1}{i} < \ln(D+1) + c_1, \tag{5}$$

$$\frac{\ln^2(D+1)}{2} + c_2' < \sum_{i=1}^{D} \frac{\ln i}{i} < \frac{\ln^2(D+1)}{2} + c_2, \tag{6}$$

where $c_1 = 0.577\ldots$ is the Euler constant, $c_2' = -0.3$, and $c_2 = -0.105\ldots$ (the value of c_2 can be obtained from the Euler–Macloren series [3]). Since

$$k \sum_{i=1}^{D} \frac{1}{i} = 1,$$

we have

$$\frac{1}{\ln(D+1) + c_1} < k < \frac{1}{\ln(D+1)}. \tag{7}$$

Applying (3)–(7) and the familiar inequality $\ln(1+x) < x$ for $x > -1$, we arrive at

$$\begin{aligned}
B_1 &< N \cdot \left(\sum_{i=1}^{D} \frac{k}{i} (\log i - \log k + 1) \right) = \frac{N \cdot k}{\ln 2} \cdot \left(\sum_{i=1}^{D} \frac{\ln i}{i} + (\ln 2 - \ln k) \cdot \sum_{i=1}^{D} \frac{1}{i} \right) \\
&< \frac{N}{\ln 2 \cdot \ln(D+1)} \cdot \left(\frac{\ln^2(D+1)}{2} + c_2 + \big(\ln 2 + \ln(\ln(D+1) + c_1) \big) \cdot \big(\ln(D+1) + c_1 \big) \right) \\
&< \frac{N}{\ln 2 \cdot \ln(D+1)} \left(\frac{\ln^2(D+1)}{2} + c_2 + \Big(\ln 2 + \ln\ln(D+1) + \frac{c_1}{\ln(D+1)} \Big) \cdot \Big(\ln(D+1) + c_1 \Big) \right) \\
&= \frac{N}{\ln 2 \cdot \ln(D+1)} \cdot \left(\frac{\ln^2(D+1)}{2} + c_1 \cdot (1 + \ln 2) + c_2 \right. \\
&\qquad \left. + \ln\ln(D+1) \cdot \big(\ln(D+1) + \ln 2 \big) + c_1 \cdot \ln\ln(D+1) + \frac{c_1^2}{\ln(D+1)} \right).
\end{aligned} \tag{8}$$

Taking (2) into account, we obtain the estimates for the text length B_2 under the uniform coding:

$$N \log D \leqslant B_2 < N (\log D + 1). \tag{9}$$

If $n_1 = B_1/A$ and $n_2 = B_2/A$ are the text compression coefficients obtained under the nonuniform and uniform codings respectively, where A is the initial length of the text, then, using (7) and (8), we

Table 1.

No	Name	D	n_1 (%)	n_2 (%)	n_1/n_2 theor.	n_1/n_2 exp.
1	article	1,132	17.4 (17.5)	21.7 (21.8)	0.802	0.803
2	book	2,100	17.9 (18.1)	23.2 (23.3)	0.772	0.777
3	journal	2,480	17.2 (17.3)	22.5 (22.6)	0.764	0.765
4	document	3,700	17.5 (17.7)	23.8 (24.0)	0.735	0.738
5	works	21,197	16.6 (16.7)	25.4 (25.5)	0.654	0.655

obtain the upper estimate for n_1/n_2

$$\frac{n_1}{n_2}=\frac{B_1}{B_2}<\frac{\ln(D+1)}{2\ln D}+\frac{c_1\cdot(1+\ln 2)+c_2}{\ln(D+1)\cdot\ln D}+\frac{\ln\ln(D+1)}{\ln D}+\frac{\ln\ln(D+1)\cdot(\ln 2+c_1)}{\ln(D+1)\cdot\ln D}. \tag{10}$$

The right-hand side of (10) tends to $1/2$ as $D\to\infty$. Analogously, using (3)–(7), we have

$$B_1\leqslant N\sum_{i=1}^{D}p_i(-\log p_i)=N\sum_{i=1}^{D}\frac{k}{i}(\log i-\log k)=\frac{Nk}{\ln 2}\left(\sum_{i=1}^{D}\frac{\ln i}{i}-\ln k\sum_{i=1}^{D}\frac{1}{i}\right)$$
$$>\frac{N}{\ln 2\cdot(\ln(D+1)+c_1)}\cdot\left(\frac{\ln^2(D+1)}{2}+c_2'+\ln\ln(D+1)\cdot\ln(D+1)\right). \tag{11}$$

By (9), we obtain from (11) the lower estimate for n_1/n_2

$$\frac{n_1}{n_2}>\frac{1}{2}-\frac{1}{\ln(D+1)+1}+\frac{c_2'}{(\ln(D+1)+1)^2}+\frac{\ln\ln(D+1)\cdot\ln(D+1)}{(\ln(D+1)+1)^2}. \tag{12}$$

The right-hand side of (12) tends to $1/2$ as $D\to\infty$.

Hence, $n_1/n_2\to 1/2$ as $D\to\infty$. This completes the proof of the theorem. □

It follows from the theorem that, using the nonuniform word coding for the texts in natural languages, we can achieve more than two-fold improvement in compression in comparison with the uniform coding.

The above-obtained theoretical result was verified experimentally. For the experiment, a number of texts in Russian and in English with various volumes of the dictionary were taken and the coefficients of their compression were computed both under the uniform and nonuniform codings.

In Table 1, there are given the text compression coefficients n_1 and n_2 expressed by percent (the experimental results are given in parenthesis), the volumes of the dictionaries D, and also the ratio n_1/n_2 obtained theoretically and experimentally.

Here we used for the experiment:

"article" — an article from a journal (67,739 bytes);

"book" — a monograph (34,9270 bytes);

"journal" — Siberian Mathematical Journal (307,930 bytes);

"document" — documentation (962,186 bytes);

"works" — a collection of works by A. S. Pushkin (5,410,342 bytes).

As it is seen from Table 1, together with the growth of dictionary, the ratio of the compression coefficients decreases. Moreover, if the dictionary volume is large then there is observed a significant gain in compression. Let us indicate also that, in practice, because of the boundedness of the text and the dictionary, the experimental result only approaches the theoretical limit $1/2$, not attaining it (for example, for the hypertext "works" of the size of 5410 KB with a sufficiently large dictionary of 21,197 words, the ratio n_1/n_2 is equal to 0.654).

Table 2.

No.	Name	Text volume (bytes)	k with known dict. (%)	k with unknown dict. (%)
1	article	67,739	24.7	25.1
2	book	349,270	24.9	25.7
3	journal	307,930	24.2	25.3
4	document	962,186	25.4	26.2
5	works	5,410,342	25.8	26.5

3. THE SINGLE-PASS COMPRESSION SCHEME

Consider now the *single-pass compression scheme* based on the Zipf law. In distinction from the two-pass scheme where it is assumed that the dictionary is compiled on the initial stage, in the single-pass variant the dictionary is filled in the processing the text.

First, consider the case when the volume of the dictionary is known at the initial stage. Let D be the total number of words in the dictionary. We will perform the word coding as follows: Compile the frequency dictionary for the portion of the text, read up to a certain word (the word to be encoded). If it is not in the current dictionary (i.e., it is new) then we put it in the dictionary and encode letter-by-letter. On the other hand, if we have already encountered it then encode it according to the Zipf law; and, in this case, the code length will be equal to

$$l_i = \left\lceil -\log \frac{c(D)}{i} \right\rceil, \qquad c(D) \approx \frac{1}{\ln D},$$

where i is the sequential number of the word in the dictionary. After reading the next word of the text and putting it in the dictionary, all words in the current dictionary are reordered and placed according to diminishing frequencies. A special code word (flag) will be inserted in front of each new word, and the length of the flag will be

$$\left\lceil -\log \left(1 - \sum_{i=1}^{j} \frac{c(D)}{i} \right) \right\rceil,$$

where j is the number of words in the current dictionary. During the message decoding at the receiving end, the flag is easily determined (the addressee knows it) and the code sequence that follows it is determined too.

Let us now consider the case when the dictionary volume is unknown at the initial step. The encoding algorithm is constructed in the same way as in the case of known volume. The only difference is that, at each step, the unknown quantity $c(D)$ is estimated using the current dictionary; i.e., if D_j is the number of words in the dictionary compiled for the text read up to the jth word then we put

$$c(D_j) \approx \frac{1}{\ln D_j}.$$

Table 2 presents the results of compression obtained under the single-pass coding with the known and unknown volume of the dictionary for the same texts as in Table 1; i.e., article, book, journal, document, and works.

Note that the best compression is given by the two-pass variant (see Table 1). At the same time, it follows from Table 2 that, in the single-pass system, the compression with the known dictionary volume is better than the compression obtained with the unknown volume of the dictionary.

Table 3.

No	Name	Compression (bit/symbol)						
		HUF	PPMC	WORD	DMC	MTF	LZ	**ZIPF**
1	article	2.09	2.46	2.58	2.67	2.72	2.84	**2.25**
2	book	2.02	2.32	2.75	2.55	2.90	3.12	**2.18**
3	journal	2.03	2.48	2.54	2.51	2.66	2.76	**2.16**
4	document	2.05	2.36	2.39	2.82	2.71	2.78	**2.23**
5	works	2.01	2.26	2.36	2.55	2.49	2.60	**2.12**

4. THE MAIN ALGORITHMS OF COMPRESSION; COMPARISON OF THE RESULTS

To evaluate the effectiveness of the above-proposed single-pass scheme of coding using the Zipf law, we conducted comparison of the compression results with the results given by other available algorithms.

One of the available compression scheme, based on minimization of redundancy, is the Lempel–Zev algorithm [14, 15] (called the LZ-algorithm) and its modifications (for example, the Welch modification [12]). In the LZ77-algorithm [14], the word to be encoded is subdivided, according to a certain rule, into some subwords whose codes are the pairs of numbers. The LZ78 coding scheme [15] differs from the previous since at each step the longest beginning of the remainder is chosen which coincides with a certain already marked out subword and one more letter is added to it. Note that the LZ78-algorithm can be considered as a coding with a dynamic dictionary. The dictionary can be arbitrarily numbered and the words whose all continuations are already in the dictionary can be deleted from it.

In addition, the most widespread and effective methods of coding are indicated in [5]. Consider some of them. Among the algorithms of [5], fairly high compression is provided by the PPM method of J. Cleary and I. Witten. In the process of coding by this method, there are evaluated, instead of the unconditional probabilities of letters, their conditional probabilities in the available "context"; i.e., under the known preceding letters. In the PPM, there are used the contexts of various length, depending on the preceding encoded sequence of data. The main problem of PPM is the evaluation of the probability of the symbols not yet having appeared in the context. To solve this problem, some variants of PPM are proposed in [5], i.e., the PPMA and PPMB algorithms. They use a priori methods based on the assumptions about the nature of the compressed data. Another improved version of PPM is the PPMC algorithm in [10]. This method allows us to improve compression and increase the encoding speed. Of interest are also the DMC method of statistical modeling which is described in [5], the WORD algorithm where the symbols are words, and the MTF scheme. The MTF algorithm was developed by B. Ya. Ryabko [2] and was called the *book pile method*. Later, the book pile method was rediscovered by P. Elias [7] under the name of *coding according to the recency rank*, and by J. L. Bentley et al. [6], under the name of *move up*. In the MTF algorithm, the text is considered as an alternating sequence of words and the symbols which are not words. After the next appearance, each word in the text is deleted from the current position and moved forward. In result, the words that appear frequently in the text are placed in front of the words that appear rarely. Moreover, the shorter codes are assigned to the words which are placed before. If a word (a symbol) appears for the first time, it is encoded letter-by-letter; otherwise, encoding is performed taking into account the frequency of the word appearance in the text. An extra code word (flag), which is sent to the addressee, allows us to determine whether the incoming word has already appeared or it is new.

To compare the results, the author has realized in practice some of the most efficient algorithms of [5], in particular, the PPMC, WORD, DMC, and MTF scheme, as well as the new and above-described algorithm ZIPF based on the Zipf law (in particular, the single-pass scheme of coding with the unknown dictionary volume). The known Huffman algorithm of optimal coding (HUF) and the Lempel–Zev LZ78 algorithm have also been realized. The experimental results of compression given by these algorithms for various test files were obtained on IBM PC with the Pentium processor and the short-term memory of

Table 4.

No	Name	Time (s)						
		HUF	PPMC	WORD	DMC	MTF	LZ	**ZIPF**
1	article	2.95	1.61	1.44	1.15	1.07	0.83	**0.97**
2	book	9.78	8.32	7.16	6.48	5.69	5.11	**5.42**
3	journal	8.81	7.45	6.33	5.96	5.18	4.76	**4.90**
4	document	24.14	22.83	20.84	19.17	19.09	18.51	**18.85**
5	works	75.22	73.05	70.62	68.94	68.13	66.48	**67.01**

256 MB under the Windows 98 operation system. Table 3 gives the results of compression in terms of the most convenient and most frequently used in the literature unit of the compression degree of bit/symbol.

Alongside the compression degree, an important characteristic of the method is the speed (time) of encoding. For the above methods, the experimental results are given in Table 4 for the compression time, where the time of the program work was taken averaged over 10 runs (a 1,000 Hz timer was used).

It is seen from Table 3 that the compression coefficient obtained by the Zipf-law-based method is greater only than the compression coefficient obtained by the Huffman optimal code but less than the compression coefficients provided by the other methods under consideration. Comparison of the algorithms with respect to time (Table 4) shows that the Huffman code has the largest encoding time; whereas, the proposed ZIPF method has a smaller encoding time. This time is greater than the time of the Lempel–Zev algorithm, but the compression coefficient of the ZIPF method is smaller. Thus, the results of comparison demonstrate a high degree and a high speed of compression of the encoding method based on the Zipf law, which confirms the effectiveness of the above-constructed algorithm.

REFERENCES

1. P. G. Piotrovskii, *Text, Machine, and Human Being* (Nauka, Leningrad, 1975) [in Russian].
2. B. Ya. Ryabko, "An Effective Method of Encoding Information Sources Using the Fast Multiplication Algorithm," Problemy Peredachi Informatsii **31** (1), 3–12 (1995). [Problems Inform. Transmission **31** (1), 1–9 (1995)].
3. G. M. Fikhtengol'ts, *A Course of Differential and Integral Calculus*, Vol. 2 (Fizmatlit, Moscow, 2001) [in Russian].
4. D. A. Huffman, "A Method for the Construction of Minimum-Redundancy Codes," Proceedings of the IRE **40** (9), 1098–1101 (1952).
5. T. C. Bell, J. H. Cleary, and I. H. Witten, *Text Compression* (Englewood Cliffs, Prentice Hall, 1990).
6. J. L. Bentley, D. D. Sleator, and R. E. Tarjan, "A Locally Adaptive Data Compression Scheme," Comm. ACM **29** (2), 320–330 (1986).
7. P. Elias, "Interval and Recency Rank Source Encoding: Two On-Line Adaptive Variable-Length Schemes," IEEE Trans. Inform. Theory **33** (1), 3–10 (1987).
8. B. Mandelbrot, "On Recurrent Noise Limiting Coding," in *Proceedings of the Symposium on Information Networks* (Polytechnic Institute of Brooklyn, New York, 1954), pp. 205–221.
9. B. Mandelbrot, "On the Theory of Word Frequencies and on Related Markovian Models of Discourse," in *Proceedings of the Symposium on Applied Mathematics*, Vol. 12: *The structure of Language and Its Mathematical Aspects* (Amer. Math. Soc., Providence, RI, 1961), pp. 190–219.
10. A. M. Moffat, "Word Based Text Compression," Software-Practice and Experience **19** (2), 185–198 (1989).
11. E. S. Schwartz, "A Dictionary for Minimal Redundancy Encoding," J. Assoc. Comput. Math. **10** (4), 413–439 (1963).
12. T. A. Welch, "A Technique for High-Performance Data Compression," IEEE Computers **17** (6), 8–19 (1984).
13. G. K. Zipf, *Human Behavior and the Principle of Least Effort* (Addison Wesley, Cambridge, 1949).
14. J. Ziv and A. Lempel, "A Universal Algorithm for Sequential Data Compression," IEEE Trans. Inform. Theory **IT-23** (3), 337–343 (1977).
15. J. Ziv and A. Lempel, "Compression of Individual Sequences via Variable-Length Coding," IEEE Trans. Inform. Theory **IT-24** (5), 530–536 (1978).