1975

On the Limits of Text File Compression

V. Y. Shen

M. H. Halstead

Report Number:

77-257

ON THE LIMITS OF TEXT FILE COMPRESSION

V. Y. SHEN and M. H. HALSTEAD

Computer Sciences Department
Purdue University
W. Lafayette, IN 47907

CSD-TR 257

KEY WORDS AND PHRASES: file compression, compression ratio computation

CR Categories: 3.73

Frank Rubin, in his paper "Experiments in Text File Compression" [6], discussed several computationally feasible algorithms in detail with the goal of achieving "the greatest possible degree of compression". The measure of compression is defined as a ratio

$$CR = \frac{\text{len(input string)} - [\text{len(output string)} - \text{len(output rep)}]}{\text{len(input)}}$$

expressed as a percentage. The author pointed out that all the methods described required a "fairly high" computation time and "large" storage requirements. It would be reassuring if one could show that the limits of the compression ratio were "almost" reached by some of the methods, so that additional efforts could not be profitably applied. We believe that a recently validated hypothesis of software science may be used to establish such limits [3, 4].

Software science was originally developed to measure properties of computer programs. Let η_1 be the count of distinct or unique operators and η_2 be the count of unique variables or constants (operands) in a program. The length of of a program, which is defined as the sum of the total uses of operators and operands, can be expressed as

$$N = \eta_1 \log_2 \eta_1 + \eta_2 \log_2 \eta_2 \qquad (1)$$

Equation (1) has been experimentally shown to be valid for computer programs covering a wide range of sizes and languages [1, 2]. Equation (1) therefore provides a direct relationship between unique entities and total entities in written material. From the same sources [3, 4], the number of bits required to represent that program is

$$V = N \log_2 \eta \qquad (2)$$

where $\eta = \eta_1 + \eta_2$. Several other properties of computer algorithms and their representation in programs were also derived based on the observation that a program is a string of operators and operands.

The hypothesis may be applied to English prose by noting that it is a string of two classes of words: the "function" words and the "content" words [5]. The function words include those which are traditionally called articles, prepositions, pronouns, conjunctions, and auxiliary verbs, plus certain irregular forms; and the content words include those which are traditionally called nouns, verbs, and adjectives, plus most of the adverbs. If we consider the function words as operators and the content words as operands, Equations (1) and (2) may be used to relate the length of a text file and its non-redundant representation (compressed form) in bits. The limit on the compression ratio may then be expressed as

$$CR = \frac{B_w N - (V + B_w \eta)}{B_w N} , \tag{3}$$

where B_w is the average number of bits per word in the text file and $B_w \eta$ is the length of the output representation in bits.

Since we do not have Rubin's test file available, we replace Equation (1) by the approximation

$$N = \eta \log_2 (\eta/2) . \tag{4}$$

Equation (4) is the same as Equation (1) when $\eta_1 = \eta_2 = \frac{\eta}{2}$. It can be shown that, for the range of values that we are interested in, less than 3% error is introduced even for $\eta_2 = 4\eta_1$. Substituting Equations (2) and (4) into (3), we have

$$CR = 1 - \frac{\log_2 \eta}{B_w} - \frac{1}{\log_2 \frac{\eta}{2}} \tag{5}$$

The first test file used by Rubin contained 29,305 characters, of which 85 are distinct. Since an English word contains five letters on the average, the approximate number of words in the file is $N = 29305/5 = 5861$ words. Solving Equation (4) for η, about 694 of these words would be unique.

We have

$$\log_2 \eta = 9.44$$

$$\log_2 \frac{\eta}{2} = 8.44$$

and

$$B_w = 5 \lceil \log_2 85 \rceil = 35.$$

Substituting these values into Equation 5, the limit of compression is

$$CR = 1 - \frac{9.44}{35} - \frac{1}{8.44} = 61.2\%$$

The best compression ratio observed by Rubin for this text file is 58.1% (Table V). Thus the incremental method appears to be very effective in text compression.

REFERENCES

1. Bohrer, Robert. Halstead's Criteria and Statistical Algorithms. Proc. Eighth Annual Computer Science/Statistics Interface Symposium, Los Angeles (February 1975), 262-266.

2. Elshoff, James L. Measuring Commercial PL/I Programs using Halstead's Criteria. ACM SIGPLAN Notices 11, 5 (April 1976), 38-46.

3. Funami, Y., and Halstead, M. H. A software physics analysis of Akiyama's debugging data. Proc. Symp. Software Engineering (1976), Polytechnic Press, New York, 6 pages.

4. Gordon, R. D., and Halstead, M. H. An experiment comparing Fortran programming times with the software physics hypothesis. Proc. National Computer Conference (1976), 935-937.

5. Miller, G. A., Newman, E. B., and Friedman, E. A. Length frequency statistics of written English. Information and Control (1958), 370-389.

6. Rubin, F. Experiments in text file compression. Comm. ACM 19,11 (1976), 617-623.