Empirical Software Engineering, 5, 155–159 (2000)

# Correction of the Software Science Length Estimator Skewness for ‘C’ Language Programs

MICHAEL G. GONZALES, PHD
*Motorola, Inc., Broadband Communications Sector, Horsham, PA 19044*

SESHADRI PARAVASTU
*Motorola, Inc., Broadband Communications Sector, Horsham, PA 19044*

**Abstract.** Numerous studies have confirmed the skewness of Halstead’s Software Science Length Estimator (Beser, 1983; Gonzales, 1990). The Length estimator consistently underestimates the size of ‘small’ programs (program size $<$ 400 tokens), and overestimates the size of ‘large’ programs (program size $>$ 4000 tokens). This paper verifies and corrects the Halstead Length Estimator skewness for a large collection of ‘C’ programs of varying sizes.

**Keywords:** Software metrics, Halstead metrics, software science

## 1. Introduction

The Halstead Length Estimator

$$\mathbf{N}' = \eta_1 \mathbf{log}_2\, \eta_1 + \eta_2 \mathbf{log}_2\, \eta_2 \tag{1}$$

where $\eta_1$ is the number of unique operators in a program, and $\eta_2$ , the number of unique operands, is perhaps the central metric in Halstead’s Software Science (Halstead, 1977).

A study performed in 1983 by Beser (1983) verified and corrected the Halstead Length Estimator skewness for a large collection of FORTRAN programs of varying sizes. In 1990 Gonzales (1990) verified and corrected the skewness for ‘small’ Pascal programs.

This paper considers a database from Motorola, Broadband Communications Sector, consisting of approximately 875 ‘C’ programs ranging in length from 20–30 tokens to approximately 8,000 tokens. The skewness of the Halstead length estimator for ‘small’ and ‘large’ programs is verified and corrected.

## 2. Flaws in the Software Science Model

Beser (1983) showed that there is a fundamental flaw in the Software Science model for program generation. The program generation model assumes a uniform probability distribution for token selection in program construction. In (Beser, 1983) it is shown that this is not true; in fact, the distribution changes as a function of program size. It is the erroneous assumption regarding token selection distribution that leads to the skewness in the length metric.

## 3. Gustav Herdan and Statistical Theory of Language

Gustav Herdan has done a significant amount of work in the statistical theory of language (Herdan, 1956, 1964, 1966, 1960) taking into account that the distribution of token selection changes in relation to the size of a body of text. A simple Herdan-type length metric relating total tokens, **N**, to total unique tokens, **V**, is

$$\mathbf{N} = \alpha \mathbf{V}^{\beta}, \tag{2}$$

where $\alpha$ and $\beta$ are constants [to be determined]. Beser (1983) took Eq. 2, converted it to log-linear form, and solved for $\alpha$ and $\beta$ using a large database of FORTRAN programs. His results are as follows:

| | |
|---|---|
| Exponent | 1.5062 |
| ln (constant) | −1.4122 |

## 4. Fitting the Model to ‘C’ Language Programs

Approximately 875 ‘C’ programs from a Motorola database were parsed using the tool CMT++ (Testwell Oy, 1997), yielding total unique token count, **V**, and total token count, **N**, for each program. The Herdan-type length metric [Eq. 2] was converted to the log-linear equation

$$\mathbf{ln\ N} = \mathbf{ln}\ \alpha + \beta\, \mathbf{ln\ V} \tag{3}$$

Simple linear regression was used to solve for the exponent and log-constant in Eq. 3. Results were as follows:

| | |
|---|---|
| Exponent | 1.5553 |
| 95% Confidence | +/ − 0.0304 |
| ln(constant) | − 1.0432 |
| 95% Confidence | +/ − 0.1410 |

## 5. Comparison of Predicted Lengths Using Halstead Metric vs Herdan-type Metric

First, predicted program length, $\mathbf{N}'$, was obtained for each program in the database using Eq. 1. Next, for each program, percentage error of predicted length vs actual length, $[(\mathbf{N}' - \mathbf{N})/\mathbf{N}] \times \mathbf{100}\%$, was computed. A scattergram of **N** vs $[(\mathbf{N}' \times \mathbf{N})/\mathbf{N}] \times \mathbf{100}\%$ appears in Figure 1 below:

Figure 1 clearly indicates that the Halstead length estimator ‘works best’ in the 400–4000 token range. In fact, in this range, the mean error of estimate is −10.57%. For sample programs that are less than 400 tokens in length, the mean error of estimate is −84.23%,

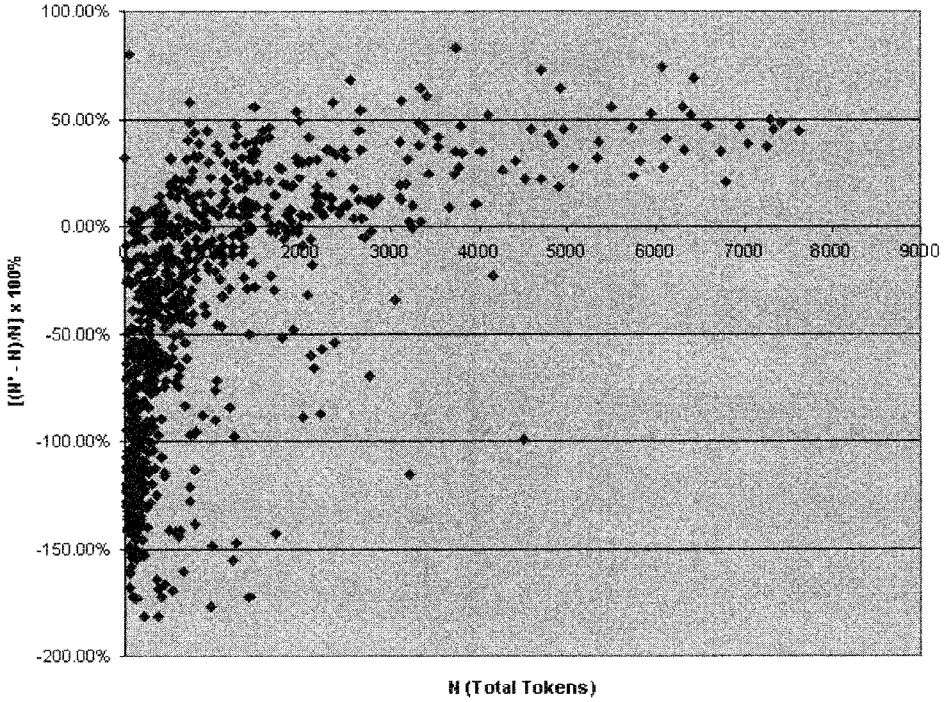

*Figure 1.* Total length (tokens) vs percent error estimate (Halstead).

and for programs whose length is greater than 4000 tokens, the mean error of estimate is 38.77%.

Next, predicted program length, **N**′, for each program in the database was obtained using Eq. 2, with the constant and exponent noted in Section 3. Percent error of estimation, $[(\mathbf{N}' - \mathbf{N})/\mathbf{N}] \times \mathbf{100}\%$, was computed for each program. A scattergram of **N** vs $[(\mathbf{N}' - \mathbf{N})/\mathbf{N}] \times \mathbf{100}\%$ appears in Figure 2 below:

In Figure 2 we observe the correction of the Halstead length estimator skewness for small programs ($<$ 400 tokens). In this range the mean percent error estimate is $-17.30\%$. There is also improvement in the program length range 400–4000 tokens, with a mean percent error estimate of $-1.21\%$. Estimation in the large program size range ($>$ 4000 tokens) improves as well, with a mean percent error estimate of 22.23%.

## 6. Conclusions and Open Issues

The Herdan-type estimator corrects skewness for small programs, and yields better length estimation over the Halstead length estimator for both ‘medium’ sized programs (400–4000 tokens), and ‘large’ programs (no. of tokens $>$ 4000).

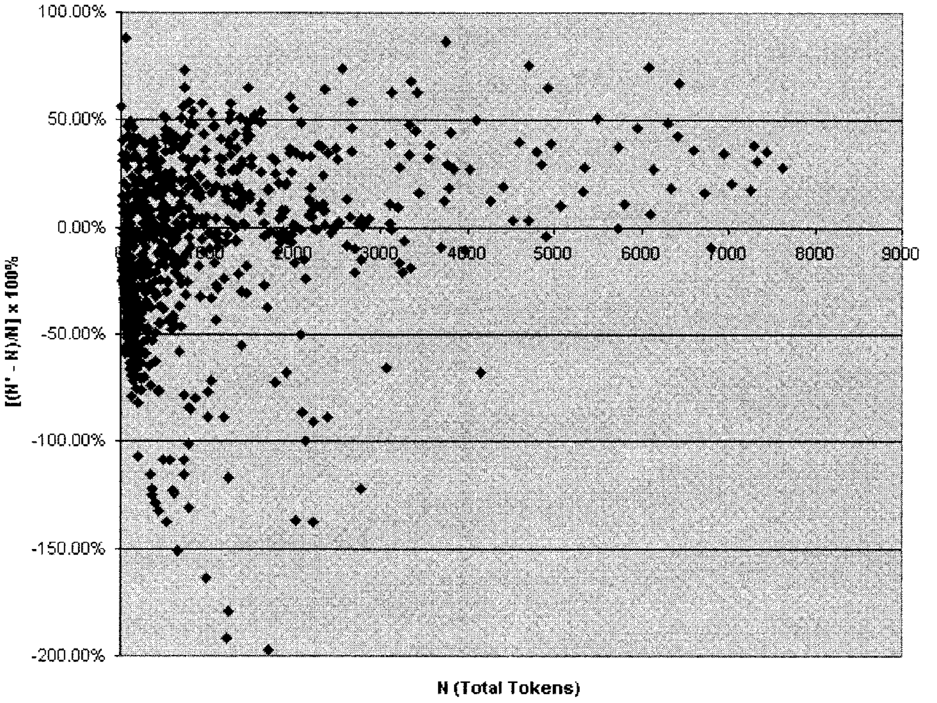

*Figure 2.* Total length (tokens) vs percent error estimate (Herdan).

Consider the exponents and ln(constant) values computed thus far:

| | Exponent | ln(constant) | Language Base |
|---|---|---|---|
| Beser (1983) | 1.5062 | −1.4122 | FORTRAN (included programs to a Length of approx. 3000 tokens) |
| Gonzales (1990) | 1.6876 | −1.0564 | Pascal (‘small’ programs only) |
| Current Study | 1.5553 | −1.0432 | ‘C’ (program lengths from approx. 20–8000 tokens) |

The magnitude of difference between Pascal and ‘C’ exponents and ln(constant) values is smaller than for any other pairwise combination of the three languages Pascal, ‘C’, and FORTRAN. The authors suggest an open issue of whether this occurs because Pascal and ‘C’ are structurally more ‘nearly isomorphic’ than other pairwise combinations of studied languages.

## References

Beser, N. 1983. *Foundations and Experiments in Software Science*. PhD thesis. University of Pennsylvania.

Gonzales, M. 1990. Correction of the Halstead length estimator skew for small Pascal programs. *ACM Sigmetrics Performance Evaluation and Review* 17(2).

Halstead, M. 1977. *Elements of Software Science*. New York: Elsevier North-Holland.
Herdan, G. 1956. *Language as Choice and Chance*. Groningen: Noordhoff.
Herdan, G. 1964. *Quantitative Linguistics*. Washington: Butterworths.
Herdan, G. 1966. *The Advanced Theory of Language as Choice and Chance*. New York: Springer-Verlag.
Herdan, G. 1960. *Type-Token Mathematics*. The Hague: Mouton & Co.
*Complexity Measures Tool for C/C++ (CMT++)*. 1997. Tampere, Finland: Testwell Oy.

**Dr. Gonzales** is currently employed by Motorola, Inc., Broadband Communications Sector, Digital Network Systems, as a Senior Quality Assurance Engineer. Dr. Gonzales joined Motorola about four years ago. Prior to joining Motorola, Dr. Gonzales was a professor of computer science and mathematics for seventeen years. Dr. Gonzales holds Doctor of Philosophy and Master of Science degrees in Computer Science, and Master of Arts and Bachelor of Arts degrees in Mathematics. His research interests include software metrics, reliability theory, and phase transitions and critical phenomena.

**Seshadri Paravastu** was born in Madras, India. He recieved his Bachelor of Engineering [Electrical and Electronics] degree with distinction in 1995 from Andhra University, Waltair (India). Seshadri studied at State of University of New York, Stony Brook, NY, where he was a research assistant for Dr. Green. He received his Master of Science Degree in Electrical Engineering with emphasis in Computer Engineering in 1996. Seshadri is currently a Staff Engineer with Systems Engineering in Motorola, Broadband Communications Sector, Horsham, PA. Seshadri's research interests include Broadband Communications and Software Metrics.