The potency of the efficiency and compression rate of Cirebon language script using the Binary Huffman algorithm

N Ismail¹*, T Yusuf¹, R Fajar¹, P Alqinsi¹, A Kodir² and H Qodim³

¹ Electrical Engineering Department, Faculty of Science and Technology, UIN Sunan Gunung Djati Bandung, Jalan A.H Nasution No. 105, Cibiru, Bandung, Indonesia
² Language Center, UIN Sunan Gunung Djati Bandung, Jalan A.H Nasution No. 105, Cibiru, Bandung, Indonesia
³ LP2M, UIN Sunan Gunung Djati Bandung, Jalan A.H Nasution No. 105, Cibiru, Bandung, Indonesia

*nanang.is@uinsgd.ac.id

Abstract. Each language usually has several important features that are hidden statistically and certain redundancy. These features can be utilized to perform text compression that is suitable for the optimal use of resources. This study proposes an analysis the potency of the efficiency and compression rate of Cirebon language script using the Binary Huffman algorithm. The analysis of the potency is based on the entropy of the Cirebon language script. The study begins with an analysis of the Cirebon language script to calculate the probability of each symbol. These probabilities are used to calculate the value of entropy. The results showed that the entropy of the Cirebon language script was 3.976 bits per symbol, with an expected code length of 4.02 bits per symbol. Then, the estimated efficiency of compression with the Binary Huffman Code is 98.89% and the compression rate is 0.80402. These results can be considered in the use of Cirebon language compression for the telecommunications transmission process.

1. Introduction

In a language, there are symbols that appear more often than other symbols or symbol pairs. Frequently or not a symbol or symbol pair appears, it can be seen by looking at the frequency distribution [1]. Language has potential in the application of communication technology, especially related to source compression [2]. In his paper, Khoirul et al analyzed the potential of source compression for Indonesian local languages namely Sundanese, Javanese, and Balinese based on outage probability using the Slepian-Wolf coding theorem [2].

Source compression can provide changes to the size of the data that can reduce data duplication so that the data representation becomes less than the original data [2,3]. The provision of source compression also answers the challenges of the future telecommunications needs which are expected to bring up many beneficial applications [4].

Information technology exchange in the future is expected to be more effective and efficient that can provide changes for better speed. Characters and grammar in a language usually differ from one language to another. The frequency of the appearance of one language symbol can be more than another symbol [1,5,6]. Because of this nature, every language has the potential for efficiency to be used in future communication.
This paper discusses the potency of the efficiency and compression rate of Cirebon language script using the Binary Huffman algorithm. Cirebon language was chosen because it is a regional language that has slices with Sundanese and Javanese but still has its own uniqueness. Besides the Huffman Code, there are several other methods for data compression including Arithmetic, LZW, and Run Length Encoding [7-9].

Huffman codes are used in research because Huffman has decodable properties [10,11]. Decodable means that the data that has been compressed can be returned to the initial data [2,7]. The aim of this research is finding out entropy, labeling patterns using the Huffman algorithm code to get the expected length, and efficiency as an analytical material to find out the compression level of Cirebon language text. It is expected that in the future the results obtained can participate in developing information theory in the planning of language communication in future communication.

2. Method
The study began with a static analysis of Cirebon text. The total characters/symbols sampled totaled 1544 consisting of 25 types of symbols that appear with different frequencies. Probability is calculated by

\[
Prob = \frac{\text{frequency symbol}}{\text{total symbol}}
\]  

(1)

Entropy calculations are performed using:

\[
H(X) = -\sum_{x \in X} p(x) \log_2 p(x),
\]  

(2)

where \(p(x)\) is the probability of the symbol appearing.

The next stage is making binary trees based on the Huffman Code Algorithm. The two symbols with the lowest probability add up to form a new node. Repeat the same thing for the other symbols. The sum is done for two symbols or nodes that have the same probability value or the upper probability value is higher than the lower probability value. These nodes are added together with other nodes until a complete Huffman tree is obtained with the number of probabilities 1. The codeword for each symbol is performed by giving a value of 0 for the upper branch and giving a value of 1 for the lower branch. Giving a codeword for each symbol is performed by taking a sequence of bits from the root to the leaf. The results of making binary trees are codewords and length codes (bit lengths of codewords) for each symbol in Cirebon script.

The expected code length calculation \(L(C)\) is performed using

\[
L(C) = \sum_{x \in X} l(x) p(x),
\]  

(3)

where \(l(x)\) is the code length and \(p(x)\) is the probability of the symbol in Cirebon script.

Efficiency \(\eta\) is obtained by comparing the entropy results with the expected code length that has been obtained previously using

\[
\eta = \frac{H(X)}{L(C)} \times 100\%.
\]  

(4)

Compression rate calculations are performed by comparing the expected code length \(L(C)\) of the encoded text with the standard expected length \(L(n)\)

\[
R_c = \frac{L(C)}{L(n)},
\]  

(5)

where the standard expected length \(L(n)\) is a value that can cover all symbols in the text used for Cirebon language is \(\approx 5\).
3. Results and analysis
From the number of frequencies ($F$) of each symbol that appears and the probability ($P$) of each character, an entropy value ($H$) will be obtained. Codeword ($C$) and code length $l(x)$ will be obtained from the Huffman Tree as seen in Figure 1. While expected code length $L(C)$ is obtained from the number of multiplications between the probability ($P$) of each character and codeword length $L(x)$.

![Huffman Tree](image)

**Figure 1.** Binary tree Huffman code.

Based on the Huffman Tree, obtained data contained in Table 1. In this study, the symbols "A", "SPACE", and "n" have the highest probability values, so that they have the shortest length codes of 3 bits with codeword 000, 001, and 110. While the symbols "c" and "." have the lowest probability value so it has the longest length code of 9 bits with codeword 010110110, and 010110111. Entropy value is obtained using Equation (2).

\[
H = 0.166 \times \log_2(1/0.166) + 0.158 \times \log_2(1/0.158) + \ldots + 0.001 \times \log_2(1/0.001)
\]

\[
= 3.976 \text{ bits per symbol.}
\]
Table 1. Analysis result data.

| Symbol | F(x) | P(x) | H(x) | l(x) | C   | L(C) |
|--------|------|------|------|------|-----|------|
| A      | 257  | 0.166| 0.431| 3    | 000 | 0.499|
| Space  | 244  | 0.158| 0.421| 3    | 001 | 0.474|
| N      | 141  | 0.091| 0.315| 3    | 110 | 0.274|
| e      | 125  | 0.081| 0.294| 4    | 0100| 0.324|
| i      | 98   | 0.063| 0.252| 4    | 0111| 0.254|
| r      | 94   | 0.061| 0.246| 4    | 1000| 0.244|
| g      | 70   | 0.045| 0.202| 4    | 1011| 0.181|
| s      | 66   | 0.043| 0.194| 4    | 1110| 0.171|
| u      | 55   | 0.036| 0.171| 5    | 01010| 0.178|
| k      | 52   | 0.034| 0.165| 5    | 01100| 0.168|
| w      | 48   | 0.031| 0.156| 5    | 10010| 0.155|
| d      | 41   | 0.027| 0.139| 5    | 10100| 0.133|
| m      | 40   | 0.026| 0.137| 5    | 10101| 0.130|
| l      | 33   | 0.021| 0.119| 5    | 11110| 0.107|
| t      | 27   | 0.017| 0.102| 6    | 010111| 0.105|
| b      | 26   | 0.017| 0.099| 6    | 011010| 0.101|
| p      | 26   | 0.017| 0.099| 6    | 011011| 0.101|
| o      | 24   | 0.016| 0.093| 6    | 100110| 0.093|
| .      | 19   | 0.012| 0.078| 6    | 100111| 0.074|
| y      | 16   | 0.010| 0.068| 6    | 111110| 0.062|
| ,      | 15   | 0.010| 0.065| 6    | 111111| 0.058|
| h      | 14   | 0.009| 0.062| 7    | 0101100| 0.063|
| j      | 9    | 0.006| 0.043| 8    | 01011010| 0.047|
| c      | 3    | 0.002| 0.018| 9    | 010110110| 0.017|
| -      | 1    | 0.001| 0.007| 9    | 010110111| 0.006|
| Total  | 1544 | 1     | 3.976| -    | -   | 4.020|

Efficiency is obtained by Equation (4)

\[
\text{Efficiency} = \frac{3.976}{4.02} \times 100\% = 98.89\%\]

In this study, the expected original file length code refers to the standard equal length value of 5 bits. A value of 5 is enough to cover the total number of symbols which are only 25 symbols. Thus, using Equation (5), the Compression Rates are equal to

\[
R_c = \frac{4.020}{5} = 0.80402.
\]

4. Conclusion

Entropy Cirebon language script with the Binary Huffman Code of 3.976 bits per symbol, with an expected code length of 4.020 bits per symbol. So, the estimated efficiency of compression with the Binary Huffman Code is 98.89% and the compression rate is 0.80402. These results can be considered in the use of Cirebon language compression for the telecommunications transmission process.
Acknowledgments
On this occasion, the authors would like to thank the Research and Publishing Center of UIN SGD Bandung for its research funding and publication.

References
[1] Shelly A and Livianthy Indah L 2011 Analisis Metode Huffman untuk Kompresi Data Citra dan Teks pada Aplikasi Komresi Data (Palembang: STMIK GI MDP)
[2] Anwar K, Baihaqi R F and Julian Y 2017 Source Coding-based Compressions of Indonesian Local Languages for 5G Potential Applications International Symposium on Electronics and Smart Devices (ISESD) 158-162
[3] Fauzan S, Saepulloh M, Ismail N and Hamidi E A Z 2018 Entropy of Sundanese and Javanese Script for Text Compression Using Binary Huffman Code Algorithm Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati Bandung 248-255
[4] Kokovin V A, Uvayssova S S and Uvaysov S U 2016 Lossless compression algorithm for use in telecommunication systems International Siberian Conference on Control and Communications (SIBCON) 1-4
[5] Kuruvila M and Gopinath D P 2014 Entropy of Malayalam Language and Text Compression Using Huffman Coding Computational Systems and Communications (ICCSA), First International Conference on IEEE 150-155
[6] Dhandra B and Hangarge M 2007 Global and Local Features Based Handwritten Text Words and Numerals Script Identification 2007 International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007) 471-475
[7] Sharma M 2010 Compression Using Huffman Coding IJCSNS International Journal of Computer Science and Network Security 10 5 133-141
[8] Aulí-Llinàs F 2015 Context-Adaptive Binary Arithmetic Coding With Fixed-Length Codewords IEEE Transactions on Multimedia 17 8 1385-1390
[9] Malik A, Sikka G and Verma H K 2017 A high capacity text steganography scheme based on LZW compression and color coding Engineering Science and Technology Journal 20 1 72-79
[10] Shoba D J and Sivakumar S 2017 A Study on Data Compression Using Huffman Coding Algorithms International Journal of Computer Science Trends and Technology (IJCST) 5 1 58-63
[11] Gheraibia Y, Kabir S, Moussaoui A and Mazouzi S 2018 Optimised cost considering Huffman code for biological data compression International Journal of Information and Communication Technology 13 3 275–290