An entropy analysis of the Cirebon language script using the Ternary Huffman code algorithm

A Kodiri\textsuperscript{1}, R Fajar\textsuperscript{2}, A S Awalluddin\textsuperscript{3}, U Ruswandi\textsuperscript{4}, N Ismail\textsuperscript{2,}\* and D Miharja\textsuperscript{5}

\textsuperscript{1} Language Center, UIN Sunan Gunung Djati Bandung, Jalan A.H. Nasution No. 105, Cibiru, Bandung, Indonesia
\textsuperscript{2} Electrical Engineering Department, Faculty of Science and Technology, UIN Sunan Gunung Djati Bandung, Jalan A.H. Nasution No. 105, Cibiru, Bandung, Indonesia
\textsuperscript{3} Mathemetic Department, Faculty of Science and Technology, UIN Sunan Gunung Djati Bandung, Jalan A.H Nasution No. 105, Cibiru, Bandung, Indonesia
\textsuperscript{4} Graduate School, UIN Sunan Gunung Djati Bandung, Jalan A.H. Nasution No. 105, Cibiru, Bandung, Indonesia
\textsuperscript{5} Center of Research and Publishing, UIN Sunan Gunung Djati Bandung, Jalan A.H. Nasution No. 105, Cibiru, Bandung, Indonesia

*nanang.is@uinsgd.ac.id

Abstract. Entropy is a statistical parameter that measures how much average information is generated for each symbol in a text. Each language usually has several important features that are hidden statistically and certain redundancy. These features can be utilized to form appropriate text compression tools for optimal use of resources. This study proposes an analysis of the entropy of the Cirebon language text for text compression using the Ternary Huffman Code algorithm. This entropy value then becomes the reference level of the Cirebon script compression level. The probability of each symbol in the Cirebon Regional Text is used to calculate the entropy value. The result shows the entropy of the Cirebon language script was 2.508 bits per symbol, with an expected code length of 2.565 bits per symbol. Estimated compression efficiency with Ternary Huffman Code is 97.77% and compression rate is 0.51308.

1. Introduction
At this time, the process of sending information directly is still often facing problems, one of them is caused by the large size of the data sent. Besides, storage media has a limited capacity. If the data to be stored on storage media is large, then the data cannot be saved [1].

The method of transmitting an information symbol sequence is very important because each message is ensured to use more than one symbol. If it is assumed that each symbol requires the same time for transmission, the time for transmitting messages is directly proportional to the number of symbols associated with it [2]. Each communication system follows the general process of transmitting messages from one point to another. The hidden statistical nature of the communication process was first recognized by mathematician, Claude Elwood Shannon. One of the most important features of Shannon's theory is the concept of entropy. With that concept, many sentences can be abbreviated without losing their meaning [3].
Entropy is a statistical parameter that measures how much information is produced on average for each letter of a text in a language. Each language usually has some important features that are hidden statistically and certain redundancy. These features can be utilized as a means of text compression that is suitable for the optimal use of resources [4].

Data compression is a technique to reduce the data size from the original data to be more effective and smaller in storage, efficient and faster in the process of transmitting data [1]. Data compression consists of two main processes, namely compression, and decompression. If a file is compressed, then the file must be read again after the file is decompressed [5]. There are two types of data compression, namely lossless data compression and lossy data compression [1,6,7].

There are many methods for data compression, including Huffman, Shannon, Shannon-Fano, Run-length Encoding, Run-length Huffman, Lempel-Zip-Welch (LZW), and several other methods [1,8-10]. One decodable compression method is the Huffman code [11]. The Huffman code method uses the principle that each character is coded with a series of several bits. The characters that appear often will be given the shortest bit set while the characters that rarely appear will be given the longest bit series [12].

This paper discusses the efficiency and compression rate of the Cirebon language script using Ternary Huffman code based on entropy. Cirebon is chosen because it is a regional language that has intersection of Sundanese and Javanese. This entropy value then becomes the reference level of the Cirebon script compression level. The research results can be used as a basis for developing local language-based compression applications.

2. Method
The study began with a statistical analysis of the Cirebon language script. The total characters/symbols sampled are 1544 consisting of 25 types of symbols that appear with different frequencies. Probability P is calculated by:

\[ P = \frac{F}{N} \]  

where \( P \) is the probability of symbol, \( F \) is the frequency of symbol, and \( N \) is the total of symbol.

After the frequency and occurrence probability of each symbol are obtained, then the coding is done with the Ternary Huffman Code algorithm. Huffman Code was built using a tree structure with three branches. These tree structure is called a ternary tree. First of all, the probability data from the text is sorted from the largest to the smallest. The three symbols that have the smallest probability are then combined to form a new node. The sum is done for 3 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 is 1. Codeword formation for each symbol is done by giving a value of 0 for the upper branch, 1 for the middle branch, and 2 for the lower branch. Giving a codeword for each symbol is done by taking a sequence of bits from the root to the leaves [1].

From the frequencies (\( F \)) of each symbol that appears and the probability (\( P \)) of each symbol, entropy (\( H \)), Codeword (\( C \)) and codeword length \( l(x) \) can be obtained, which can ultimately be used to calculate efficiency and compression rate of a Cirebon manuscript.

3. Results and analysis

3.1. Ternary Huffman code
The Ternary Huffman tree is developed based on the frequency and probability of each symbol that has been analyzed. The Ternary Huffman tree can be seen in Figure 4. Meanwhile, the expected code length \( L(C) \) is obtained from the number of multiplications between the probability and the codeword length of each character.
The 5th Annual Applied Science and Engineering Conference (AASEC 2020)  
IOP Conf. Series: Materials Science and Engineering 1098 (2021) 042042  
doi:10.1088/1757-899X/1098/4/042042

Figure 1. Ternary tree Huffman code.

The rules for using the Huffman Tree are similar to the Binary Huffman Code algorithm but they are based on 3 symbols. Entropy calculations are performed using a base 3 logarithmic equation:

\[ H(X) = - \sum_{x \in X} p(x) \log_{3} p(x) \]  

\[ = 0.166 \times \log_{3} 1/0.166 + 0.158 \times \log_{3} 1/0.158 + \ldots + 0.001 \times \log_{3} 1/0.001 \]  

\[ = 2.508 \text{ bits per symbol}. \]

Expected code length \( L(C) \) calculation is done using:

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

\[ = (2 \times 0.166) + (2 \times 0.158) + (2 \times 0.091) + (2 \times 0.081) + \ldots + (6 \times 0.001) \]  

\[ = 2.565 \text{ bits per symbol}. \]

Efficiency \( \eta \) calculations are performed using:

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

\[ = \frac{2.508}{2.565} \times 100\% \]  

\[ = 97.77\%. \]
The complete statistical analysis data can be seen in Table 1.

### Table 1. Entropy and expected code length.

| Symbol | F(x) | P(x) | H(x) | l(x) | C | L(C) |
|--------|------|------|------|------|---|------|
| a      | 257  | 0.166| 0.272| 2    | 00| 0.333|
| space  | 244  | 0.158| 0.265| 2    | 01| 0.316|
| n      | 141  | 0.091| 0.199| 2    | 02| 0.183|
| e      | 125  | 0.081| 0.185| 2    | 10| 0.162|
| i      | 98   | 0.063| 0.159| 2    | 11| 0.127|
| r      | 94   | 0.061| 0.155| 3    | 200| 0.183|
| g      | 70   | 0.045| 0.128| 3    | 201| 0.136|
| s      | 66   | 0.043| 0.123| 3    | 210| 0.128|
| u      | 55   | 0.036| 0.108| 3    | 211| 0.107|
| k      | 52   | 0.034| 0.104| 3    | 220| 0.101|
| w      | 48   | 0.031| 0.098| 3    | 221| 0.093|
| d      | 41   | 0.027| 0.088| 3    | 120| 0.080|
| m      | 40   | 0.026| 0.086| 3    | 121| 0.078|
| t      | 33   | 0.021| 0.075| 3    | 122| 0.064|
| t      | 27   | 0.017| 0.064| 4    | 2020| 0.070|
| b      | 26   | 0.017| 0.063| 4    | 2021| 0.067|
| p      | 26   | 0.017| 0.063| 4    | 2022| 0.067|
| o      | 24   | 0.016| 0.059| 4    | 2120| 0.062|
| .      | 19   | 0.012| 0.049| 4    | 2121| 0.049|
| y      | 16   | 0.010| 0.043| 4    | 2122| 0.041|
| ,      | 15   | 0.010| 0.041| 4    | 2220| 0.039|
| h      | 14   | 0.009| 0.039| 4    | 2221| 0.036|
| j      | 9    | 0.006| 0.027| 5    | 22220| 0.029|
| c      | 3    | 0.002| 0.011| 5    | 22221| 0.010|
| -      | 1    | 0.001| 0.004| 6    | 222220| 0.004|
| Total  | 1544 | 1     | 2.508|      |    | 2.565|

#### 3.2. Results analysis of ternary Huffman code

In this study, the total symbols sampled were 1544 symbols with 25 symbols appearing. The symbols "A", "SPACE", "n", "e", and "I" have the highest probability values, so that they have the shortest code length of 2 bits with successive codewords 00, 01, 02, 10, and 11. While the symbol "," has the lowest probability value, so it has the longest code length of 6 bits with codeword 222220. Entropy obtained is 2.508 bits per symbol with an expected code length of 2.565 bits per symbol and an efficiency of 97.77%. The analysis only considers the symbols that appear. Compression rates $R_c$ is calculated by comparing the expected code length $L(C)$ resulting from the Huffman codes with the expected code length of the original file $L(n)$ using

$$R_c = \frac{L(C)}{L(n)}$$  \hspace{1cm} (5)

In this study, the expected code length of the original file refers to a standard equal length value of 5 bits. A value of 5 is enough to cover the total number of symbols which is only 25 symbols, so, using equation (5) we can calculate the compression rate...
\[ R_c = \frac{2.565}{5} = 0.51308. \]

4. Conclusion

Based on the analysis that has been done, the entropy of the Cirebon language script with Ternary Huffman Code is 2.508 bits per symbol, with an expected code length of 2.565 bits per symbol. So that the estimated efficiency of compression with Ternary Huffman Code is 97.77% and the compression rate is 0.51308. These results can be considered in the use of Cirebon language compression for the telecommunications transmission process.

Acknowledgments

On this occasion, we present our gratitude to the Research and Publishing Centre of UIN SGD Bandung for its funding assistance.

References

[1] Shelly A and Livianthy Indah L 2011 Analysis of Huffman Methode for Compression of Image and Text Data in Data Compression Application (in Bahasa) (Palembang: STMIK GI MDP)
[2] Huffman D A 1952 A Method for the Construction of Minimum-Redundancy Codes *Proceedings of the IRE* **40** 9 1098-1101
[3] Shannon C E 2001 A mathematical theory of communication *ACM SIGMOBILE Mobile Computing and Communications Review* **5** 1 3-55
[4] Kuruvila M and Gopinath D P 2014 Entropy of Malayalam Language and Text Compression Using Huffman Coding *in Computational Systems and Communications (ICCSC), First International Conference on*. IEEE 150-155
[5] 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
[6] Nemati K and Ramakrishnan K 2017 Hybrid lossless and lossy compression technique for ECG signals *Third International Conference on Sensing, Signal Processing and Security (ICSSS)* 450-455
[7] Jaferzadeh K, Gholami S and Moon I 2016 Lossless and lossy compression of quantitative phase images of red blood cells obtained by digital holographic imaging *Applied Optics* **55** 36 10409-10416
[8] Sharma M 2010 Compression Using Huffman Coding *IJCSNS International Journal of Computer Science and Network Security* **10** 5 133-141
[9] Auli-Llinàs F 2015 Context-Adaptive Binary Arithmetic Coding With Fixed-Length Codewords *IEEE Transactions on Multimedia* **17** 8 1385-1390
[10] 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
[11] Anwar K, Baihaqi R F and Julian Y 2017 Source Coding-based Compressions of Indonesian Local Languages for 5G Potential Applications *in International Symposium on Electronics and Smart Devices (ISESD)* 158-162
[12] Gheraibia Y, Kabir S, Moussaouli 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