

Improvement Text Compression Performance Using Combination of Burrows Wheeler Transform, Move to Front, and Huffman Coding Methods

Mohammada Aprilianto¹ and Maman Abdurrohman²

Telkom University

Jl. Telekomunikasi, 1 Bandung, West Java, Indonesia

E-mail: mohammada.aprilianto@yahoo.com

Abstract. Text is a media that is often used to convey information in both wired and wireless-based network. One limitation of the wireless system is the network bandwidth. In this study we implemented a text compression application with lossless compression technique using combination of Burrows wheeler transform, move to front, and Huffman coding methods. With the addition of the compression of the text, it is expected to save network resources. This application provides information about compression ratio. From the testing process, it concludes that text compression with only Huffman coding method will be efficient when the number of text characters are above 400 characters, meanwhile text compression with burrows wheeler transform, move to front, and Huffman coding methods will be efficient when the number of text characters are above 531 characters. Combination of these methods are more efficient than just Huffman coding when the number of text characters are above 979 characters. The more characters that are compressed and the more patterns of the same symbol, the better the compression ratio.

1. Introduction

Compression is a technique to shrink the size of data. There are two kinds categories in compression based on the output, which is lossless and lossy compression. In lossless compression, the outcome data after decompression is exactly as before compression. While on lossy compression, there is the possibility of losing some data after decompression [4].

This compression can also be applied to SMS (Short Message Service). SMS that delivered by mobile phone to SMS Centre is sent in PDU (Protocol Data Unit) format. As well SMS that received by mobile phone from SMS Centre is saved on mobile phone in PDU format [11].

Burrows wheeler transform change the order of the characters so that there will be more redundancy characters, to take advantage of it, so it is forwarded by move to front method so that there will be many index zero as recurring characters will be placed in index zero by this method [6]. Due to the large frequency of zeros, Huffman coding will be used which will transform the character into a shorter codeword. The greater the frequency of characters, the shorter the codeword.

2. Research Method

In This research try to improve text compression performance by using burrows wheeler transform, move to front, and Huffman coding. After that, the author analysed the compression ratio. This



research is done by making a text compression application using Net Beans 7. The Huffman Coding frequency table will be in the text. Here is the explanation about the methods and the system design.

2.1. Burrows Wheeler Transform

Burrows wheeler transform is a reversible transformation algorithm. This algorithm is used in lossless compression like bzip2 [8]. This method just change the order of characters, so when the data is reverted, the outcome will be just like before. In this algorithm the author will use text 'title' for the example. Perform shift one character to the right. Repeat until (n-1) times where n is the sum of characters on the input data. The result would be like in figure 1.

	title
	etitl
	letit
	tleti
	itlet

Figure 1. Burrows Wheeler Transform First Step.

After that sort those words in lexicography and the result would be on figure 2.

1	etitl
2	itlet
3	letit
4	title
5	tleti

Figure 2. Burrows Wheeler Transform Second Step

The output of this transformation is the last character in every line from top to bottom and the line number where the original text is, so the output is 'lttei, 4'.

2.2. Move to Front

Move to front is a data transformation algorithm that is designed to improve the performance of entropy encoding compression technique. The idea of this algorithm is change every symbol with an index from a recent list. A symbol that shows often will be changed with index zero which will be make an advantage to entropy encoding compression technique like Huffman coding [5]. In this algorithm, first we need to make a list of all possible character that will be shown from the input data, for example of recent list is 'abcde'. The left character has index zero, move to right, and the index increase by one. For example, the first input data is character 'b', character 'b' is on index one, so write 1 on the output, then move 'b' on the recent list to the most left.

2.3. Huffman Coding

Huffman coding is a compression algorithm founded by David Huffman [9]. This algorithm works by building a tree based on data frequency, the more the frequency of a character, the shorter the codeword. The codeword is in binary form. First we need to calculate the frequency of every character that is shown from the input data. Then make a tree by take two characters with lowest frequency first. Mark the left leaf of tree with '0' and the right leaf with '1'. After that, make a codeword for a

character by browsing where the leaf of that character from the root is. Finally, change every character on the input with codeword that has been made.

2.4 System Design.

Here is the flowchart of compression process.

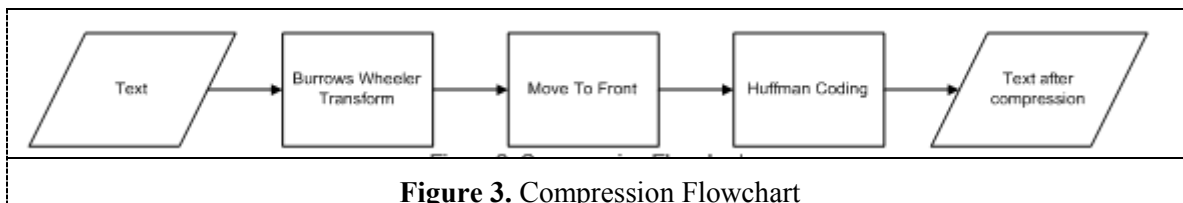


Figure 3. Compression Flowchart

In this application, the text is first transformed by burrows wheeler transform method so the arrangement of the text is changed with more redundancy characters. After that, the text is processed by move to front method. Each character will be changed to index number where recent character that showed will be change to index 0. In the end, Huffman coding will compress the text that has been changed to index number.

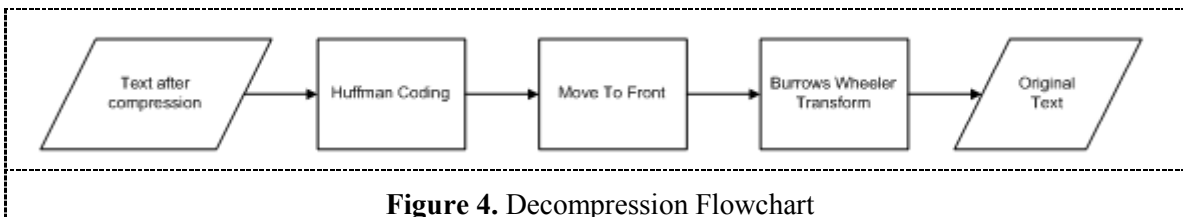


Figure 4. Decompression Flowchart

The decompression process is first change the text to index number with Huffman coding method, after that move to front will change back the index number back to text that has been reordered from the original text, and then burrows wheeler transform will change its text to original text before the compression.

3. Result and Analysis

This research analyse the compression ratio where compression ratio is ratio of the number of characters after compression by the number of characters before compression multiply 100%. This research used two scenarios, first scenario focused on number of characters. There were 80 number of texts that was tested. 40 texts were regular sentences, and the other 40 texts were sentences with more complicated characters. Second scenario focused on the pattern of characters used. Analysis based on worst case, regular text, and best case. Worst case has one frequency of every characters. Regular text has an increasing same pattern of characters. The best case has one kind character with 265 frequencies.

3.1. Analysis of Parameter Compression Ratio Testing

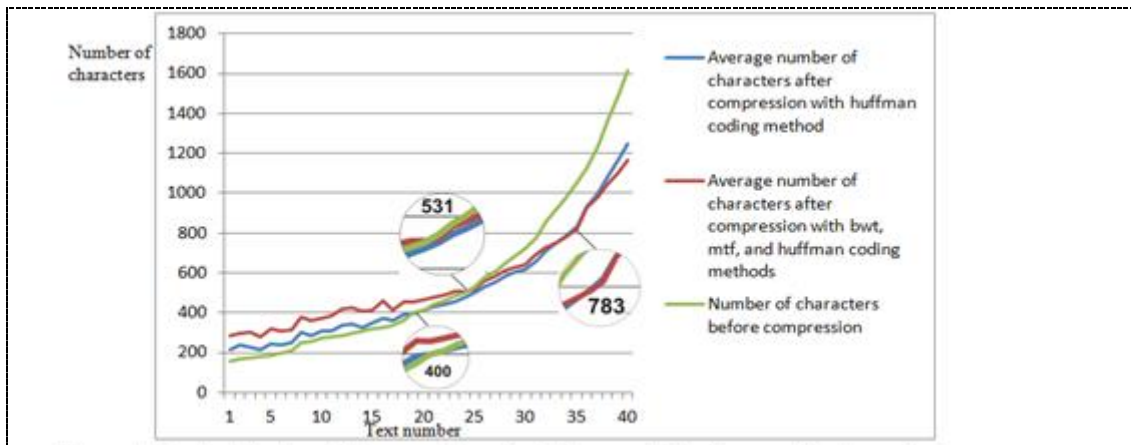


Figure 5. Graph of Number of Average Characters Before and After Compression Scenario 1

Figure 5 above is a graph about number of average characters from text with regular sentences and complicated sentences before and after compression scenario 1. The graph tells that the Huffman Coding compression method compress the text start from number of 400 characters, while the BWT, MTF, and Huffman Coding compression method compressed the text start from number of 531 characters. The BWT, MTF, and Huffman Coding compression method is better when the number of character are exceed 979 characters. At the number of 500 characters and more, the decompression time exceed 30 minutes.

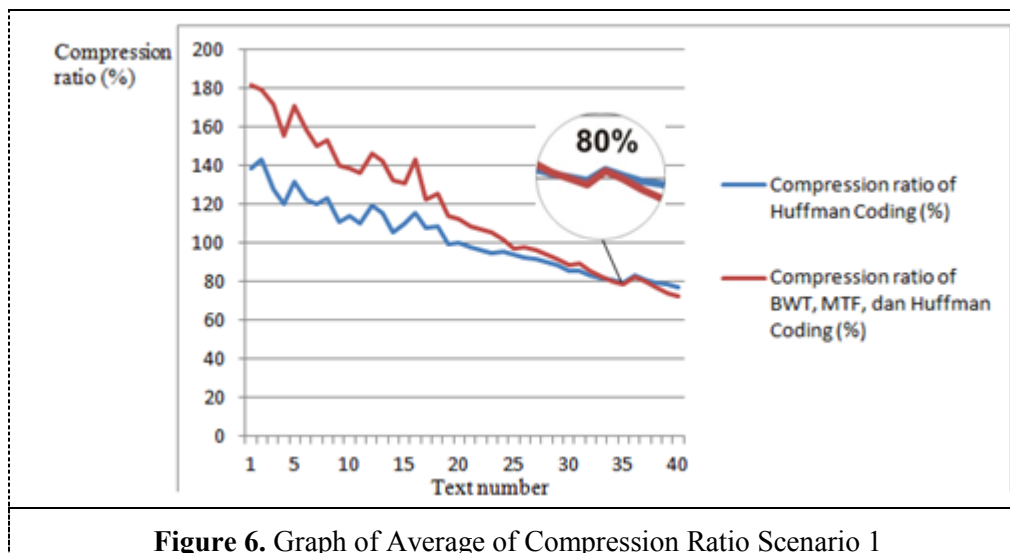


Figure 6. Graph of Average of Compression Ratio Scenario 1

Figure 6 above is a graph about average of compression ratio scenario 1. The graph tells that the more characters compressed, the better the compression ratio is, and the ratio of BWT, MTF, and Huffman Coding Compression Method is better than only Huffman Coding when it achieves 80%.

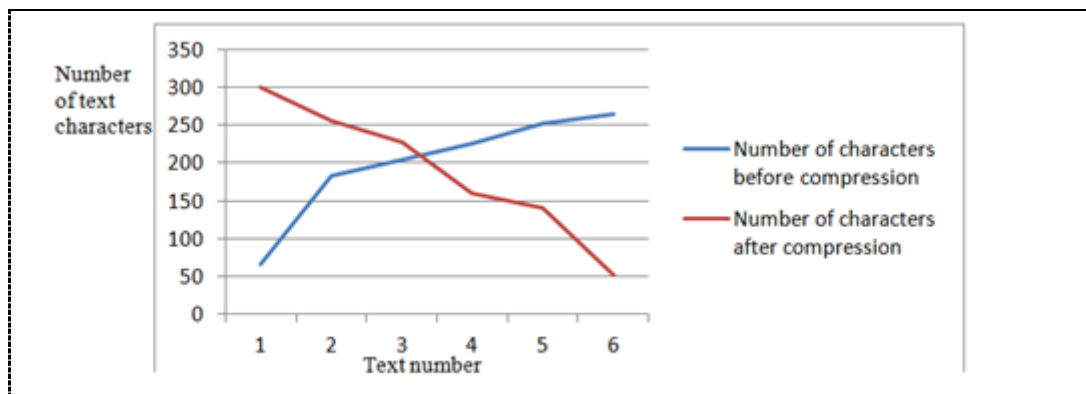


Figure 7. Graph Number of Character Before and After Compression Scenario 2

Figure 7 tells that number of characters after the compression on the worst case is increase than the original. Besides, the number of characters after the compression on the best case is greatly reduce than the original.

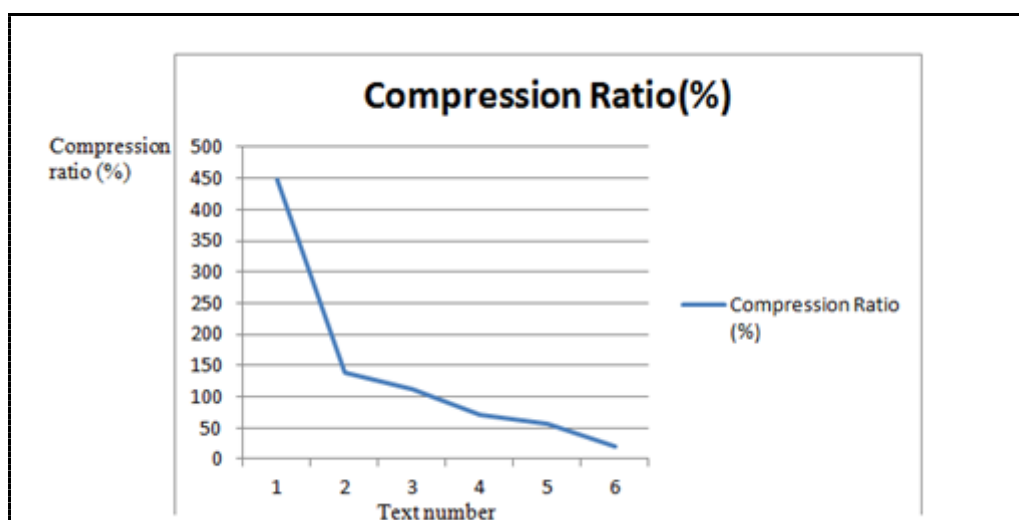


Figure 8. Graph of Compression Ratio Scenario 2

Figure 6 tells that the compression ratio is reducing, the more the number of same pattern of characters, the better the compression ratio is.

4. Conclusion

From the testing results it can be concluded that text compression with BWT, MTF, and Huffman Coding method will compress the text when the number of characters exceed 531 characters, while text compression with only Huffman Coding method will compress the text when the number of characters exceed 400. Combination of these methods are more efficient than just Huffman coding when the number of text characters are above 979 characters. The pattern of characters influence the compression ratio, the more the same pattern of characters, the better the compression ratio is.

For next research, this application can be developed with another method or using some method so the Huffman frequency table is not necessary to be included in text so the compression ratio will be better.

References

- [1] Abdullah, M.M. Kompresi String Menggunakan Algoritma LZW dan Huffman. Bandung. 2008
- [2] Amrullah, A., (et.al). Kompresi dan Enkripsi SMS dengan Metode Huffman Code dan Algoritma Enigma. Institut Teknologi Sepuluh Nopember.
- [3] Ayuningtyas, N. Implementasi Kode Huffman dalam Aplikasi Kompresi Teks pada Layanan SMS. Bandung.
- [4] Blelloch, G. E. Introduction to Data Compression. Camegie Mellon University. 2010.
- [5] Burrows, M., Wheeler, D.J. A Block-sorting Lossless Data Compression Algorithm. California. 1994.
- [6] Effros, M., Visweswariah, K., Kulkarni, S. R., Verdu, S. Universal Lossless Source Coding With the Burrows Wheeler Transform. 2002.
- [7] Mahmoud, T. M., Abdel-latef, B. A., Ahmed, A. A., Mahfouz, A. M. Hybrid Compression Encryption Technique for Securing SMS. Minia University.
- [8] Manzini, G. The Burrows-Wheeler Transform: Theory and Practice. Italy.
- [9] Patra, N., Sankar, S.S. Data Reduction by Huffman Coding and Encryption by Insertion of Shuffled Cyclic Redundancy Code. Rourkela. 2007.
- [10] Riswan. Mengenal SMS (Short Message Service). 2006.
- [11] Achmad, B. (et.al). Sistem Alarm Mobil Menggunakan Mikrokontroler AT89S52 Berbasis SMS. TELKOMNIKA. April 2008; 6(1): 15-20.