A Review of Various Data Compression Techniques to form a New Technique for Text Data Compression

Manjeet Kaur
M.Tech Student
CSE Department
Gurukashi University
Talwandi Sabo.
Manjeetsran24@gmail.com

Er. Upasna Garg
Assistant Professor
CSE Department
Gurukashi University
Talwandi Sabo.
upasnagarg@gmail.com

Abstract

Data Compression is a strategy for encoding decides that permits considerable diminishment in the aggregate number of bits to store or transmit a document. Transmission of large quantity of data cost more money. Hence choosing the best data compression algorithm is really important. In addition to different compression technologies and methodologies, selection of a good data compression tool is most important. There is a complete range of different data compression techniques available both online and offline working such that it becomes really difficult to choose which technique serves the best. In this paper we represent number of techniques to compress and decompress the text data.

1. INTRODUCTION

An information pressure calculation makes an interpretation of a data article to a compacted grouping of yield images, from which the first data can be recouped with a coordinating decompression calculation. Compressors (and their coordinating decompressors) are planned with the objective that the compacted yield is, by and large, less expensive to store or transmit than the first information. For example if one wants to store a large data file, it may be preferable to first compress it to a smaller size to save the storage space.

Also compressed files are much more easily exchanged over the internet since they upload and download much faster. We require the capacity to reconstitute the first record from the compacted rendition whenever. Information pressure is a system for encoding decides that permits considerable decrease in the aggregate number of bits to store or transmit a document. The more data being managed, the more it expenses regarding stockpiling and transmission costs. To put it plainly, Data Compression is the procedure of encoding information to less bits than the first representation so it consumes less storage space and less transmission time while conveying more than a system.

Data compression algorithms are classified in two ways i.e. lossy and lossless data compression algorithm. A compression algorithm is utilized to change over information from a simple to-utilize arrangement to one advanced for smallness. In like manner, an uncompressing system gives back the data to its unique structure.

1.2 TYPES OF DATA COMPRESSION

As of now, two essential classes of Data Compression are connected in diverse areas. One of these is lossy Data Compression, which is generally used to pack picture information documents for correspondence or files purposes. The other is lossless data compression that is regularly used to transmit or file content or parallel records needed to keep their data in place whenever. Data Compression algorithm can be classified in two ways:

- Lossy Data Compression
- Lossless Data Compression

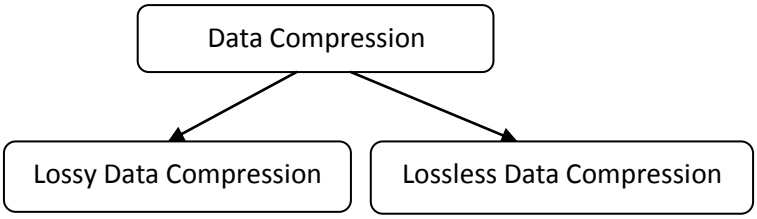

Fig: 1.3 Classification of Data Compression

1.2.1 Lossy data compression

A lossy data compression system is one where the data recovers after decompression may not be precisely same as the first data, but rather is "sufficiently close" to be valuable for particular

reason. After one applies lossy data compression to a message, the message can never be recuperated precisely as it was before it was packed. At the point when the compacted message is decoded it doesn't give back the first message. Data has been lost. Since lossy compression can't be decoded to yield the definite unique message, it is not a decent system for compression for basic data, for example, printed data. It is most valuable for Digitally Sampled Analog Data (DSAD). DSAD comprises for the most part of sound, feature, illustrations, or picture documents. In a sound document, for instance, the high and low frequencies, which the human ear can't listen, may be truncated from the record.

1.2.2 Lossless data compression

Lossless data compression is a procedure that permits the utilization of data compression calculations to pack the content data furthermore permits the precise unique data to be remade from the compacted data. This is in as opposed to the lossy data compression in which the careful unique data can't be recreated from the compacted data. The prevalent ZIP record organizes that is being utilized for the compression of data documents is likewise a use of lossless data compression approach. Lossless compression is utilized when it is vital that the first data and the decompressed data be indistinguishable. Lossless content data compression calculations typically abuse factual excess in such a path in order to speak to the sender's data all the more briefly with no blunder or any kind of loss of vital data contained inside of the content information data. Since the majority of this present reality data has factual excess, thusly lossless data compression is conceivable. Case in point, In English content, the letter "an" is a great deal more basic than the letter 'z', and the likelihood that the letter "t" will be trailed by the letter "z" is little. So this sort of repetition can be evacuated utilizing lossless compression. Lossless compression techniques may be classified by kind of data they are intended to pack. Compression calculations are essentially utilized for the compression of content, pictures and sound. Most lossless compression projects utilize two various types of calculations: one which creates a factual model for the info data and another which maps the information data to bit strings utilizing this model as a part of such a route, to the point that as often as possible experienced data will deliver shorter yield than improbable(less continuous) data.

2. A Review of Various Data Compression Techniques

R.S. Brar and B.Singh, "A survey on different compression techniques and bit reduction algorithm for compression of text data" : This paper provides a survey of different basic lossless and lossy data compression techniques. On the basis of these techniques a bit reduction algorithm for compression of text data has been proposed by the authors based on number theory system and file differential technique which is a simple compression and decompression technique free from time complexity. Future work can be done on coding of special characters which are not specified on key-board to revise better results [1].

S. Porwal, Y. Chaudhary, J. Joshi, M. Jain, "Data Compression Methodologies for Lossless Data and Comparison between Algorithms" : This research paper provides lossless data compression methodologies and compares their performance. Huffman and arithmetic coding are compared according to their performances. In this paper the author has found that arithmetic encoding methodology is powerful as compared to Huffman encoding methodology. By comparing the two techniques the author has concluded that the compression ratio of arithmetic encoding is better and furthermore arithmetic encoding reduces channel bandwidth and transmission time also [2].

S. Shanmugasundaram and R. Lourdusamy, "A Comparative Study of Text Compression Algorithms" : There are lot of data compression algorithms which are available to compress files of different formats. This paper provides a survey of different basic lossless data compression algorithms. Experimental results and comparisons of the lossless compression algorithms using Statistical compression techniques and Dictionary based compression techniques were performed on text data. Among the statistical coding techniques the algorithms such as Shannon-Fano Coding, Huffman coding, Adaptive Huffman coding, Run Length Encoding and Arithmetic coding are considered. A set of interesting conclusions are derived on their basis. Lossy algorithms achieve better compression effectiveness than lossless algorithms, but lossy compression is limited to audio, images, and video, where some loss is acceptable. The question of the better technique of the two, "lossless" or "lossy" is pointless as each has its own uses with lossless techniques better in some cases and lossy technique better in others [3].

S. Kapoor and A. Chopra, "A Review of Lempel Ziv Compression Techniques" : This Paper direct several key issues to the dictionary based LZW algorithm existing today. In contrast to the Previous LZW ,we would like to improve LZW algorithm in

future which definitely get good results like: Better compression ratio , time taken for searching in the dictionary for pattern matching in encoding and decoding got reduced and dictionary size become Dynamic. In the future the authors would like to present modifications possible in existing Lempel Ziv Welch Algorithm by changing its Data structure. The authors conclude that the Lempel Ziv Algorithm is quite old but still is the standard algorithm for various proposes ranging from text compression to image and video compression. Authors also discuss that the LZ algorithm has various issues regarding its performance and internal structure, which various other authors describes in the review tried to improve. LZ algorithm is a Dictionary Based algorithm, but as the dictionary structure gets flatten out after compression process. The authors conclude by saying that they would like to implement an LZW variant using Hash Sets and find out the performance with respect to LZ family [4].

I. M.A. D. Suarjaya, "A New Algorithm for Data Compression Optimization" : In this paper the author propose a new algorithm for data compression, called j-bit encoding (JBE). This algorithm will manipulates each bit of data inside file to minimize the size without losing any data after decoding which is classified to lossless compression. This basic algorithm is intended to be combining with other data compression algorithms to optimize the compression ratio. The performance of this algorithm is measured by comparing combination of different data compression algorithms. This paper proposes and confirms a data compression algorithm that can be used to optimize other algorithm. An experiment by using 5 types of files with 50 different sizes for each type was conducted, 5 combination algorithms has been tested and compared. This algorithm gives better compression ratio when inserted between move to front transform (MTF) and arithmetic coding (ARI). Because some files consist of hybrid contents (text, audio, video, binary in one file just like document file), the ability to recognize contents regardless the file type, split it then compresses it separately with appropriate algorithm to the contents is potential for further research in the future to achieve better compression ratio [5].

S.R. Kodituwakku and U.S. Amarasinghe, "Comparison Of Lossless Data Compression Algorithms For Text Data" : In this paper a set of selected algorithms are examined and implemented to evaluate the performance in compressing text data. An experimental comparison of a number of different lossless data compression algorithms is presented in this paper. The article is concluded by considering the Shannon Fano algorithm as the most efficient algorithm among the selected ones [6].

R. Kaur and M. Goyal "An Algorithm For Lossless Text Data Compression": In this paper the authors present a technique to reduce the number of bits required to represent a character by using 6-bit binary coding instead of a 8-bit binary coding technique. The important feature of this algorithm is its simplicity. A different technique is developed to reduce the size of text files which is based on number theory and bit reduction. The authors conclude that the most important feature of this algorithm is its simplicity. An entirely different technique is developed to decrease the size of text files. The technique of saving space have shown in this algorithm. Since every character is taken care of, so the output codes do not depend upon the redundancy, like the traditional compression algorithms. After the code formulation, ASCII code modifies the binary numbers, which finally reduce the file size. A lot of research and findings led to the conclusion that there are no such algorithms in data compression that emphasis on different compression based on number theory and bit reduction [7].

H. Altarawneh and M. Altarawneh, "Data Compression Techniques on Text Files: A Comparison Study" : This paper presents various methods of data compression such as LZW, Huffman, Fixed-length code (FLC) and Huffman after using Fixed-length code (HFLC) on text files. The authors have evaluated and test the algorithms on various sizes of text files and compared their performance on various parameters such as compression size, compression ratio, compression time and entropy [8].

U. Khurana and A.Koul, "Text Compression And Superfast Searching": A new compression technique that uses referencing through two-byte numbers (indices) for the purpose of encoding has been presented. The technique is efficient in providing high compression ratios and faster search through the text. It leaves a good scope for further research for actually incorporating phase 3 of the given algorithm. The same should need extensive study of general sentence formats and scope for maximum compression. Another area of research would be to modify the compression scheme so that searching is even faster. Incorporating indexing so as to achieve the same is yet another challenge [9].

A. Singh and Y. Bhatnagar, "Enhancement of data compression using Incremental Encoding" : This paper describes the two phase encoding technique which compresses the sorted data more

efficiently. This research paper provides a way to enhance the compression technique by merging RLE compression algorithm and incremental compression algorithm. In first phase the data is compressed by applying RLE algorithm that compresses the frequent occur data bits by short bits. In the second phase incremental compression algorithm stores the prefix of previous symbol from the current symbol and replaces with integer value. This technique can reduce the size of sorted data by 50% using two phase encoding technique [10].

A.J Mann, "Analysis and Comparison of Algorithms for Lossless Data Compression" : In this paper the author discusses and compares selected set of lossless data compression algorithms such as RLE, Huffman and Arithmetic coding. The author compares the performance of these algorithms on the basis of various parameters such as Compression Ratio, Compression speed, Decompression speed, Memory space etc. The author has concluded that the compression speed of Huffman is better than the Arithmetic coding, but the compression ratio of Arithmetic coding is better as compared to the Huffman coding. The author has also concluded that Arithmetic coding is the efficient compression algorithm among the selected ones [11].

K. Rastogi and K. Sengar, "Analysis and Performance Comparison of Lossless Compression Techniques for Text Data" : In this paper, the authors discuss about the lossless text data compression algorithms such as Run Length Encoding, Huffman Coding and Shanon Fano coding. The authors have concluded the article by doing a comparison of these techniques. The authors have also concluded that Huffman technique is most optimal for lossless data compression [12].

M. Sharma, "Compression Using Huffman Coding" : In this paper, the author has analyzed Huffman algorithm and compare it with other common compression techniques like Arithmetic, LZW and Run Length Encoding. The author has concluded that arithmetic coding is very efficient for bits and reduces the file size dramatically.RLE is simple to implement and fast o execute. LZW algorithm is better to use for TIFF, GIF and Textual Files [13].

S. Shanmugasundaram and R. Lourdusamy, "IIDBE: A Lossless Text Transform for Better Compression" : In this paper, the authors propose an approach to develop a Dictionary based reversible lossless text transformation called Improved Intelligent Dictionary Based Encoding (IIDBE). This approach is used in conjunction with BWT that can be applied to source text to improve the existing or backend algorithm's ability to compress and also offer a sufficient level of security of the transmitted information. The experimental results of this compression method are also analyzed. IIDBE gives 18.32% improvement over Simple BWT, 8.55% improvement over BWT with 2.28% improvement over BWT with IDBE and about 1% over BWT with EIDBE [14].

P. Kumar A.K. Varshney, "Double Huffman Coding" : In this paper, the authors present a novel technique that work on Huffman Coding and after getting codeword for the Symbol the authors compress it on the Basis of its Binary no. 0 and 1. The authors have analyzed the results by comparing Double Huffman coding with Huffman coding and concluded that the Double Huffman coding despite being costly provides better results in terms of performance and space savings [15].

R.Gupta1, A. Gupta, S. Agarwal, "A Novel Data Compression Algorithm For Dynamic Data" : A compression algorithm for dynamic data is presented in this paper. The size of data keeps on increasing rapidly. It is a memory efficient data compression technique comprising of a block approach that keeps the data in compressed form as long as possible and it also enables the data to be appended to the already compressed text. The algorithm requires only a minimal decompression for supporting update of data using a little preprocessing which reduces the unnecessary time spent in compression-decompression. The text document can be modified further as required without decompressing and again compressing the whole document. The paper also presents the design of the required data structures for the algorithm proposed by the author and also the performance results of the proposed system [16].

A. Kattan, "Universal Intelligent Data Compression Systems: A Review" (IEEE) Researchers have addressed the problem of universal compression using two approaches. The first approach has been to develop adaptive compression algorithms, where the system changes its behavior during the compression to fit the encoding situation of the given data. The second approach has been to use the composition of multiple compression algorithms. A third approach has also been adopted by researchers in order to develop compression systems: the application of computational intelligence paradigms. This has shown remarkable results in the data compression domain improving the decision making process and outperforming conventional systems of data compression. This paper reviews some of the

previous attempts to address the universal compression problem within conventional and computational intelligence techniques [17].

M.H Btoush, J. Siddiqi, B. Akhgar, "Observations on Compressing Text Files of Varying Length" (IEEE): The paper compares different data compression algorithms of text files: LZW, Huffman, Fixed-length code (FLC), and Huffman after using Fixed-length code (HFLC). Authors compare these algorithms on different text files of different sizes in terms of compression scales of: Size, Ratio, Time (Speed), and Entropy. Their evaluation reveals that initially for smaller size files the simplest algorithm LZW performs worst for first two scales than the more complex Huffman algorithm but as the size of the text increases ,the position is reversed. LZW performs better than Huffman for the scales time and entropy but for larger files once again the position is reversed [18].

3. CONCLUSION

In this paper, various data compression techniques has been reviewed. For future work, we will develop a text data compression technique which will provide better efficiency and effective compression ratio, which may be the combination of one or two techniques.

4. REFERENCES

[1] R.S. Brar and B.Singh, "A survey on different compression techniques and bit reduction algorithm for compression of text data" International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Volume 3, Issue 3, March 2013

[2] S. Porwal, Y. Chaudhary, J. Joshi and M. Jain , " Data Compression Methodologies for Lossless Data and Comparison between Algorithms" International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2, Issue 2, March 2013

[3] S. Shanmugasundaram and R. Lourdusamy, "A Comparative Study of Text Compression Algorithms" International Journal of Wisdom Based Computing, Vol. 1 (3), December 2011

[4] S. Kapoor and A. Chopra, "A Review of Lempel Ziv Compression Techniques" IJCST Vol. 4, Issue 2, April - June 2013

[5] I. M.A.D. Suarjaya, "A New Algorithm for Data Compression Optimization", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 8, 2012, pp.14-17

[6] S.R. Kodituwakku and U. S. Amarasinghe , "Comparison Of Lossless Data Compression Algorithms For Text Data" Indian Journal of Computer Science and Engineering Vol1No 4 416-425

[7] R. Kaur and M. Goyal, "An Algorithm for Lossless Text Data Compression" International Journal of Engineering Research & Technology (IJERT), Vol. 2 Issue 7, July - 2013

[8] H. Altarawneh and M. Altarawneh, " Data Compression Techniques on Text Files: A Comparison Study " International Journal of Computer Applications, Volume 26– No.5,July 2011

[9] U. Khurana and A. Koul, "Text Compression And Superfast Searching" Thapar Institute Of Engineering and Technology, Patiala, Punjab, India-147004

[10] A. Singh and Y. Bhatnagar, " Enhancement of data compression using Incremental Encoding" International Journal of Scientific & Engineering Research, Volume 3, Issue 5, May-2012

[11] A.J Mann, " Analysis and Comparison of Algorithms for Lossless Data Compression"International Journal of Information and Computation Technology, ISSN 0974-2239 Volume 3, Number 3 (2013), pp. 139-146

[12] K. Rastogi, K. Sengar, "Analysis and Performance Comparison of Lossless Compression Techniques for Text Data" International Journal of Engineering Technology and Computer Research (IJETCR) 2 (1) 2014, 16-19

[13] M. Sharma, "Compression using Huffman Coding " IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.5, May 2010

[14] S. Shanmugasundaram and R. Lourdusamy, " IIDBE: A Lossless Text Transform for Better Compression " International Journal of Wisdom Based Computing, Vol. 1 (2), August 2011

[15] P. Kumar and A.K Varshney, " Double Huffman Coding " International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) Volume 2, Issue 8, August 2012

[16] R. Gupta, A. Gupta, S. Agarwal, "A Novel Data Compression Algorithm For Dynamic Data" IEEE REGION 8 SIBIRCON

[17] A. Kattan, "Universal Intelligent Data Compression Systems: A Review" 2010 IEEE[18] M. H Btoush, J. Siddiqi and B. Akhgar, "Observations on Compressing Text Files of Varying Length " Fifth International Conference on Information Technology: New Generations, 2008 IEEE 2012, pp.1-6

[18] M. H Btoush, J. Siddiqi and B. Akhgar, "Observations on Compressing Text Files of Varying Length " Fifth International Conference on Information Technology: New Generations, 2008 IEEE