GLIMPSE - Journal of Computer Science • Vol. 2(2), JULY-DECEMBER 2023, pp.31-35

A SYSTEMATIC REVIEW OF WAVELET TREE COMPRESSION TECHNIQUES

[1]Pradeep Gupta, [2] Dr Sonam Gupta
[1]Assistant Professor, Ajay Kumar Garg Engineering College, Ghaziabad, UP, India
[2]Assiociate Professor, Ajay Kumar Garg Engineering College, Ghaziabad, UP, India
[1]guptapradeep@akgec.ac.in, [2]guptasonam@akgec.ac.in

Abstract: **The wavelet tree is a proficien tdata structure utilized for compressing and indexing files, which in turn improves information retrieval for specific data. This research work explores the numerous applications ofcompressionalgorithmsacrossvariousfields.Additionally,theresearchalsoaddressesseveralresearchquestionsrelated to the use of compression algorithms. The wavelet tree is a powerful tool that can help compress largedatasets and facilitate faster information retrieval. Through our research, we have uncovered numerous practicalapplications of compression algorithms, such as in data mining, bioinformatics, and natural language processing.We have also discussed various research questions, such as how compression algorithms can be optimized forvarious data sets and their potential enhancements through utilization the accuracy of machine learning models. Overall, this research work highlights the significant benefits of using compression algorithms, particularly thewavelet tree, in a wide range of applications. It provides a deeper understanding of the potential of these algorithmsand the importantrole theycanplayinimprovingdata processingandanalysis.**

Keywords- **Wavelet trees, Compression Algorithm, LZW Alo, WBTC Coding**

I. INTRODUCTION

The exponential expansion of digital content has led to the need for efficient text summarization techniques thatcan quickly and accurately provide relevant information to users. Text summarization involves condensing largeamounts of information into a concise summary, without compromising the accuracy or meaning of the originaltext. One approach to text summarization involves the use of wavelet trees, which are data structures that canefficientlystore andretrieve data.

Wavelet trees are extensively employed in diverse applications such as bioinformatics, data compression, and textretrieval. In text summarization, wavelet trees are used to store and handle substantial volumes of text data togenerate concise summaries. The construction of wavelet trees for text summarization has been an area ofinvestigation for several years, and researchers have developed various algorithms for constructing wavelet treesusingdifferentcompressiontechniques.

One popular compression technique used in the construction of wavelet trees for text summarization is the Lempel-Ziv-Welch (LZW) compression algorithm. LZW compressionis a algorithm for lossless data compression that employs a dictionary-sed approach to encode repeated patterns in the text. This algorithm has been shown to be effective in constructingwavelettrees for text summarization, a sitcan efficientlyencode large volumes of text while preserving their structure and meaning.

Wavelettrees constructed using LZW compression have been used for information retrieval in various applications, including web search engines and data mining. In these applications, wavelet trees are used toefficiently store and retrieve large volumes of data, making them ideal for use in text summarization. By usingwavelet trees and LZW compression for text summarization, it is possible to generate high-quality summaries quickly and accurately, while reducing the computational cost of processing large amounts of data.

The use of wavelet trees for text summarization has several advantages over other summarization techniques.Wavelettrees can efficiently store and retrieve large amounts of data, making the mideal for processing text data. Additionally, wavelet trees can be used to generate summaries that preserve the structure and meaning of theoriginal text, ensuring that the summary accurately represents the content of the original document. Overall, the use of wavelettrees and LZWcompression for text summarization exhibits significant potential for enhancing the efficiency and accuracy of text summarizationin various applications.

Wavelet trees have proven to be effective data structures for storing and processing large volumes of text data.They have been used in various applications, including information retrieval and text summarization. However, the accuracy and efficiency of wavelet tree-based systems can be improved by incorporating machine learningand deep learning techniques. Overall, implementing a wavelettree using ML&DLfor in formation retrieval and text summarization requires a combination of data preprocessing, algorithm development, and testing and evaluation. With meticulous strategizing and execution, the reexists the potential to develop are markably precise andefficient system that can handle large volumes of text data.

Research Question

Can we further improve the efficiency of wavelettree-basedalgorithms for range query operations, such as range counting or range reporting, in the presence of noisyorun certain data? How can wavelet trees be adapted to handle streaming or dynamic data, where new elements are continuously add edor-removed from the sequence or collection being represented? How can wavelet trees be combined with other data structures, such as Bloom filters or hash tables, to achieve faster and more efficient query processing in various applications,such as bioinformatics or text retrieval?

This paper is structured into the subsequent sections: Section 2 outlines the wavelet tree's application, section 3discussed about the literature study, while Section 4 goes into research topics that arose throughout the review. Finally, wesummarise our finding sand provide references at the end of our research (Section5).

1. Application of Wavelet Tree

Wavelet trees are a versatile data structure that can find utility in diverse tasks, encompassing text indexing among others, computational geometry, and data compression.They are particularly well-suited for problems where it is necessary to answer range queries efficiently. A wavelet tree is a compact representation of asorted sequence of elements. It is constructed by recursively partitioning the sequence into two halves, based on the value of the first element. The left portion comprises all elements that are less than or equal to the initial element, while the right portion consists of all elements that exceed the initial element.This process is repeated for each half, until all elements have been partitioned. The wavelet tree can be employed for addressing range queries efficiently by following the route spanning from the tree's root to the leaf node that contains the desire drange. For example, to find all elements in arange [L,R] of asorted sequence, we would commence from the tree's root and proceed by traversing the left branch until we reach a leaf that contains an element less thanor equal to L. We would then follow the right branch untilwe reach a leaf that contains an element greater than R. All of theelements in the leaves that we visit will be in the range [L,R].

II. LITERATURE REVIEW

In this paper[1], the authors propose a technique for using LZW compression on wavelet trees to efficiently store and retrieve textual, visual, auditory, and multimedia data. The wavelet tree is a data structure that has been extensively utilized in full-text categorizing and compression due to its ability to provide efficient time and space complexity. The literature review suggests that wavelet trees have been extensively used in various fields, suchas cyberspace, healthcare sector, agricultural practices, computational biology, and seismic event detection. The authors provide an overview of the fundamental concepts of wavelets, including the cataloguing process, practical metrics, wavelet packets, wavelet information, wavelet array, and intricacy of wavelets. The literature review highlights the importance of wavelet trees in modern full-text indexing, where they have been successfully usedin indexing large amounts of text data with high efficiency. The literature review further investigates the open challenges where wavelet trees can be used to establish indexing for various databases.

The authors of this study [2] offer a system for effectively storing, classifying, and indexing huge datasets that blends ML methods with wavelet trees. Wavelet trees are small data structures that have been widely employed in computational geometry for data management, compression, and indexing. According to the literature study, machine learning algorithms are rapidly being employed for data categorization and analysis. The authors address the use of wave lettrees, wave letentropy, wave letmatrix, and wavelet packets in machinelearning. The literature review emphasises the benefits of employing machine learning algorithms with wavelet trees for data storage and categorization. The suggested solution incorporates three techniques: LZW compression, a wavelet tree, and the support vector machine (SVM) algorithm. The authors show that their suggested technique can compress and categorise datasets efficiently.

In their paper [3], the authors present an ovel approachfor constructing a wavelet tree dedicated to processing text files and employing the Lempel-Ziv-Welch (LZW) compression-method for compression purposes. They highlight the increasingly challenging task of retrieving information from larger-epositories containingun structured or semi-structured data due to the exponential growth of data volume. Consequently, the necessity arises for the development of techniques capable of efficiently storing data in reduced space and optimizing-memory requirements.

The authors propose [4] an efficient algorithm for compressed pattern matching (CPM) that matches patterns in a compressed text without decompressing it.The proposed algorithm, WBTC_WT, use satagged sub-optimal code category called word-based tagged code (WBTC) and a self-indexed data structure called wavelet tree. WBTC isused to encode the text, while the wavelet tree provides fast searching over the compressed text. The authors demonstrate that WBTC_WT can match arbitrary portions of text without decompressing it, and it eliminates theproblem of false matches encountered in some previous approaches.

In the paper [5], authors propose two algorithms for compressing textual indexes to improve the storage and retrieval of information from the internet. Previous research has focused on optimizing indexes for reduced storage space, construction time, and retrieval time. The two proposed algorithms use different combinations of the traditional wavelet tree and Lempel-Ziv-Welch(LZW) compression techniques. The first algorithm, TWL, applies LZW compression to the traditional wavelet tree to construct the index. The second algorithm, PWTL, proposes LZW compression with parallel wavelet tree to construct the textual index. The authors evaluate both algorithm son I3, I5, and I7 process or swith multiplesizes of index. There sults show that the proposed algorithms outperform traditional wavelet tree in terms of index construction time. The use of succinct data structures for indexing textual data has become increasingly popular in recent times. The proposed algorithms in this paper offer an efficient solution for compressing textual indexes, thereby reducing index construction time while maintaining search efficiency. This research is significant because it provides a practical approach to handling large volumes of information on the internet, which is an essential requirement for modern-day businesses and organizations.

The authors of paper [6] propose a new technique for efficient indexing of the rapidly increasing text corpus onthe World Wide Web. Their proposed algorithm, WBTC_PWT, uses the Word-Based Tagging Coding (WBTC) compression technique and a parallel wavelet tree to construct the index. The technique is designed to reduce both space and time complexity during the search process. The authors demonstrate that the proposed algorithm is significantly faster than other existing search algorithms based on compressed text and reduces the chances offalse matching results. The proposed algorithm utilizes the features of compressed pattern matching to minimize search time complexity. The unique words present in the text corpus are divided in to different levels based on the word frequency table and a different wavelet tree is constructed for each level in parallel. The use of WBTC encoding technique reduces the space complexity of the index, while the parallel wavelet tree construction reduces the time complexity. The authors evaluate the performance of the proposed algorithm on a large data set and compare it with other existing search algorithms based on compressed text. The results show that the proposed WBTC_PWT algorithm out performs the existing algorithms in terms of speed and accuracy.

In this paper [7], the authors proposed a method for text compression and matching using compressed patternmatching (CPM). The proposed method utilizes word-based tagged coding for compression and Wavelet Trees for efficient representation of the compressed text. The method is fast, reduces storage space, and minimizes thetime required for text matching without compromising the accuracy of the results. The authors evaluated theproposed approach and found it to be better than other existing methods that support CPM in terms of compression ratio and accuracy of matching results. Overall, this method has the ability to boost efficiency of text matching and retrieval inlarge-scale data sets.

This research paper [8] proposes a new indexing techniqueusing a parallel wavelet tree algorithm with hybridization of the Map-Reduce concept for efficient textual search. The proposed algorithm significantly lowers the time required to build an index, especially for large data sets, by utilizing multiplethreadsworking in parallel. Results of experiments on various core processors show consistent performance, except for 16-core processorswhen the dataset is not sufficiently large. With existing indexing approaches, the proposed algorithm provides a reasonable compromise and shows promising results.

In the research paper [9] the author proposes a blind watermarking method that uses particle swarm optimization (PSO) on discrete wavelet transform (DWT). The method involves embedding a watermark in a digital image by quantizing adjacent wavelet coefficients on wavelet trees.

In [10], the authors propose news equential and parallel algorithms for constructing wavelet trees. The algorithms are based on a new bottom-up technique that computes the leaves of the tree first and then propagates the information upwards to the root. The authors describe new sequential algorithms for both RAM and external memory, and they adapt these algorithms to parallel computerswith shared and distributed memory. The algorithms out perform previous ones in bothtime and memory efficiency, as they only require auxiliary in formation that can be computed from the leaves. Most of the algorithms are also adapted to the wavelet matrix,which makes the msuitable for large alphabets. Overall, this paper presents a promising approach for constructing wavelet trees efficiently.

The authors of this study [11] propose that wavelet trees be used to solve fundamental algorithmic issues such asrange-quantile searches, range next value queries, and range intersection queries. They demonstrate wavelet trees' adaptability in encoding sequences, permutations, text collections, binary relations, and other concise datastructures. The authors investigate several uses of these queries in information retrieval, including documentretrieval for hierarchical and temporal documents, as well as the representation of inverted lists. This work sheds light on the use of wavelet trees in tackling algorithmic challenges in information retrieval and other disciplines.

The authors give a comparative assessment of the practical performance of the wavelet tree data structure in thispaper [12]. Wavelet trees, which can accommodate multi character alphabets, are extensively employed in full-text indexing and data reduction for rank and select queries. The work covers both theoretical and practical data, demonstrating that run-length coding size approaches the original string's 0-order empirical entropy size. They also present a complete generic package of wavelet trees that can handle various coding methods and tree forms.The experimental investigation sheds light on the practical performance of various wavelet tree modifications. Overall, this study is useful for scholars and practitioners who want to use wavelet trees for effective datacompression and searching.

The authors of this study [13] present concurrent light weight methods for generating wavelet trees, rank and choose structures, and suffix arrays in shared memory. They introduce two parallel wavelet tree methods that use less memory and perform better than existing parallel techniques. In addition, the authors create the first parallel suffix array technique based on induced copying, which is more efficient in practise than previous parallel implementations. They also assess the parallel creation of rank and choose structures and demonstrate how the algorithms may be used to create the FM-index in parallel.

III. RELATED DISCUSSION

RQ 1: Can we further improve the efficiency of wavelet tree-based algorithms for range query operations, such as range counting orrange reporting, in the presence of noisyoruncertain data?

Ans: There is ongoing research in exploring various techniques to improve the efficiency of wavelet tree-basedalgorithms for range query operations, especially in the presenceof noisy or uncertain data. Some approaches include incorporating probabilistic models or machine learning techniques into the algorithm design. However, more study is required to thoroughly investigate the possibilities of these technologies and assess their efficacy inpractical scenarios.

RQ 2: How can wavelet trees be adapted to handle streaming or dynamic data, where new elements are continuously added or removed from the sequence or collection being represented?

Ans: Several methods exist for modifying wavelet trees to accommodate streaming or dynamic data. A dynamic wavelet tree, which can effectively manage insertions and deletions of elements in the underlying sequence, is one solution. Another method is to employ a wavelet matrix, which allows for fast up dates to individual elements in the data matrix format. Further more, approaches like as sluggish propagation and batched updates may be utilised to efficiently manage wavelet tree changes in a streaming or dynamic environment.

RQ 3: How can wavelet trees be combined with other data structures, such as Bloom filters or hash tables, to achieve faster and more efficient query processing in various applications, such as bioinformatics or text retrieval?

Ans: One way to combine wavelet trees with other data structures is to use Bloom filters or hash tables to reduce the number of elements to be queried in the wavelet tree. This can speed up query processing and reduce memory usage. For example, in bioinformatics, Bloom filters can be used to store a compressed version of a large DNA sequence, and wavelet trees can be used to index and search for specific patterns within the sequence. In text retrieval, a hash table can be used to store the inverted index of a large text collection, and a wavelet tree can be used to efficiently process range queriesover the index.

IV. CONCLUSION

The use of wavelet trees has been extensively explored in various domains due to their efficiency and versatility. In this research paper, we have discussed the use of wavelet trees incompression techniques, specifically with the LZW compression algorithm. This technique has been applied to compress textfiles, image files, audio, and video files, there by significantly reducing the storage and retrieval time for users. Moreover, the compression capabilities of wavelet trees have been combined with other data structures like hash tables and Bloom filters toenhance query processing in various applications, such as bioinformatics and text retrieval. The wavelet tree-based data structures have been found to achieve faster and more efficient query processing, making them apopular choice in these domains. Additionally, wavelet trees have been adapted to handle streaming or dynamicdata, where new elements are continuously added or removed from the sequence or collection being represented. Further research on combining wavelet trees with other data structures can open up new avenues for improving the efficiency and performance of query processing in various applications.

REFERENCES

[1] S. Gupta, N. Katiyar, A. K. Yadav, and D. Yadav, “Index Optimization Using Wavelet Tree and Compression,” in *Proceedings of Data Analytics and Management*, D. Gupta, Z. Polkowski, A. Khanna,

[2] S. Bhattacharyya, and O. Castillo, Eds., in Lecture Notes on Data Engineering and Communications Technologies. Singapore: Springer Nature, 2022, pp.809–821.doi:10.1007/978-981-16-6289-8_66.

[3] N. Katiyar, S. Gupta, A. K. Yadav, and D. Yadav, “Wavelet Tree ensembles with Machine Learning andits classification,” *J. Phys. Conf. Ser.*, vol. 1998, no. 1, p. 012001, Aug. 2021, doi: 10.1088/1742-6596/1998/1/012001.

[4] S. Gupta, A.K. Yadav, D.Yadav, and B.Shukla, “Ascalableapproach for index compressionusing wavelet tree and LZW,” *Int. J. Inf. Technol.*, vol. 14, no. 4, pp. 2191–2204, Jun. 2022, doi:10.1007/s41870-022-00915-y.

[5] S.P. Mishra, R. Prasad, and G. Singh,“Fast Pattern Matching inCompressed Textusing Wavelet Tree,” *IETEJ. Res.*, vol. 64, no.1, pp.87–99, Jan.2018, doi:10.1080/03772063.2017.1347071.

[6] A. K. Yadav, S. Gupta, D. Yadav, and B. Shukla, “Optimization of Textual Index Construction Using Compressed Parallel Wavelet Tree,” in *Proceedings of International Conference on Computing andCommunication Networks*, A. K. Bashir, G.

Fortino, A. Khanna, and D. Gupta, Eds., in Lecture Notes in-Networks and Systems. Singapore: Springer Nature, 2022, pp. 457–466. doi: 10.1007/978-981-19-0604-6_43.

[7] S. Srivastav, P. K. Singh, and D. Yadav, "A Method to Improve Exact Matching Results in CompressedText using Parallel Wavelet Tree," *Scalable Comput. Pract. Exp.*, vol. 22, no. 4, Art. no. 4, Nov. 2021,doi:10.12694/scpe.v22i4.1870.

[8] S. Srivastav and P. K. Singh, "An approach for fast compressed text matching and to avoid false matchingusing WBTC and wavelet tree," *EAI Endorsed Trans. Scalable Inf. Syst.*, vol. 8, no. 30, pp. e6–e6, 2021,doi:10.4108/eai.23-10-2020.166717.

[9] A. K. Yadav, D. Yadav, A. Verma, Mohd. Akbar, and K. Tewari, "Scalable thread based index construction using wavelet tree," *Multimed. Tools Appl.*, vol. 82, no. 9, pp. 14037–14053, Apr. 2023, doi:10.1007/s11042-022-13906-9.

[10] S. Dubnov, Z. Bar-Joseph, R. El-Yaniv, D. Lischinski, and M. Werman, "Synthesizing sound texturesthrough wavelet tree learning," *IEEE Comput. Graph. Appl.*, vol. 22, no. 4, pp. 38–48, Jul. 2002, doi:10.1109/MCG.2002.1016697.

[11] P. Dinklage, J. Ellert, J. Fischer, F. Kurpicz, and M. Löbel, "Practical Wavelet Tree Construction," *ACMJ. Exp. Algorithmics*, vol.26, p. 1.8:1-1.8:67, Jul.2021, doi:10.1145/3457197.

[12] T.Gagie, G.Navarro, and S.J.Puglisi, "Newalgorithmson-wavelet trees and applications to information retrieval," *The or.Comput. Sci.*, vol. 426–427, pp.25–41, Apr. 2012, doi:10.1016/j.tcs.2011.12.002.

[13] T. Gagie, S. J. Puglisi, and A. Turpin, "Range Quantile Queries: Another Virtue of Wavelet Trees," in *String Processing and Information Retrieval*, J. Karlgren, J. Tarhio, and H. Hyyrö, Eds., in Lecture Notes in Computer Science. Berlin, He idelberg: Springer, 2009, pp. 1–6.doi:10.1007/978-3-642-03784-9_1.

[14] R. Grossi, J. S. Vitter, and B. Xu, "Wavelet Trees: From Theory to Practice," in *2011 First International Conference on Data Compression, Communications and Processing*, Jun. 2011, pp. 210–221. doi:10.1109/CCP.2011.16.

ABOUT THE AUTHOR

Pradeep Gupta received his B.E. (CSE) in 2006, MTech (CSE) in 2011.He has 15 years of experience in teaching.He is currently employed as an Assistant Professor at Ghaziabad's Ajay Kumar Garg Engineering College. Artificial Intelligence, Machine Learning, Deep Learning, and Cyber Security are some of his research interests.

Dr. Sonam Gupta is B.E., M.Tech, and Ph.D. She has over 14 years of experience in teaching. Currently, she is working as an Associate Professor (CSE) in Ajay Kumar Garg Engineering College, Ghaziabad. She has published over 20 papers in various national/ international journals indexed in SCIE/Scopus/ESCI. She is a reviewer of many international journals/ conferences. Her research interest includes software evolution, machine learning, and data analytics. She is guiding various undergraduate and postgraduates projects in the field of machine learning and software maintenance.