International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-6, March 2020

Lossless Tamil Compression using ASCII Substitution and Modified Huffman Encoding Technique

B. Vijayalakshmi, N. Sasirekha

Abstract: ***Tamil language is a longest existing classical language in the humankind. It is one of the scheduled languages in India and also official language for many countries. Communication using Tamil language is drastically growing after the practice of internet. Storage of Tamil documents also emerged greater than before. So there is a high requirement for data compression to improve the efficiency of storage and fast communication of Tamil documents. This research paper provides a novel approach for Lossless compression technique especially for Tamil documents. The compression process involves three major steps: separation of English alphabets appears with in Tamil text, substitution of ASCII in the place of Unicode Tamil characters using static dictionary and building a Huffman tree with a variation method for encoding the Tamil document. Performance of Tamil compression is measured by finding the space efficiency of memory storage needed to store the compressed file. The space efficiency can be measured by finding the parameters of compression ratio, compression factor and percentage of compression. Time efficiency is calculated by finding the time taken by the algorithm to compress and decompress a file. The average compression achieved through this compression technique is 72.08%. The decompression process restores the original file without any loss of data.***

Keywords- Text compression, dictionary, Unicode, ASCII and Huffman encoding.

I. INTRODUCTION

A. Lossless Text Compression for Unicode Tamil Documents

Compression method is used usually to minimize the usage of resources like data, space and to enhance the transmission speed. The compression techniques are generally classified into two major categories. They are lossless and lossy compression techniques. Lossless algorithms conserve the same information again so that the original data restores without any loss of data. Popular lossless compression techniques are Huffman encoding, run length encoding, Lempel-ZivWelch (LZW) Coding. The lossy algorithms do not preserve the original data so they are called irreversible. Lossy algorithms make use of both data redundancy and perception properties of human. As a result of eliminating a part of information, higher compression rates can be achieved using lossy compression techniques. Traditional lossy techniques are used for compressing digital image and video information. Text compression involves in minimizing the amount of bits required to indicate the data. In this paper the lossless text compression technique for Tamil documents was presented. Compressed data can accumulate less storage capacity, expand the velocity of communication and diminish the cost for storage hardware and network bandwidth.

B. Dictionary Based Compression

The most prominent compression technique is dictionary based compression. The dictionary contains a record of strings of possible symbols stored in a table like arrangement. It uses the index of entries to represent larger and repeated dictionary word or character replaced by a smaller one. The dictionary compression can be a static or dynamic scheme type. In this paper, the compression technique is based on a static dictionary in the earlier stage of compression process which is easy and a permanent one. This static dictionary contains the subset of all the common prototype of Unicode Tamil characters indexed by ASCII characters. The size of Unicode character ranges from 1 byte to 4 bytes depending upon the document storage encoding style [6].

C. Tamil Language Alphabets

Tamil is an abugida language. An abugida is a kind of syllabify in which the vowel is changed by modifying the base consonant symbol, so that all the forms that match to a given consonant plus each vowel be similar to one another. Amharic, Hindi and Burmese are also abugida languages. There are 12 vowels, 18 consonants and 1 aytam alphabet character (neither vowel nor consonant) in Tamil language script. Apart from that a set of 216 combining letters produced by accumulating vowel marker to the consonant. Totally there are 247 characters available in basic Tamil script [14]. Some vowels require the basic shape of the consonant to be modified in such a way that it is specific to that vowel. Others are written by adding vowel specific suffix to consonant, specific prefix to consonant and both suffix & prefix to a consonant. The Unicode Tamil characters are used for document storage [23].

Manuscript received on February 10, 2020.
Revised Manuscript received on February 20, 2020.
Manuscript published on March 30, 2020.

* Correspondence Author
B.Vijayalakshmi*, Ph.D. Research Scholar, Department of Computer Science, Vidyasagar College of Arts and Science, Udumalpet, Tamilnadu, India. Email: vijib79@yahoo.com
Dr.N.Sasirekha, Associate Professor, Department of Computer Science, Vidyasagar College of Arts and Science, Udumalpet, Tamilnadu, India. Email:nsasirekhaudt@gmail.com

It is the commonly accepted encoding technique for storage and communication of almost all languages in the world [28 and 30].

D. ASCII Character Set

ASCII character set is a popular and most broadly used encoding technique for data transmission and storage of documents. Character set used in many text editors and word processors are ASCII code format. ASCII codes are classified as standard or basic ASCII characters and extended ASCII characters. The standard ASCII character set uses 7 bits for each character, whereas the extended ASCII characters take 8 bits. It is used to represent non-English characters, graphics symbols, and mathematical symbols. Text file stored in ASCII format are called ASCII files.

E. Huffman Encoding Technique

Huffman coding is a widely used lossless data compression algorithm [27]. In this algorithm, a variable-length code is assigned to the input which consists of diverse characters. The code length is associated to how frequently characters are occurred in the input stream of characters. Most frequent characters have the smallest codes and longer codes are assigned for least frequent characters. The output from Huffman's algorithm can be viewed as a variable-length code table for encoding a source symbol [17]. There are two major steps in Huffman coding are constructing a Huffman tree from the input characters and assigning codes to the characters by traversing the Huffman tree. A leaf node is created for all the given characters from the input. All the nodes are set in the increasing order based on the frequency value contained in the nodes. Huffman tree is produced by considering the first two nodes having minimum frequency. Based on that, it creates a new internal node having frequency equal to the sum of the two nodes frequencies and makes left child node. The other nodes are placed as a right child of the newly created node. Repeat the steps until all the nodes form a single Huffman tree.

II. RELATED WORKS

In paper [3] the authors Apte, Akash and Harshad explains the importance of Unicode encoding techniques and it is apt for Tamil software and documents. Each language has its own grammatical rule and properties and is different from English language.

The paper [14] describes the Tamil language alphabets and its classification as vowels and consonants.

The paper [18 and 28] explains about the Tamil Unicode characters which are widely used for transmission with the universal Industrial standards.

The author of paper [16] describes the difference of lossy and lossless compression techniques. Further, it explains that lossless algorithms preserve the same information so that the original data can be obtained at anytime and they are actually exploits the data redundancy in the original data.

Graefe, Goetz and Leonard, the authors of papers [13] used the text compression technique to improve database performance. They further extended that Data compression is widely used in data management to save storage space and network bandwidth.

In paper [21] the author presents a study of converting methods used in text compression of lossless by preprocess the text. This is performed by exploiting the inner redundancy of patterns in the source file. They made use of BWT that converts the original blocks of data into a format that is particularly well matched for compression. The block length is selected in a range different a type of text file was presented, evaluating the compression ratio and compression time.

Salomon in the book [25] explains all the compression techniques starting from text compression to video compression. He said that data compression is converting from input data stream to another smaller size data stream. The data compression is carried out for two reasons. One to save memory storage and another one is to make transmission of data faster.

Siva Jyothi Chandra et al in paper [30] explained about the creation of font by mapping ASCII character with Unicode characters. For Indian languages the combination of characters can be replaced by ASCII characters.

Storer and Jamws, the author of book [31] explains about the types of dictionary in detail. The classification of dictionary as static and dynamic and its subtypes are described.

Authors of paper [32 and 33] have implemented the static dictionary concept for compressing Tamil documents. Two static dictionaries were used for the compression and decompression purpose.

The importance of Huffman encoding technique was explained in paper [1]. The author added that Huffman coding is a famous example of coding redundancy. The idea is to assign variable length codes to combinations depending on the frequency of appearances of these combinations in the original data. Statistical way of method is used to combine the appropriate combination with the corresponding code.

Awan, Fauzia, et al. made use of static dictionary for encoding and decoding the input text file. The main concept is to encode all words in the text file given as input, which is also available in the English text dictionary. Based on the encoding a static dictionary was built. These dictionary words give shorter length for the input words and also preserve some context and redundancy.

Arafat and Enas of paper [4] presented a hybrid technique that uses the linguistic features of Arabic language to improve the compression ratio of Arabic texts. This technique works in phases, multilayer model-based approach and applying the Burrows-Wheeler compression algorithm.

In paper [29] the author describes the compression work for Czech language. It deals with usage of word order, word categories and grammatical rules in sentences and sentence units in Czech language. Special grammatical properties of this language which are different from English language are used. An algorithm was designed for searching similarities of sentence structures and later compression of file was performed.

Barua, Linkon, et al. of paper [6] explains about the text compression algorithm performs at the character level and the Bangla text has some unique features different from English language. He further extended that the conventional Lempel-Ziv-Welch (LZW) algorithm is not suitable for compressing Bangle text, so a modified LZW (MLZW) algorithm was used to compress Bangla text more effectively and efficiently.

Sajila Divakaran et al in paper [24] represented about compression algorithm converts the input message to a new form with a fewer number of bits by exploiting the probability distribution. A variable length encoding technique in which most probable Unicode character was represented by less number of bits.

Authors of paper [27] proposed a novel approach of constructing data compression dictionary of Gujarati text for the purpose of text compression.

Bhattacharjee, Arup Kumar et al. of paper [7] examines the performance of many algorithms of lossless data compression, on various forms of text data. The authors selected some algorithms. They implemented to evaluate the performance in compressing text data. For testing a set of defined text file are used.

In paper [20] Porwal, Shrusti, et al. compared various lossless compression algorithms based on the measurement parameters like compression ratio, compression factor, percentage of compression.

Pannirselvam, S., and D. Selvanayagi of paper [19] compared four different lossless compression algorithm. The efficiency are measured based on space and time taken for compression algorithm. They concluded that Huffman encoding technique gives better result in both compression ratio and in compression time.

III. OVERVIEW OF TAMIL COMPRESSION TECHNIQUE

Data Compression is the process of converting an input data stream to another data stream that has a smaller size [7]. Text compression using lossless technique enables the re-establishment of a file to its original state without the loss of a single bit of data, when the file is uncompressed [10]. The Unicode is the most acceptable industrial standards for storing, transmitting and documentation [2 and 12]. It was developed by combining the Universal Coded Character Set (UCS) standard and published as the Unicode Standard. The Unicode contains a repertoire of more than 128,000 characters. It covers over 135 modern and historic scripts, as well as multiple symbol sets in its latest version. Unicode is designed to represent almost all characters in every language in the world [30]. All the alphabets of Tamil language are now encoded as according to the universal principle of Unicode. The Tamil characters are range from U+0B80 to U+0BFF in Unicode character set [9 and 33]. It is large enough to encompass all characters that are likely to be used in general text interchange, including those in major international, national and industrial character sets. But Tamil Unicode characters occupy more space than ASCII characters in storage [15 and 22]. There are various lossless techniques available like run length encoding, Huffman encoding, Shanon Fanon encoding, arithmetic encoding and many dictionary based compression like LZW, LZ77 and LZ78.

Lossless compression involves in reduction of file size to smaller without any loss of single character. Compression is performed by replacing a large size character or stream of characters by a small one or removing repeated characters by a smaller one or rearrangement of characters to save memory storage space. Many compression techniques are available for many languages in the world. An efficient compression can be performed by careful analysis of the nature of documents or languages, since every language has its own features and specialty. An efficient compression technique for Tamil language is designed with a novel approach after knowing its basic nature.

The lossless Tamil text compression involves 3 basic steps. Steps in Tamil compression technique are shown in the Fig.1. Tamil document is given as input to the compression process. A Tamil document may commonly have the usage of English words or characters between the Tamil words or characters. So the first step involves separation of English characters from Tamil alphabets by placing a special separator. A tilde symbol '~' is placed before and after the English alphabets. This separator avoids the accidental wrong replacement of characters during decompression process at the later stage.

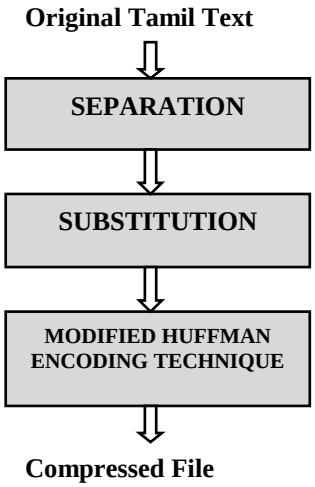

Figure 1: Steps in Tamil Compression Technique

The substitution process was performed in the second step. Here the Unicode Tamil character was replaced with ASCII characters. The static dictionary is used for this substitution purpose. The replacement of Tamil alphabets by ASCII characters was carried out in two stages using two separate tables in the static dictionary. At the end of substitution process the file size is reduced to an average of 49.7% [32]. This is due to the size occupied by Unicode Tamil character is 16 bits where as the size of basic ASCII character is 8 bits. Almost 49.7% of the file size is reduced.

The final step involves in building a Huffman tree based on the output produced from the second step. A tree was built in that instance with a modified Huffman encoding technique based on the frequency of word occurrences.

Huffman encoding technique is a famous greedy algorithm used to assign a variable length code to the given input data [27].

This algorithm improves the compression rate further after the substitution process. The outcome of the modified Huffman encoding technique is the compressed file which is stored in the binary form. The decompression is the reverse of compression process. It produces the original file without any loss of data.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

The novel lossless text compression for Tamil documents was executed by developing programs in ASP.NET. The results was extracted and analyzed based on memory capacity needed to store a compressed file and time taken to compress a file. The compression process was carried out using the steps of algorithm given in the figure 2.

BEGIN
1. A Tamil document file is given as input.
2. **BEGIN: SEPARATION PROCESS**
   i. Scan the entire file to find any English alphabets or words in the input.
   ii. If English alphabets found, separate it from Tamil alphabets by placing a '~' symbol before and after.
   END
3. **BEGIN: SUBSTITUTION PROCESS**
   i. The output of step 2 is given as input to step 3.
   ii. Replace the Tamil alphabets by basic ASCII characters using static dictionaries.
   iii. The result is the document with collection of ASCII characters which is the intermediate form of compression process.
   END
4. **BEGIN: APPLY MODIFIED HUFFMAN ENCODING PROCESS**
   i. The intermediate file obtained in step 3 is given as input.
   ii. Select the words from the input based on number of characters and repetition of words.
   iii. Build a dynamic Huffman tree with leaf node has selected words from left to right in decreasing order.
   iv. The leaf node words are encoded in the input file.
   v. The encoded file is the output stored in the binary form.
   END
5. The result of step 4 is the compressed file.

END

The compression process starts by giving a Tamil text file as input which is the collection of Unicode Tamil alphabets. The size of Unicode ranges from 2 bytes to 4 bytes based on the encoding type of file storage. Basic ASCII character occupies only one byte in memory for storage. The substitution process of compression technique involves the replacement of Unicode Tamil characters by ASCII characters using static dictionary. Before the substitution process begins the separation process of compression technique is used for separating English characters from Tamil by including a pair of special symbol before and after the English characters. The Fig. 3 and Fig. 4 show the file story2.txt before and after the separation and substitution process. The size of the file before compression is 10.9 KB. After the first step of compression the English characters are separated from the Tamil characters. A special symbol tilde is placed before and after the English alphabets. The second step involves substitution of ASCII characters in the place of Unicode Tamil characters. The Fig. 4 shows the intermediate file obtained after the separation and substitution process of compression of story2.txt.

தெனாலி ராமன் கதைகள் Moral stories for children
சுமார் நானூற்று எண்பது ஆண்டுகளுக்கு முன் ஆந்திர மாநிலத்தில் உள்ள ஒரு சிற்றூரில் ஓர் ஏழை அந்தணக் குடும்பத்தில் பிறந்தான் தெனாலிராமன். இளமையிலேயே அவன் தன் தந்தையை இழந்தான். அதனால் அவனும் அவனுடைய தாயாரும் தெனாலி என்னும் ஊரில் வசித்து வந்த அவனுடைய தாய் மாமன் ஆதரவில் வாழ்ந்து வந்தனர்.

சிறு வயதிலேயே அவனைப் பள்ளிக்கு அனுப்பியும் பள்ளிப்படிப்பில் அவனுக்கு நாட்டம் செல்லவில்லை. சிறு வயதிலேயே விகடமாகப் பேசுவரில் வல்லமை பெற்றான். அதனால் அவன் பிற்காலத்தில் "விகடகவி" என்னும் பெயர் பெற்று பெரும் புகழுடன் விளங்கினான்.

காளி மகாதேவியின் அருட்கடாட்சம் பெற்றவன். பின் வரலாற்றுப் புகழ்பெற்ற விஜயநகர சாம்ராஜ்யத்தின் அரசன் கிருஷ்ணதேவராயரின் அரண்மனை "விகடகவி"யாக இருந்து மன்னரையும் மக்களையும் மகிழ்வித்தான். அவனுடைய நகைச்சுவைக்காக மன்னர் அவ்வப்போது ஏராளமான பரிசுகளை அளித்து ஊக்குவித்தார்.

Figure 3: File before compression

The output of the separation and substitution process is given as input to the third step. A modified Huffman encoding technique is applied in the third step. A dynamic Huffman encoding tree was built with the leaf node contains the most frequently occurred words. The words are selected based on the word length and the number of frequency of occurrences of the word in the document. After the completion of building the Huffman tree and encoding, the entire document is compressed to the size of 2.64 KB. The percentage of compression for story2.txt is 75.79%.

```
sGuCMD yCwuB msImNB ~Moral stories for
children
~oEwCyB tCuFzBzE hrBvsE crBqEmNEmBmE wEuB
ctBsDy wCtDMsBsDMB fNBN kyE oDzBzFyDMB lyB
iOI btBsrmB mEqEwBvsBsDMB vDztBsCuB
sGuCMDyCwuB. dNwIxDMHxH bPuB suB stBsIxI
dOtBsCuB. bsuCMB bPuEwB bPuEqIx sCxCyEwB
sGuCMD huBuEwB gyDMB PoDsBsE PtBs bPuEqIx
sCxB wCwuB csyPDMB PCOBtBsE PtBsuyB.

oDzE PxsDMHxH bPuIvB vNBNDmBmE buEvBvDxEwB
vNBNDvBvqDvBvDMB bPuEmBmE tCqBqwB
oGMBMPDMBMI.
oDzE PxsDMHxH PDmqwCmvB vHoEPyDMB PMBMwI
vGzBzCuB. bsuCMB bPuB vDzBmCMsBsDMB
"PDmqmPD" huBuEwB vGxyB vGzBzE vGyEwB
vEmOEquB PDNnBmDuCuB.

mCND wmCsHPDxDuB byEqBmqCqBowB vGzBzPuB.
vDuB PyMCzBzEvB vEmOBvGzBz PDஜxtmy
oCwByCஜBxsBsDuB byouB mDyEஷBrsHPyCxyDuB
byrBwuI "PDmqmPD"xCm dyEtBsE wuBuyIxEwB
wmBmNIxEwB wmDOBPDsBsCuB. bPuEqIx
tmIoBoEPImBmCm wuBuyB bPBPvBvKsE iyCNwCu
vyDoEmNI bNDsBsE gmBmEPDsBsCyB.
```

Figure 4: Intermediate Form of File after the Separation and Substitution Process of compression

The Tamil compression technique performance was measured in terms of space efficiency and time efficiency. The space efficiency is calculated by finding the compression ratio, compression factor and percentage of compression [19].

The compression ratio is the ratio between size of compressed file and the original file represented in formula 1.

$$\text{Compression Ratio} = \frac{\text{Compressed File Size}}{\text{Original File Size}} \quad (1)$$

The compression factor is the reverse of compression ratio. It is shown in the formula 2. It gives the ratio between original file size and the compressed file size

$$\text{Compression Factor} = \frac{\text{Original File Size}}{\text{Compressed File Size}} \quad (2)$$

The percentage of compression is also called as saving percentage; because it gives the percentage of reduce in memory size from the original file size. The compression percentage is calculated by subtracting the size of original file and compressed file and then divided it by original file size using the formula 3

$$\text{Percentage of Compression} = \frac{\text{Bits Before Compression} - \text{Bits After Compression}}{\text{Bits Before Compression}} * 100 \quad (3)$$

The Table1 shows the size of original file, compressed file and decompressed file in bytes. The content of the table was arranged in the alphabetical order of file name. The comparison of original file size with the compressed file size in bytes was displayed in the Fig. 5.

The decompression process is applied to the compressed file to retain the original file. The reverse process of compression is applied for this purpose. Here, first the decoding of compressed file is performed using the Huffman tree. The second step involved in decompression is substitution process. In this process the ASCII characters was replaced by Unicode Tamil characters using the static dictionary. The final stage of decompression is removing the special character in the file this results in original file.

The output of decompression preserves the original data without any loss of a single character. The size of compressed file and decompressed file is shown in the Table 1. The decompressed file size is same as the original file size given for the input to compression process.

The Table 2 shows the measurement parameters like compression ratio, Compression factor and percentage of compression of Table 1 using the formula 1, 2 and 3 respectively. The average compression percentage was calculated for all the files in Table 2. The average percentage of compression for the files in the Table 2 is 72.08%.

Table 1: Size of Original File, Compressed and Decompressed File

File Name	Original File Size (bytes)	Compres-sed File Size (bytes)	Decompre-ssed File Size (bytes)
Bharathi	14746	4035	14746
Chennaithagaval	4260	1383	4260
Ettuthokkai	3072	904	3072
Nambikkai	6124	1782	6124
Natrrinai	5634	834	5634
Pathittrupathu	13074	2324	13074
Story1	3625	1178	3625
Story2	11167	2704	11167
Tamil_text	1065	501	1065
Vairamuthuvaralaru	26222	6444	26222

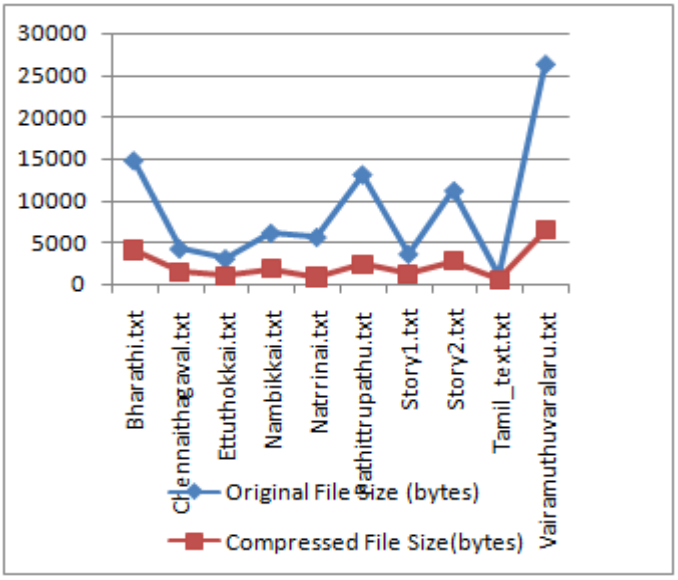

Fig. 5: Comparison of File size before and after compression

A compression algorithm performance is also measured in terms of efficiency of time. It is measured by finding the time required for compressing and decompressing the input file. The time taken for compression and decompression is not same. In this section the analysis was further extended by finding the speed of Tamil compression technique followed by space efficiency. It is observed that the compression process takes more time than decompression process. Table 3 lists the time taken for compression and decompression process of Tamil compression technique. The Table 3 shows the list of same files in Table 1 and 2 with the time for compression and decompression in seconds.

Table 2: Measurement Parameters of Compression

File Name	Compre-ssion Ratio	Compres-sion Factor	Percent-age of compre-ssion
Bharathi	0.274	3.655	72.64
Chennaithagaval	0.325	3.08	67.54
Ettuthokkai	0.294	3.398	70.57
Nambikkai	0.291	3.437	70.9
Natrrinai	0.148	6.755	85.2
Pathittrupathu	0.178	5.626	82.22
Story1	0.325	3.077	67.5
Story2	0.242	4.13	75.79
Tamil_text	0.47	2.126	52.96
Vairamuthuvaralaru	0.246	4.069	75.43

Compression time is the time taken by the algorithm to compress a file and the decompression time is time taken by the algorithm to decompress and to retrieve the original file from compressed file. These time parameter plays an important factor to transmit the data as well as for storage and retrieval of file from memory. Time taken for compression process is more when compared to the decompression processing time.

Table 3: Time taken for compression and decompression process

File Name	Time taken for Compression (Seconds)	Time taken for Decompression (Seconds)
Bharathi	0.19	0.09
Chennaithagaval	0.13	0.05
Ettuthokkai	0.1	0.05
Nambikkai	0.13	0.05
Natrrinai	0.08	0.04
Pathittrupathu	0.14	0.11
Story1	0.18	0.04
Story2	0.87	0.18
Tamil_text	0.05	0.04
Vairamuthuvaralaru	0.36	0.06

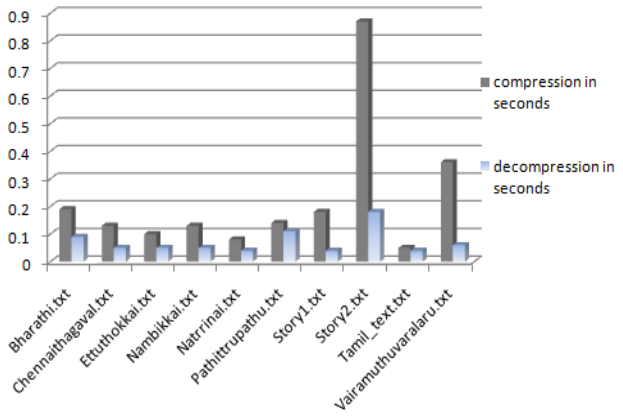

Figure 6: Comparison of time taken for compression and decompression process

The average compression time for the files in table 3 is 0.2 seconds (223482.6 microseconds) and for decompression process the average time taken is 0.07 seconds (72339 microseconds). Comparison of compression and decompression time for Tamil compression technique is shown as the bar chart in figure 6.

V. CONCLUSION

This lossless Unicode Tamil document compression technique surely paves a way to store the Tamil documents in a minimum storage space and improves the transmission speed also. This compression process works well even though the Tamil document contains English alphabets by separation process. Transforming text into some intermediate form by using static dictionary is used to achieve better compression ratio. The file size is reduced apparently after the usage of static dictionary which has the collection of Tamil alphabets and its ASCII replacement. The compression ratio is enhanced by using the modified Huffman encoding process further. The decompression process is performed successfully by restoring the original document without any loss of data. Various measurement parameters are calculated like compression ratio, factor and compression percentage to find the efficiency of space and time for the Tamil compression technique. The average percentage of compression is 72.08%. The average time taken to compress the file is 0.2 seconds and for decompression is 0.07 seconds. This technique can be applied to other Indian or foreign languages which have abugida type of syllabify of vowels and consonant symbol combinations.

VI. FUTURE ENHANCEMENT

The lossless Tamil document compression and decompression process works well.

The compression process can be improved by making use of other compression algorithm instead of Huffman encoding after the intermediate form of conversion process. This technique can be applied to other languages which has Unicode characters. The intermediate form of document obtained after the substitution process can be given as input to the existing compression tools like winzip, gzip and deflate, which will further enhances the compression ratio. The efficiency of space and time of lossless Tamil compression was compared with the existing compression tools to check its performance in future.

REFERENCES

1. AbuBaker, Ayman, Mohammed Eshtay, and Maryam AkhoZahia. "Comparison Study of Different Lossy Compression Techniques Applied on Digital Mammogram Images." IJACSA) International Journal of Advanced Computer Science and Applications 7.12 (2016).
2. Ajantha Devi, Dr.S.Santhosh Baboo, "Embedded Optical Character Recognition on Tamil Text Image Using Raspberry Pi." International Journal of Computer Science Trends and Technology, Vol 2, Issue4, Jul-Aug 2014.
3. Apte, Akshay, and Harshad Gado. "Tamil character recognition using structural features." (2010).
4. Arafat Awajan and Enas Abu Jrai, "Hybrid Techniques for Arabic Text Compression", Global Journal of Computer Science and Technology:C Software and Data Engineering, Vol 15 Issue 1 Version 1.0 2015, Print ISSN: 0975-4350 (2015).
5. Awan, Fauzia S., et al. "LIPT: A Reversible Lossless Text Transform to Improve Compression Performance." *Data Compression Conference*. 2001.
6. Barua, Linkon, et al. "Bangla text compression based on modified Lempel-Ziv-Welch algorithm." Electrical, Computer and Communication Engineering (ECCE), International Conference on. IEEE, 2017.

7. Bhattacharjee, Arup Kumar, Tanumon Bej, and Saheb Agarwal. "Comparison study of lossless data compression algorithms for text data." IOSR Journal of Computer Engineering (IOSR-JCE) 11.6 (2013): 15-19.
8. Blelloch, Guy E. "Introduction to Data Compression." Computer Science Department, CarNegie Mellon University (2001).
9. Dr.J.Venkatesh and C.Sureshkumar, "Tamil Handwritten Character Recognition Using Kohonon's Self Organizing Map", International Journal of Computer Science and Network Security, Vol. 9 No. 12, December 2009.
10. Frank E., Chang Chui, andlan H. Witteh. "Text categorization using Compression Models."(2000). Http://online.redwoods.cc.ca.us/instruct/darnold/LAPROJ/Fall98/PKen /dct.pdf (Accessed on 28/1/2020).
11. Fu, Chi-Yung, and Loren I. Petrich. "Image compression technique." U.S. Patent No. 5,615,287. 25 Mar. 1997.
12. Gleave, Adam, and Christian Steinruecken. "Making compression algorithms for Unicode text." arXiv preprint arXiv:1701.04047 (2017).
13. Graefe, Goetz, and Leonard D. Shapiro. "Data compression and database performance." Applied Computing, 1991, [Proceedings of the 1991] Symposium on IEEE, 1991.
14. Hewavitharana, S., and H. C. Fernando. "A two stage classification approach to Tamil handwriting recognition." Proc. TI (2002).
15. Juul, Svend, and Morten Frydenberg. "UNICODE2ASCII: Stata modules to translate between Unicode and ASCII." Statistical Software Components (2016).
16. K. Cabeen, & P. Gent. "Image compression and the discrete cosine transform." Journal of Image and Information Processing, Vol. 4 No. 4 (October 9, 2013).
17. Kodituwakku, S. R., and U. S. Amarasinghe. "Comparison of lossless data compression algorithms for text data." Indian journal of computer science and engineering 1.4 (2010): 416-425.
18. Kuo, Shihjong. "Processors, methods, systems, and instructions to transcode variable length code points of unicode characters." U.S. Patent No. 9,626,184. 18 Apr. 2017.
19. Pannirselvam, S., and D. Selvanayagi. "A Comparative Analysis On Different Techniques In Text Compression." International Journal of Innovative Technology and Creative Engineering, ISSN:2045-8711, Vol.5 No.8 (August 2015).
20. Porwal, Shrusti, et al. "Data compression methodologies for lossless data and comparison between algorithms." International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 2 (2013): 142-147
21. Radescu, Radu. "Transform methods used in lossless compression of text files." Romanian Journal of Information Science and Technology 12.1 (2009):101-115.
22. Raja, J. Nelson, P. Jaganathan, and S. Domnic. "A new variable-length integer code for integer representation and its application to text compression." Indian Journal of Science and Technology 8.24 (2015).
23. Ramachandran, Raj, and Ashik Ali. "Social challenges faced by technology in developing countries: focus on Tamil Nadu State." (2017).
24. Sajilal Divakaran, Biji C.L., Anjali. C, Achuthsankar s. Nair, "Malayalam Text Compression", International Journal of Information Systems and Engineering, Vol 1, No. 1, (April 2013).
25. Salomon, "Data Compression: The Complete Reference", Springer, Published by Springer Science and Business Media, 3rd edition (2004).
26. Sandip V Maniya, MJ Sheth, K Lad - pdfs.semanticscholar.org, "Compression Technique based on Dictionary approach for Gujarati Text", International Journal of Engineering Research and Development eISSN : 2278-067X, pISSN : 2278-800X, www.ijerd.com Volume 4, Issue 8, PP. 101-108 (November 2012).
27. Sangwan, Nigam. "Text encryption with huffman compression." International Journal of Computer Applications 54.6 (2012).
28. Seethalakshmi.R, Sreeranjani.T.R, Balachandaar.T, "Optical Character Recognition for Printed Tamil Text using Unicode", Journal of Zhejiang University Science, ISSN 1009-3095, 2005 6A(11):1297-1305 (2005).
29. Ševčík, Jiří, and Jiří Dvorský. "Techniques of Czech Language Lossless Text Compression." IFIP International Conference on Computer Information Systems and Industrial Management. Springer International Publishing, 2016.
30. Siva Jyothi Chandra, Ashlesha Pandhare, Mamatha Vani, "Multilingual Font Creation by Mapping Unicode to Ascii", International Journal of Advanced Research in Computer Science and Software Engineering, Vol 5, Issue 9, Sep 2015, ISSN: 2277 128X (2015).
31. Storer, Jamws A., ed. Image and text compression. Vol. 176. Springer Science & Business Media, 2012. https://www.springer.com/gp/book/9780792392439 (Accessed on 2/2/2020).
32. Vijayalakshmi, B., and N. Sasirekha. "Lossless Text Compression Technique Based on Static Dictionary for Unicode Tamil Document." International Journal of Pure and Applied Mathematics Vol 118, Issue 7, 2018.
33. Vijayalakshmi, B., and N. Sasirekha. "Lossless Text Compression For Unicode Tamil Documents." ICTACT Journal on Soft Computing 8.2 (2018).

AUTHORS PROFILE

Mrs. B. Vijayalakshmi received her Bachelor degree in Computer Science. She completed her M.Sc. degree in Computer Science at V.L.B.Jannaki Ammal College of Arts and Science, Coimbatore and secured second rank in Bharathiar University. She obtained her M.Phil.(CS) in the area of Mobile Computing. At present she is working as an Assistant Professor in PG Department of Computer Science in Vidyasagar College of Arts and Science, Udumalpet, Tirupur district. She is pursuing Ph.D. in Computer Science in the area of Data mining. Her main interest is to develop application for Tamil document compression which will be useful to the society.

Dr. N. Sasirekha, completed MCA., M.Phil., Ph.D in Computer Science. She is currently working as Associate Professor, PG and Research Department of Computer Science, Vidyasagar College of Arts and Science, Udumalpet, Tamilnadu, India. She has thirteen years of teaching experience and presented more than twenty five papers in various National/International Conferences /Seminars. She has published twenty five research articles in various reputed International Journals. Her area of research is Software Engineering, Security, Data Mining, and Image Processing. She is also a Reviewer, Programme Committee member, Editorial board member in several International Conferences and Journals. She is a member of Computer Society Teachers Association, Universal Association of Computer and Electronics Engineers, International Association of Engineers and Computer Scientists and International association of Computer Science and Information Technology.