EFFICIENCY LOSSLESS DATA TECHNIQUES FOR ARABIC TEXT COMPRESSION

Enas Abu Jrai

Department of Basic sciences, Ma'an University College, Al-Balqa Applied University, Ma'an- Jordan

ABSTRACT

Study and evaluate the efficiency of LZW and BWT techniques for different categories of Arabic text files of different sizes. Compare these techniques on Arabic and English text files is introduced. Additional to exploiting morphological features of the Arabic language to improve performance of LZW techniques. We found that the enhanced LZW was the best one for all categories of the Arabic texts, then the LZW standard and BWT respectively.

KEYWORDS

Text compression, Arabic Texts, BWT, LZW.

1. INTRODUCTION

Data compression has important applications in the areas of data transmission and data storage. It aims at reducing the size of data in order to improve the speed of transmission and reduce the size that is needed for the storage. Text compression is a subfield of data compression; it focuses on compressing natural language texts as they occur in real world.

There are many techniques used to compress data in general and the texts in particular. Some of these techniques proceed at the level of characters. They use the frequency of characters in order to replace the most frequent characters by short codes. Therefore, they are called statistical compression methods. Examples of this category of techniques are the Run Length Coding and Huffman Coding. Other techniques look at strings in the text and put pointers to strings or substrings that have been already appeared. These techniques are called dictionary based techniques. Example of this category of compression techniques is Lempel-Ziv Codes (LZ). There are also techniques which look at the frequency of the character and at the character that occurs at its nearby when they decide how to encode a character. Examples of the last category are Burrows Wheeler Transform (BWT) and Prediction by Partial Matching (PPM).

Developing new compression techniques based on the morphological and grammatical features of Arabic and other Semitic language may present a new paradigm which will be able to improve the compression ratio and performance [1-4].

This paper aims to study two different methods of data compression techniques, and to compare their performance on Arabic and English text files. Also to exploit one of morphological features of the Arabic language, Citing way [5] (Omer and Khatatneh ,2010) to improve performance of Lempel Ziv Welch (LZW) techniques.

DOI:10.5121/ijcsit.2014.6505

2. RELATED WORK

Two approaches of research on Arabic text compression can be found in the literature. The first approach considers the general purpose techniques and do not take into account the features of Arabic languages.

A study for a variety of data compression techniques had presented by Khafagy [6]. This study was applied on English or Arabic texts for compression. The best compression ratio had been obtained in case of Neural Compression. Next come the PPM, Lempel–Ziv–Storer–Szymanski (LZSS), LZ77, Lempel–Ziv-Welch (LZW), Lempel–Ziv Ross Williams (LZRW), Huffman techniques. Then finally the RLE program. On the other hand, PPM and Neural technique achieved the best compression time compared to other techniques. In Neural Network, a network structure was suggested with the detailed information which included a number of layers, number of nodes in the layers and the weights between them. Cascading was also observed and it showed negligible advances in performance.

Ghwanmeh, at el [7] employed the dynamic Huffman coding with variable length bit coding to compress Arabic texts. They proved that this approach was able to improve the efficiency of compression of Arabic text much better than compression of English text. They showed that the frequency of symbols on the text affected the compression ratio and the average message length.

Alasmer, at el [8] had presented a comparison between Static Huffman and LZW techniques on Arabic and English texts had presented . The results had shown that Huffman got better and efficient results in Arabic. An LZW technique was better in English documents especially when it was converted into a binary file.

The second one uses the features of Arabic language to develop new compression techniques in order to improve the compression ratio. Some research work had used the morphological and grammatical features of Arabic language to improve the compression performance of the data.

Wiseman and Gefner [1] suggested a suitable technique for Semitic languages. This technique includes two phase: at the first phase, morphological analysis is used to segment the text into two files. The first file includes index values for each pattern. The second file includes the root of the words. The second phase included compressing both files, using traditional BWT. The Hebrew bible file had been used for testing the compression. The size of file was 260 KB. Results showed that using the technique which had been suggested achieved 28.97% of compression rate whereas the traditional BWT compresse this file to 40.13%.

Daoud [2] had presented a hybrid technique to compressing diacritical Arabic texts, benefiting from Arabic morphological features for split the entered word into root and infix, and exploited the unused 128 location of the ASCII, to represent the most frequent articles, root and affixes. The optimal Huffman compression algorithm was used to measure performance of the proposed techniques which improvements 20% of the traditional technique.

In 2011 there was another technique suggested by Akman et al [3]. It was a suitable technique to compress Semitic languages. It depended on syllable-based morphology. This technique had been implemented for the Turkish language. It is tested of files of sizes ranging from 4.6 to 725 KB; the results had showed that a compression ratio up to 43% was a achieved.

Awajan [4] used the morphological structure of words to build a multilayer model of the Arabic text. The proposed model worked in two phases: the morphological analysis phase and the encoding phase. Text was split into three categories of words: derivative, non-derivative and

functional words. A fourth layer, called the Mask, introduced to aid with the reconstruction of the original text from the three layers in the decoding side. Both the function words and the derivative words (which were divided into pattern and root of the word) had been replaced with their index value. The third category of words has compressed using Huffman coding. The results of this Multilayer model had showed that its performance was better than many of the traditional compression techniques applied on the whole text. Another advantage of this work was that the layer file reserved for derivative words might be used by the research engine and the information retrieval systems to find out terms or word by looking on their root or stem without the need to decode the compressed text.

Omer and Khatatneh [5] used the fact that Arabic letters have a single case to present a new technique which had been applied on Arabic short text. Huffman technique was used to measure the efficiency of the proposed technique. Two passes had been implemented on the input text through: (i) calculating the frequency each of character. (ii) Assigning numerical code between 0 and 127, but checking if the numerical value >=64 encode the character using 7 bits else using 6 bits. This approach outperformd the two pass Huffman compression.

3. BURROWS-WHEELER TECHNIQUE

The BWT technique was invented by Michael Burrows and David Wheeler in 1994. It applies BWT then Move to Front transformation (MTF) to reduce the redundancy of letters and converts the original blocks of data into a format that is extremely well suited to perform the compression technique, usually Arithmetic coding or Adaptive Huffman technique is used after Run-length encoding (RLE) [9]. We have suggested Adaptive Huffman technique to apply in our work, as shown in Fig. 1.

Fig. 1 Steps of the Burrows-Wheeler Compression technique.

4. STANDARD LZW TECHNIQUE

LZW is dictionary based technique. The main idea of LZW is to replace strings of characters with single codes, usually the first 256 codes are used as the standard character set (for the case of 8-bit characters).

The encoder begins with a string table that contains all ASCII characters from (0-255). It reads one character each time from the input file, then it checks if the string is in the table. If this is case, the fixed code value will get without any analysis. If the string is not in the table, will add it as a new entry at its end, and output the last string.

5. IMPROVEMENT OF THE PERFORMANCE OF LZW TECHNIQUE (LZWA)

The most important characteristic of Arabic language characters is that each character has a single case (" أ"-"ي ") , unlike the English characters where each character has two cases, the upper case ("A"-"Z") and the lower case ("a"-"z"). Therefore we have suggested to take advantage of this feature to reduce number of bits by representing each character and symbol by

the index number (0-73), which has the lowest number of bits compared with their ASCII code (170-239). So the compression process applies the following steps:

1) Reading the text and entered it into the memory
2) Reading one character each time from memory.
3) Examining if the character exists in the dictionary (where each character has an index number), replacing the character with it's index value (7-bit) rather than ASCII code (8-bit). If isn't found, add it to the dictionary and assign a new index to it. This approach is efficient compression scheme to compress short Arabic texts

6. EXPERIMENTS AND RESULTS

Two groups of experiments were implemented out on Arabic texts. The first group of experiments aimed to compare and evaluate the efficiency of the Lossless data compression techniques for Arabic and English texts. The second group of experiments aimed to exploiting morphological features of the Arabic language to improve performance LZW.

We tested several files, which had different sizes. These files had been taken from multiple resources from the internet. Compression ratio (CR) is calculated for measuring the performance of the LZW and BWT techniques. The experiments were carried out on Laptop equipped with a Pentium (R) Core CPU of 2.20 GHz and 2GB RAM, running MS Windows 7 (32-bit) operating system. The source code was written for each technique in vb.net.

A. The First Group of Experiments

Several experiments had been performed to Study of Efficiency LZW and BWT Techniques in Arabic Text Compression. Each technique had been applied on all Arabic texts categories (vowilized, partial vowilized, non vowilized), and on English texts.

Fig.2 shows results of the compression ratio using LZW compression technique for all categories of Arabic texts. We note that LZW is more suitable to compress vowelized texts than the other categories, followed by compressing unvowelized then partially texts respectively. Additionally we concluded that LZW is suitable to compress Arabic texts (for all of their categories) more than English texts.

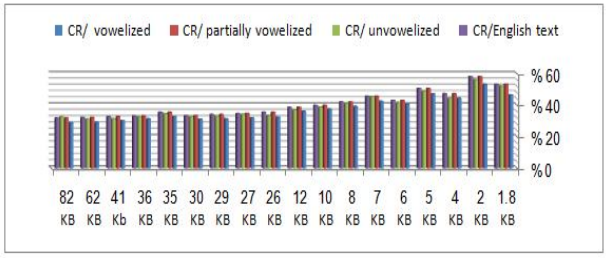

Fig. 2 Compression Ratios on Arabic and English text using LZW

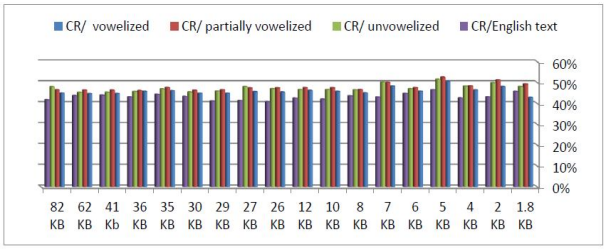

Fig. 3 Compression Ratio on Arabic and English Text using BWT

According to the results shown in Fig.3, we found that BWT is more suitable to compress vowelized texts than other categories. Compression ratio for unvowelized and partially texts were relatively similar. The results also showed that BWT performance is better just to compress vowelized Arabic text than for English texts

B. Results Analysis

The performance of the data compressor depends on: the features of the files, the different symbols contained in it, and symbols frequencies. From the first experiment, it is was noted that compression efficiency of Arabic texts using LZW was very high for all vowelization states, and it give better results than when it was used for compressing English texts.

The most important features of the Arabic language is that it has a single case of each letters, unlike the English language which has 26 letters, and each one has two cases (upper and lower case), In addition, the diacritical marks, which their number is low, just 9 and has high frequencies, hence the ratio of the compression rate of an Arabic text is better than the compression of an English text, especially when the Arabic text is for vowelized texts.

From the second experiment, it was noted that the BWT technique was suitable just to compress vowelized texts compared with English texts because it contained of the diacritical marks. BWT technique was very sensitive to the structure of the word. The Arabic language is rich in morphology, i.e. the origin of the word changed to several forms varying from the original form of the word. Unlike the English language where the origin of the word is not changed to several forms varying from the original form of the word the increase might be in end or the beginning of the word [4].

Table.I Compression ratio for LZW and BWT techniques

.

Texts type	Text category	LZW	BWT
Arabic texts	vowelized texts	37%	37%
	Un vowelized texts	39%	42%
	Partilly vowelized texts	40%	42%
English texts		40%	41%

According to the results shown in Table.1 the compression ratio using LZW is better technique to compress Arabic texts, than BWT technique. The difference in the compression ratio is related to the different mechanisms of the techniques in the compression process. LZW depends on replace strings of characters with single codes. BWT depends on preprocessing the text before the compression process by data compression technique.

C. The Second Group of Experiment

Depending on the results of the first experiments of techniques, which aimed to evaluated compressing efficiently of Arabic texts, both LZW and BWT will be used to test and evaluate the proposed method, which aims to improve performance the LZW to compress the Arabic texts.

Table.II Compression ratio for LZWA techniques.

Text Category	File Size	LZW	BWT	LZWA
Vowelized texts	Text1 (310 KB)	0.30	0.27	0.26
	Text2 (570 KB)	0.30	0.30	0.27
Unvowelized texts	Text3 (200 KB)	0.31	0.30	0.27
	Text4 (558 KB)	0.32	0.36	0.30
Partilly vowelized texts	Text5 (216 KB)	0.33	0.38	0.27
	Text6 (360 KB)	0.35	0.39	0.30

Table.2 shows the files size and the compression ratio for each file for each technique. It is noted that LZWA was the best for all categories of the Arabic texts, then LZW standard and BWT respectively. This technique most suitable for small flies and mobile phone, but we must note that, LZWA suitable just for Arabic texts because the dictionary contains characters of Arabic languages.

7. CONCLUSION

Two techniques of data compression had been compared and evaluated. They were tested on the different category of Arabic texts (vowelized, partially vowelized and unvowelized). Their performances were studied in term of compression ratio. After testing those techniques on different files of different sizes, we can conclude that LZW was the best one for all categories of the Arabic texts.

The English texts had been used to compare the performance of the two techniques against their performance with Arabic texts. The results had shown that LZW is suitable to compress Arabic texts (for all of their categories) more than English texts. But BWT performance is better just to compress vowelized Arabic text than for English texts.

The fact that Arabic letters have single case had been used in improve the performances of LZW. The proposed approach (LZWA) had achieved a better compression ratio compared with LZW and BWT.

REFERENCES

[1] Akman, I. , Bayindir, H. , Ozleme, S. , Akin, Z. and Misra, Sanjay (2011). "Lossless Text Compression Technique Using Syllable Based Morphology". The International Arab Journal of Information Technology, VOL.(8) No. (1). pp (66-74).

[2] Awajan,.A (2011). "Multilayer Model for Arabic Text Compression", The International Arab Journal of Information Technology, Vol.(8) No(2), pp (188-196).

[3] Daoud, A. M. (2010). “Morphological Analysis and Diacritical Arabic Text Compression”, The International Journal of ACM Jordan (ISSN 2078-7952), Vol.(1) No (1), pp (41-49).

[4] Wiseman Y. and Gefner I. (2007), "Conjugation-based Compression for Hebrew Texts", Computer Journal of ACM Transactions on Asian Language Information Processing, vol.(6), No.(1) ، pp. (1-10).

[5] Omer, E. and Khatatneh, .K (2010). "Arabic Short Text Compression". Journal of Computer Science, Vol.(6) No(1), pp (24-28).
[6] Khafagy , M. A. M. (2005)."Arabic Text Data Compression", PhD thesis, Zagazig University.
[7] Ghwanmeh, S., Al-Shalabi, R. , and Kanaan, G. (2006). "Efficient data compression scheme using dynamic Huffman code applied on Arabic language". Journal of Computer Science, Vol.(2), pp (885-888).
[8] Alasmer , Z. M. , Zahran B. M., Ayyoub B. A., Kanan M. A. (2013)."A Comparison between English and Arabic Text Compression". Journal of Contemporary Engineering Sciences, Vol. (6), No. (3), pp. (111-119).
[9] Blelloch, G. E., (2010). Introduction to Data Compression, Computer Science Department Carnegie Mellon Universit. 22-41 (On-Line), available: http://www.cs.cmu.edu/~guyb/realworld/compression.pdf , Last Visited 2014.

Author

Enas abu jrai is faculty member at Al-balqa Applied university, Maan, Jordan. She received Master degree in Computer Science from the University of Middle East - Jordan in 2013. Her research interests include text compression, and Keystroke Dynamic.