Journal of Ubiquitous Systems & Pervasive Networks
Volume 15, No. 1 (2021) pp. 11-15

Multistage Arabic and Turkish Text Compression via Characters Encoding and 7-Zip

Tariq Abu Hilal[a*], Hasan Abu Hilal[a], Ala' Abu Hilal [b]

[a]*Higher Colleges of Technology, Abu Dhabi, UAE, 41012*
[b]*Zayed University, Abu Dhabi, UAE, 41012*

Abstract

Turkish lossless text compression was proposed by converting the character's from UTF-8 to ANSI system for space-preserving. Likewise, we present a decoding method that transforms the encoded ANSI string back to its original format. Unlike the one-byte ANSI characters, some of the Turkish alphabets are being stored in 2 bytes size. All that space comes at a price. The developed sequential encoding technique will reduce the size of the text file up to 9%. Moreover, the Turkish encoded text will retain its original form after decoding. According to our proposal, it is considered as a lossless text compression, where it's a common concern today. Thus, many parties have become interested in Unicode compression. Basically, our algorithm is mapping Unicode Turkish characters into ANSI, by using the available 8-bit legacy. For Arabic Text Compression, a sequential encoding technique was suggested that efficiently converts Arabic characters string from UTF-8 to ANSI characters coding. The encoding algorithm presented in this paper significantly reduces the file size. The decoding method transforms the encoded ANSI string back to its original format. Unlike the one-byte ANSI characters, Arabic alphabets are currently being stored in 2 bytes size which leads to inefficient space utilization. The newly developed sequential encoding technique reduces the space required for storage up to fifty percent. In addition, the proposed technique will retain the Arabic encoded text to its original form after decoding, which is leading to a lossless text compression. Thus, addressing the common concern of the currently available Arabic characters compression techniques.
In this research, a multistage compression process was implemented on Turkish and Arabic languages, by using the new encoding technique, in addition to the 7-Zip application, which has shown a significant file size reduction.

Keywords: Unicode; ANSI; UTF-8 Encoding; Turkish Text Compression; Arabic Text Compression; 7-Zip Application.

1. Introduction

Character encoding is exceptionally fundamental for standardization. For worldwide communications, characters are usually bytes encoded. In the current literature, there are numerous encoding strategies that shift from straightforward to direct. Over decades and around the world, a long list of international and national encodings has been built up, which covers various sets of characters. The foremost complicated and the biggest character encoding records have been created are right now conveyed in Asia [1].

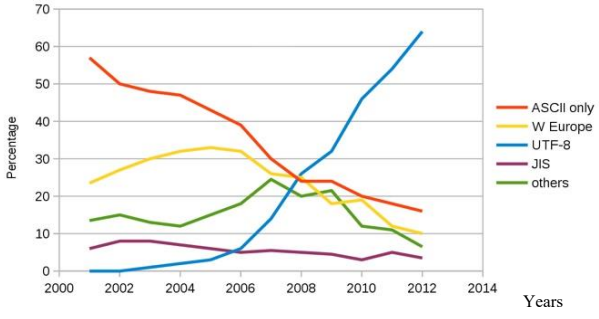

Fig. 1. Webpage encoding measurements

As of now, characters are more than one 8-bit byte, and this requires bigger spaces within the storage. Subsequently, many analysts are concerned with the text and space content files utilization. Indeed, in an age of cheap capacity, sending and transmitting Unicode information possesses very huge space. The Unicode encoding scheme UTF-8 is the foremost commonly utilized scheme. In fact, this character set is embraced by 87.0% of the websites. While the bequest ASCII encoding represents each character by a single byte, UTF-8 maps non-ASCII characters to groupings of two to four bytes by a variable width character encoding competent of encoding all possible characters. Numerous websites are presently in dialects other than English and have multi-byte UTF-8 characters. However, text of content compression has to be adjusted in the same way, as the most compression procedures still work on single bytes. To consider and address this issue, the Unicode Consortium characterized the Standard Compression Plot for Unicode (SCSU) and Parallel Requested Compression for Unicode. [1]

* Corresponding author. Tel.: +9712 206 2545
Fax: +9876543210; E-mail: tabuhilal@hct.ac.ae

DOI: 10.5383/JUSPN.15.01.002

Fig. 1. demonstrates the utilization of the most encodings on the internet from 2001 and ahead as found by Google. UTF-8 surpassing all others in 2008 and over 60% of the internet in 2014. UTF 8 has been the driving character encoding procedure for the Web since 2009, and in December 2018 it got to be the source for more than 92% of most of the internet pages (exceptionally few were in ASCII, as it's a portion of UTF-8) and more than 95% of the top 1,000 beat positioned web pages. The internet mail consortium proposed that all mail programs are able to appear and make sends by utilizing UTF-8, and the W3C proposes UTF-8 as the default encoding in XML and HTML [2] [3].

The organization of the paper is as follows. Section 2. Shows the literature review and general concept about the topic. Section 3. Explains the Turkish character coding scheme. In section 4. We demonstrate the proposed Turkish character. Characters Unicode system is explained in section 5. Section 6 and 7 shows the proposed Arabic system. Finally, conclusions are drawn in section 8.

2. Literature Review

The volume of 21-bit values for the encoding was considered as a major disadvantage within the framework in 2003. Essentially, UTF-8 encoding method is built up to four bytes measure considering the grouping of assigned bits at the beginning point. Table 1 portrays the dispersion of the encoding plans. The x characters are substituted by the bits of the code point at the starting. Depending on the number of critical bits in a string, in case less than or rise to 7, the primary line is taken; in case less than or rise to eleven bits, the moment line applies, and so on [4].

Number of bytes	Bits for code point	First code point	Last code point	Byte 1	Byte 2	Byte 3	Byte 4
1	7	U+0000	U+007F	0xxxxxxx			
2	11	U+0080	U+07FF	110xxxxx	10xxxxxx		
3	16	U+0800	U+FFFF	1110xxxx	10xxxxxx	10xxxxxx	
4	21	U+10000	U+10FFFF	11110xxx	10xxxxxx	10xxxxxx	10xxxxxx

Fig. 2. Encoding Structure

ASCII stands for "American Standard Code for Information Interchange" and was made by the American Benchmarks Affiliation (afterward renamed the American National Benc Standard Institute). The ASCII standard was begun in 1960 and discharged in 1963. It was an expansion of transmitted codes and was initially utilized by Bell labs. In arrange to back a wider swath of dialects, the Unicode encoding pattern was formulated in conjunction with the Widespread Character Set. Unicode incorporates a couple of encoding sorts, UTF-8 is the 8-bit encoding which has compatibility with ASCII, and which has risen to replace ASCII as the transcendent character encoding standard on the net nowadays. One byte is required to speak to the primary 128 characters (ASCII). Besides, 1920 characters requires two bytes to be spoken to which covers the leftover portion of all Latin script letter sets, additionally Greek, Japanese, Azerbaijani, Turkish, Arabic and etc. Three bytes are required for characters within the rest of the distant east dialects in addition to the musical notes, which contains virtually all characters in common use including most Chinese and Korean characters. Four bytes are needed for characters in the other domains of the Unicode, such as historic manuscripts, mathematical symbols, and emoji [5].

The encoding name of UTF-8 is used by all the standards conforming to the internet Assigned Numbers Authority (IANA) which means all HTML, CSS, and XML. IANA is a department of the larger ICANN, which is the non-profit which determines internet protocol and domain names [6].

As there was no work that considering Turkish language compression other than the commercial compression applications (i.e., 7- Zip, WinZip and WinRar), this work has studied and compared the results to the existing techniques, which showed significant decrement in the new compressed files.

A a	B b	C c	Ç ç	D d	E e	F f	G g
a	be	ce	çe	de	e	fe	ge
[a]	[b]	[ʤ]	[ʧ]	[d]	[e]	[f]	[g/ɟ]
Ğ ğ	H h	I ı	İ i	J j	K k	L l	M m
yumuşak ge	he	ı	i	je	ke	le	me
[Ø/j]	[h]	[ɯ]	[i]	[ʒ]	[k/c]	[ɬ/l]	[m]
N n	O o	Ö ö	P p	R r	S s	Ş ş	T t
ne	o	ö	pe	re	se	şe	te
[n]	[o]	[ø]	[p]	[ɾ]	[s]	[ʃ]	[t]
U u	Ü ü	V v	Y y	Z z			
u	ü	ve	ye	ze			
[u]	[y]	[v]	[j]	[z]			

Fig. 3. Turkish Alphabets

3. Turkish Characters Unicode

In the world of computers, character encoding strategy is the foremost fitting working standard utilized to stand for a collection of characters for both of the communications [7]. It was created in conjunction with the Universal Coded Character Set (UCS) standard and distributed as the Unicode Standard. The most recent adaptation of Unicode contains a collection of more than 128,000 characters covering 135 advanced and memorable scripts, as well as different image sets, numerous encoding sorts like ANSI, UTF-8, UTF-16, UTF-13 etc. are accessible [8] [9].

Unicode is intended to communicate to practically every one of the characters in each language on the planet. Every one of the characters of Turkish language are presently encoded according to the Universal Principle of Unicode [10]. It is sufficiently enormous to incorporate all characters that are probably going to be utilized, incorporating those in significant global, national, and industry character sets. Unicode systems consume more space in memory during capacity [11]. Fig.2 shows
the ongoing Unicode adaptation for Turkish Unicode character set.

The most well-known compression codes that pack records like WinZip and WinRar were created to lessen the extra storage memory. Nonetheless, they are generally managing European dialects. Also, numerous different organizations are generally utilized for packing singular documents, yet they are not supporting Turkish dialects proficiently. The serious issue will be in performing decompression process where characters are wrongly decoded which gives an inane yield. Numerous updates and upgrades were considered for improvement. For example, text compression for other eastern languages was carried out with the dictionary approach and bit replacement reduction technique [12].

The Turkish alphabet (Turkish: Türk alfabesi) is a Latin-script alphabet used for writing the Turkish language, consisting of 29 letters, seven of which (Ç, Ş, Ğ, I, İ, Ö, Ü) have been modified from their Latin originals for the phonetic requirements of the language.
Latin alphabet No. 5 is part of the ISO/IEC 8859 series of ASCII-based standard character encodings, first edition published in 1989. It is informally referred to as Latin-5 or Turkish. It was designed to cover the Turkish language, designed as being of more use than the ISO/IEC 8859-3 encoding. It is identical to ISO/IEC 8859-1 except for these six replacements of Icelandic characters with characters unique to the Turkish alphabet.
ISO-8859-9 is the IANA preferred charset name for this standard when supplemented with the C0 and C1 control codes from ISO/IEC 6429. In modern applications Unicode and UTF-8 are preferred. Since August 2019, 0.1% of all web pages use ISO-8859-9.[1][2]
Microsoft has assigned code page 28599 a.k.a. Windows-28599 to ISO-8859-9 in Windows. IBM has assigned Code page 920 to ISO-8859-9. It is published by Ecma International as ECMA-128.[3]

Table 1. Turkish Compression Results

The Turkish Text File (.txt)	The Original Turkish File Size (KB)	Compressing the Original File by Encoding (KB)	Compressing the Original File by 7-Zip (KB)	Compressing the Original File by Encoding and 7-Zip (KB) - Multistage	The Enhancement Ratio of Compressing the Original File by 7-Zip to Multistage
File 1.txt	142	130	48.0	47.2	1.69 %
File 2.txt	380	348	99.1	97.6	1.53 %
File 3.txt	218	200	73.4	72.3	1.52 %
File 4.txt	131	121	44.9	44.2	1.58 %
File 5.txt	318	291	101.0	99.9	1.10 %

4. The Proposed Turkish Encoding Technique

The suggested framework includes processing an ASCII character instead of a Unicode Turkish character, since the size of an
ASCII character is one byte (8 bits) while a Unicode character size range between 1 byte (8 bits) to 4 bytes (32 bits) relies upon the encoding system [13].
To manage storing the document with .txt extension, the accompanying encoding types are accessible, they are ANSI encoding, UTF encoding, Unicode and Unicode huge endian encoding. The bits required to store each character for the above encoding document type are 8 bits, 8 to 32 bits, 16 bits and 16 to 32 bits separately.

As known, Turkish language containing 29 letters, 22 of them are already one-byte Latin characters that cannot be reduced, and seven of them consists of two-byte characters (Ç, Ş, Ğ, I, İ, Ö, Ü). The idea is to convert the two-byte ones to one-byte for each to save space. The compression and decompression processes were developed as a Python application, three phase functions were built to compress and decompress any given text. The process of the implemented algorithm is shown in Figure 3. The first step is to read the Turk text file and classify all characters, keeping the one-byte ones the same, and converting the two-byte others (seven Turk letters) to a new assigned one-byte character. Then, generate the new ANSI encoded text file. The decoding algorithm is reversing this process.

Fig. 3. Encoding Algorithm Phases

5. Characters Unicode

In computing, character encoding technique is the most appropriate working standard used to represent a collection of characters for both storing and transmitting [7]. It was developed in conjunction with the Universal Coded Character Set (UCS) standard and published as the Unicode Standard. The latest version of Unicode contains a repertoire of more than 128,000 characters covering 135 modern and historic scripts, as well as multiple symbol sets, many encoding types like ANSI, UTF-8, UTF-16, UTF-13 etc. are available [8].

Unicode is designed to represent almost all the characters in every language in the world. All the characters of Arabic language are now encoded as per the Universal Principle of Unicode [9]. 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 industry character sets. Unicode techniques occupy more space in memory during storage. The recent Unicode version for Arabic Unicode character set [11].

The most popular compression programs that compress files like WinZip and WinRar were developed to reduce the storage space. However, they are mostly dealing with European languages. In addition, many other formats are widely used for compressing individual files, but they are not supporting Arabic languages efficiently. The major problem will be in performing decompression process where characters are wrongly decoded which gives a meaningless output. Many updates and improvements were considered for enhancement. For example, text compression for other eastern languages was carried out with the dictionary approach and bit replacement reduction technique [3].

6. The Proposed Arabic Encoding Technique

The proposed system involves substituting an ASCII character in place of a Unicode Arabic character, since the size of an ASCII character is one byte (8 bits) whereas a Unicode character size range between 1 byte (8 bits) to 4 bytes (32 bits) depends on the encoding technique.
To store the file with .txt extension, the following encoding types are available, they are ANSI encoding, UTF encoding, Unicode and Unicode big endian encoding. The bits required to store each character for the above encoding file type are 8 bits, 8 to 32 bits, 16 bits and 16 to 32 bits respectively [14].

Table 2. Arabic Compression Results

The Arabic Text File (.txt)	The Original Arabic File Size (KB)	Compressing the Original File by Encoding (KB)	Compressing the Original File by 7-Zip (KB)	Compressing the Original File by Encoding and 7-Zip (KB) - Multistage	The Enhancement Ratio of Compressing the Original File by 7-Zip to Multistage
File1.txt	1.41	0.806	0.773	0.668	15.71 %
File2.txt	6.99	3.910	2.320	2.050	13.17 %
File3.txt	8.66	4.83	1.023	0.888	15.20 %
File4.txt	18.3	10.0	1.300	1.140	14.03 %
File5.txt	197.0	109.0	2.640	2.340	12.82 %

7. Multistage Experimental Results Analysis

In these settings, more than one hundred Turkish and Arabic texts were tested. Random samples were selected, and the results are shown in the Table 1 & 2, which demonstrate the results for compressing by Encoding, compressing by 7-Zip, and compressing by Multistage (Encoding followed by 7-Zip).
The outcomes appeared in Table 1 demonstrate the enhancement ratio of compressing the original file by 7-Zip to the multistage
for different Turkish texts and sizes. The size of "File 1.txt" was reduced by 8.4% using encoding technique, and by 66.1% using 7-Zip, however, it was reduced by 66.7% applying the proposed Multistage technique (Encoding followed by 7-ZIP). Obviously, the results of the Multistage outperform the other methods individually, by 1.5% on average. Finally, we can see a little deviation in the results, actually, this depends on the Turkish text nature. And the following letters availability in the targeted text (Ç, Ş, Ğ, I, İ, Ö, Ü).
The outcomes appeared in Table 2 demonstrate the enhancement ratio of compressing the original file by 7-Zip to the multistage for different Arabic texts and sizes. The size of "File 1.txt" was reduced by 42.8% using encoding technique, and by 45.1% using 7-Zip, however, it was reduced by 52.6% applying the proposed Multistage technique (Encoding followed by 7-ZIP). Obviously, the results of the Multistage outperform the other methods individually, by 15% on average. Finally, we can see a little deviation in the results, actually, this depends on the Arabic text nature.
Obviously, the proposed Multistage compression results represents a significant file size reduction for all Arabic text files. It is highly recommended to embed the encoding technique into 7-Zip application.

8. CONCLUSION AND FUTURE WORK

Arabic and Turkish are classified as important languages that are spoken by two hundred million people in all over the world. There is a high need of storing the Turkish and Arabic documents in digital form. Many applications were developed for both computers and mobile phones. More powerful techniques are required to preserve literature, artistic and scientific work of mankind digitally. This lossless compression technique provides a beneficial storage for Arabic and Turkish documents. Almost the compressed document will be reduced to satisfactory levels. This method can be further enhanced by compressing the encoded files by the commercial applications. Furthermore, the compression can be built for other Unicode characters that exceeds the one Byte size. This is a notable achievement, and a necessary step toward universal literacy and universal storage reduction [16].
In conclusion, the integration of the Multistage compression technique with other available compression applications will significantly reduce the text file size. Finally, we tested the correlation between the documents size and the performance, and found it to be 0.71. The documents is very modest. It would be very interesting to demonstrate the performance of this technique once we deal with MBs or even GBs documents (read and compress the document in streaming mode) in the future work.

REFERENCES

[1] Tariq Abu Hilal1F & Hasan Abu Hilal. Arabic Text Lossless Compression by Characters Encoding. The 6th International Workshop on the Design and Performance of Networks on Chip (DPNoC 09), Canada, Halifax, August 2019.

[2] Martin J. Dürst. Character Encoding and Unicode WWW2005 Tutorial: Internationalizing Web Content and Web Technology. Department of Integrated Information Technology, College of Science and Engineering Aoyama Gakuin University. Tokyo: Sagamihara, 2005.

[3] Steve Atkin. Unicode Compression: Does Size Really Matter? TR CS-2002-11, IBM Globalization Centre of Competency. International Business Machines, Austin, Texas USA 78758, Ryan Stansifer, Department of Computer Sciences, Florida Institute of Technology. Melbourne: Florida USA 32901, July 2003.

[4] World Wide Web Consortium. Content first published 2008-01-31. Last substantive update 2015-04-16 16:47 GMT. This version 2019.

[5] Vijayalakshmi and N. Sasirekha. LOSSLESS TEXT COMPRESSION FOR UNICODE TAMIL DOCUMENTS B. Department of Computer Science, Vidyasagar College of Arts and Science. India; 2018

[6] Lishamol Philip and K.M. Abubeker, LiBek II. A Novel Compression Architecture using Adaptive Dictionary. Proceedings of International Conference on IEEE Emerging Technological Trends, pp. 212-218, 2016. https://doi.org/10.1109/ICETT.2016.7873674

[7] Shihjong Kuo. Processors, Methods, Systems, and Instructions to Transcode Variable Length Code Points of Unicode Characters. U.S. Patent, 2017.

[8] Alasmer , Z. M. , Zahran B. M., Ayyoub B. A., Kanan M. A. A Comparison between English and Arabic Text Compression. Journal of Contemporary Engineering Sciences; 2013. Vol. (6), No. (3), pp. (111-119). https://doi.org/10.12988/ces.2013.13010

[9] Sawalha, M. S. Open-source Resources and Standards for Arabic Word Structure Analysis: Fine Grained Morphological Analysis of Arabic Text Corpora, The University of Leeds; 2011.

[10] R. Seethalakshmi et.al. Optical Character Recognition for Printed Tamil Text using Unicode, Journal of Zhejiang University Science; Vol. 6, No. 11, pp. 1297-1305; 2005. https://doi.org/10.1631/jzus.2005.A1297

[11] Retrieved on 29/April/2019: http://www.unicode.org/charts/

[12] A. Carus and A. Mesut, 2010. Fast Text Compression Using Multiple Static Dictionaries. Information Technology Journal, 9: 1013-1021. https://doi.org/10.3923/itj.2010.1013.1021

[13] Tariq Abu Hilal , Hasan Abu Hilal, Turkish Text Compression via Characters Encoding. The 6th International Workshop on the Design and Performance of Networks on Chip, Belgium (DPNoC 2020). https://doi.org/10.1016/j.procs.2020.07.042

[14] Tariq Abu Hilal , Hasan Abu Hilal, Turkish Text Compression via Characters Encoding. The 6th International Workshop on the Design and Performance of Networks on Chip, Belgium (DPNoC 2020). https://doi.org/10.1016/j.procs.2020.07.042

[15] Arafat Awajan and Enas Abu Jrai. Hybrid Techniques for Arabic Text Compression. Global Journal of Computer Science and Technology; Vol. 15, No. 1, pp. 23-27; 2015.

[16] Haneen Ta'amneh & etl. “Compression-based arabic text classification”. 2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA), https://doi.org/10.1109/AICCSA.2014.7073253