A New Arabic Coding Scheme

Sarah Abdulkareem Al-Busaeed[1*], Umut İnan[2]

[1]Department of Engineering Management, Faculty of Engineering, Istanbul Gedik University, Istanbul, TURKEY
(*Corresponding author) sarah.obady@baghdadcollege.edu.iq

[2]Industrial Engineering Department, Istanbul Esenyurt University, Istanbul, Turkey
umutinan@esenyurt.edu.tr

Abstract: In this paper, we designed a new Arabic letter encoding scheme based on the characteristics of the Arabic language to solve many Arabic coding problems, especially those related to formulation problems. In the proposed coding scheme, we were able to represent the Arabic letter and its accent marks using one byte instead of two, thus, the size of the Arabic text was reduced in half. The suggested coding scheme can be used as a bilingual coding scheme instead of ASCII in an Arabic platform environment or as a text compression scheme.

Keywords: ASCII, Coding scheme, represents letters, compression scheme, and Arabic text compression.

1. INTRODUCTION

The most important difficulties faced by the developers of these technologies are the lack of technical academic research related to Arabic, which is why they simulate applications based on European languages with the representation of an Arabic craftsman. Although this simulation may succeed in solving one aspect of the Arabic language, it fails to build a complete processing system for it. The formation itself is one of the outstanding problems so far because the solutions used to process The Arabic language are linked to solutions based mainly on languages that are devoid of composition [1]. The current treatment of The Arabic language considers the movement to be an object in itself and can only be dealt with as part of the letters of the word, but dealing with the language without formation is a treatment that lacks perfection and is not without ambiguity in understanding and here it should be noted that the correct and complete treatment of the Arabic language must be delayed Considering the composition as a prerequisite in understanding the Arabic language, i.e. the introduction of Arabic texts or the construction of processes to form texts subjectively, such programs exist, but they are still inaccurate and cannot be relied upon in practical applications. Machine translation to and from The Arabic language is one of the most complex tasks that can be faced by those working in the field of developing Arabic language techniques and accessing practical solutions in this field is one of the very vital images of the user and Arab companies, especially the presence of the Internet and the huge content of information available in it. In multiple languages, some hasty solutions have emerged in the field of translation, but they have not succeeded and developed because they are not relying on in-depth research in linguistics, it is nothing more than an improved version of electronic dictionaries [2]. In this research, our study focused on finding an appropriate way to represent Arabic letters and their composition by returning to the basis of the Arabic language and its written letters in the past before drip and formation and we reached record results that serve the field of application of computer technologies and contribute to reducing the volume of Arabic information and data Stored and transferred as an introduction to trying to develop correct processing of Arabic language computer [3].

2. BACKGROUND THEATRICAL

2.1. Encoding

Encoding is that the way toward changing over information into an organization required for a few data preparing needs, including:

- Program assembling and execution
- Data transmission, stockpiling, and pressure/decompression
- Application information preparing, as an example, record change

Encoding can have two implications:

- In PC innovation, encoding is the way toward applying a specific code, for example, letters, images, and numbers, to information for transformation into a comparable figure.
- In gadgets, encoding alludes to simple to computerized

transformation [4].

Encoding incorporates the usage of a code to vary remarkable data into a structure that can be used by an outside technique. The sort of code used for changing over characters is known as the American Standard Code for Data Trade (ASCII), the most consistently used encoding plan for records that contain content. ASCII contains printable and nonprintable characters that address promoted and lowercase letters, pictures, diacritics checks, and numbers. An unprecedented number is given bent specific characters [5].

2.2. Unicode Encoding Model

There are a few frameworks used to coded data and dialects. Follows are the four degrees of the Unicode Character Encoding Model can be summed up as:

- ACR: Abstract Character Repertoire, the arrangement of characters to be encoded, for instance, some letters in order or image set
- CCS: Coded Character Set planning from a theoretical character collection to a lot of nonnegative numbers
- CEF: Character Encoding Form planning from a lot of nonnegative numbers that are components of a CCS to a lot of successions of specific code units of some predetermined width, for example, 32-cycle whole numbers
- CES: Character Encoding Scheme a reversible change from a lot of successions of code units (from at least one CEFs to a serialized arrangement of bytes). [6]

Notwithstanding the four individual levels, there are two other valuable ideas:

- CM: Character Map planning from arrangements of individuals from a theoretical character collection to serialized groupings of bytes crossing over every one of the four levels in a solitary activity
- TES: Transfer-Encoding Syntax a reversible change of encoded information, which could contain printed information

The IAB model, as characterized in [RFC 2130], recognizes three levels: Coded Character Set (CCS), Character Encoding Scheme (CES), and Transfer-Encoding Syntax (TES). Nonetheless, four levels should be characterized to sufficiently cover the qualifications required for the Unicode character encoding model. One of these, the Abstract Character Repertoire, is verifiable in the IAB model. The Unicode model likewise gives the TES a different status outside the model, while including an extra level between the CCS and the CES [7].

3. ARABIC STANDARD SPECIFICATIONS

In 1981, the Arab Organization for Standards and Standards, which was partnered with the Arab League and situated in Jordan, shaped an advisory group to decide the Arabic norm for Arabic letters in the field of data. A progression of these details was given from 1981, most as of late in November 1986, when the association endorsed the standard determinations (ASMO 708) for the trading of data on the PC in eight twofold numbers and recorded around the world under the number (ISO/8895-6). In this particular, it contains 120 characters, in addition to the eight balance devices, which are in progression: you plan to open, at that point you mean to add, at that point you expect to break, and afterward the opening, the join, the division, the force, the stillness [8]. The Arabic numbers in the (ASMO 708) English numerals involved similar English numbers. This determination is described by [9]:

- The presence of a location for each of the 28 characters.
- The presence of six sites of al-Hamza in different forms أ إ ؤ ئ آ
- Having a tethered t-site and another for a thousand cabins.
- The presence of eight formation sites isًٌٍَُِّْ in the order shown.
- The existence of the hiving in the ASCII table position between the connected letters.

As this standard is over 30 years late for the presentation of Arabic in PCs, PC makers have built up their particulars, making it hard for them to fix them in light of the significant expenses. Along these lines, a few arrangements of character frameworks kept on existing, contingent upon the PC producers. In as per the standard 708, a few Arab bunches were given, incorporating Arab Window Collections in Bahrain, Sakher in Saudi Arabia, IBM, and Microsoft Americas. The last gathering started to spread more than others after the oppression of windows and it is expected that it will stay prevailing, overpowering, and dropped in all or most different structures. It is noted among the restriction frameworks that the topic of the request for letters in order relies upon the gathering utilized, since every one of these sums has believed the synthesis to be letters, so the request for letters at the composing is viewed as the creation of a letter to be considered in the request. It brings about the gathering itself deciding the request technique. On the off chance that the letters are dissipated in the gathering table, the request will be dangerous, particularly if the character is taken as the letter as per its area in the table, as in the gathering number 864 for I.B.M. The areas of the uncommon letters (ء, ة, ى, –) all fluctuate from gathering to gathering, creating an alternate request when utilizing one and changing to the next [9].

4. REPRESENTATION OF ARABIC LETTERS

Before utilizing accentuations, the Arabic letter set contains 15 letters, four of them having single articulate, nine having two articulates, and 2 having three articulates for each the complete is 28 in the wake of utilizing diacritics to utilize each character with solitary elocution.

Four pieces are expected to speak to 16 letters (15 unique letters in addition to space). To achieve this, we separated the letters into two gatherings: the first letters and the inferred letters by adding accentuation to the first letters (and the assignment dependent on the strategy for drawing the letter) as appeared in Table No. 1.

Table 1. The original and derived letters

Original	Derived	Original	Derived	Original	Derived
الألف		ر	ز	ف	ق
ب	ي	س	ش	ل	ك
ن	ت، ث	ص	ض	م	
ح	ج، خ	ط	ظ	هـ	
د	ذ	ع	غ	و	

From Table 1, it very well may be noticed that there are letters that don't have a subsidiary and letters with more than one subordinate, for example, 'هـ' and 'ن'. To make the subordinate is just one for "ح" and "ن", the second subsidiary of "ن" is to be moved, and the second subsidiary for "ح", which is "خ" moved to the area in the table that relates to the first letters with no subsidiary. The quantity of passages in Table 1 is fifteen sections and it is very adequate to speak to the first letters utilizing four pairs, however, there is a letter that must be remembered for the portrayal on account of its essence in the content and it is a void and it needs to discharge one of the passages of Table No. 3.2 for example move one of the first letters and make it a subordinate of another unique letter This case is relative exactly "ء", which can be viewed as gotten from the letter "أ", and there is the letter "ة", included as inferred for the letter "هـ" and the outcome can be appeared in Figure No. 2.

Table 2. Original and Derivative Arabic Letters

Original	Derivative	Original	Derivative	Original	Derivative
ا	ء	ر	ز	ف	ق
ب	ي	س	ش	ل	ك
ن	ت	ص	ض	م	ث
ح	ج	ط	ظ	هـ	ة
د	ذ	ع	غ	و	خ

Table 3. Quartet representation of original

Char.	Rep.	Char.	Rep.	Char.	Rep.	Char.	Rep.
ا	0000	د	0100	ط	1000	م	1100
ب	0001	ر	0101	ع	1001	هـ	1101
ن	0010	س	0110	ف	1010	و	1110
ح	0011	ص	0111	ل	1011	Space	1111

The original letters in Table 3 can be represented using four bits as shown in Table 4.

Table 4. Quintet representation of the letters of the Arabic language

B_0							
1	0	Line		B_1	B_2	B_3	B_4
ء	ا	16	0	0	0	0	0
ي	ب	17	1	0	0	0	1
ت	ن	18	2	0	0	1	0
ج	ح	19	3	0	0	1	1
ذ	د	20	4	0	1	0	0
ز	ر	21	5	0	1	0	1
ش	س	22	6	0	1	1	0
ض	ص	23	7	0	1	1	1
ظ	ط	24	8	0	0	0	0
غ	ع	25	9	1	0	0	1
ق	ف	26	10	1	0	1	0
ك	ل	27	11	1	0	1	1
ث	م	28	12	1	1	0	0
ة	هـ	29	13	1	1	0	1
خ	و	30	14	1	1	1	0
chng	space	31	15	1	1	1	1

To plan a right portrayal for non-diacritic Arabic letters, a fifth paired piece is included with an estimation of (0) if the letter is unique and (1) if the letter is a subordinate, for instance (00011) speaks to the character "**ح**" and (10011) speaks to the character "**ج**" and (01101) speaks precisely **"هـ"** (**11101**) speaks exactly "**ة**", etc for the remainder of the letters in Table 4. For operational necessities, the image (11111) is viewed as the image for changing to ASCII portrayals (it will be clarified later), and space is spoken to

by (01111). The quintet is an adequate portrayal for all Arabic letters as appeared in Figure 5. Arabic letters contain two gatherings: the associated and the isolated letters as per the detailing and the spelling decides that rely now and then upon the diacritics character (will be clarified later), the supervisor program will draw the letters on the screen or the paper with its appropriate structure contingent upon the definition and spelling rules. At this point, non-diacritic Arabic letters are spoken to by five pieces rather than the eight-bit portrayal presently being used, accordingly, three pieces are decreased [10][8].

5. REPRESENTATION OF THE DIACRITICS CHARACTERS

To speak to the diacritics in the Arabic language, we have seen that the diacritics incorporate Sokon, Fatha, Dhama, Kasrah, and Tanween each of the past, which are four diacritics notwithstanding the Tanween and Shaddah, and we isolated the arrangement into two unique and inferred gatherings, given that the Tanween of every diacritic as appeared in Table 5.

Table 5. The original and derived from movements

Original	Diacritics shape	Derivative
Sokon	**Non-diacritic**	◌ْ
Fatha	◌َ	◌ً
Dhema	◌ُ	◌ٌ
Kasrah	◌ِ	◌ٍ

In Table 3.6 we put the Sokon as unique diacritics with no subsidiary. In the Arabic language, there are letters without diacritics characters that are written in words to show the possibility of more than one potential diacritic character, so we have to start a no-diacritics character in the table which must be a default diacritics character. To speak to such a case, we will consider that Sokon is diacritics gotten from "no-diacritics ". Similarly, as in the past passage, we will add three pieces to speak to the diacritics character as appeared in Table 6.

Table 6. Representation of diacritics characters

Diacritic	Representation	Diacritic	Representation
No diacritics	**000**	**Sokon**	**100**
Fatha	**001**	**Tanween Alfatha**	**101**
Dhema	**010**	**Tanween Aldham**	**110**
Kasrah	**011**	**Tanween Alkasar**	**111**

In Arabic letters, there is another diacritics character called "shadda" (ّ) utilized if a word contains a rehashed character therefore, furnished that the primary letter is with Sokon diacritic, shadda is wiped out in the proposed portrayal.

Here it ought to be noticed that the letter (ا) is without a diacritic and the letter (آ) is a similar letter with Sokon diacritic and the letter (آ) can be composed as twofold "ا" without a diacritic, while 'ى' is composed 'ا' with "ٰ diacritic. The letter "ء" has four different structures composed by the spelling decides that rely upon diacritics character go before or followed, they are (أ, إ, ؤ, ئـ). In this way, we had the option to speak to the Arabic letter and its diacritics utilizing one byte rather than two, and in like manner, the size of the Arabic writings was diminished into half. Table 6, shows the portrayals of the test of two distinct letters followed by every single imaginable diacritic.

6. ARABIC LETTERING PROGRAM

From the foregoing, it is clear that we have allocated a storage value for each of the Arabic letters, and the truth is that the letters stored in the computer storage do not resemble the letters on the screen or on the printer, and the truth of this difference is from the way letters are written within Arabic texts because most Arabic letters are not written in one form within the words but rather most letters have more forms that differ according to their position in the word.

Table 7. "ص" and "ك", with diacritic representations

Character	8-bit coding	Decimal value	Note
ص	00000111	7	Without diacritics
صَ	00100111	39	7+32
صُ	01000111	71	7+64
صِ	01100111	103	7+96
صْ	10000111	135	7+128
صً	10100111	167	39+128
صٌ	11000111	199	71+128
صٍ	11100111	231	103+128
ك	00011011	27	Without diacritics
كَ	00111011	59	27+32
كُ	01011011	91	27+64
كِ	01111011	123	27+96
كْ	10011011	155	27+128
كً	10111011	187	59+128
كٌ	11011011	219	91+128
كٍ	11111011	251	123+128

In general, each letter has three forms, which are: at the beginning of the word (frontal), at the end of the word (final), and in the middle of the word (middle). In addition, the letters of the Arabic language are divided into three groups as shown in table 8, they are:

1. 8 letters have the same form in the three positions they are: (ا,د,ذ,ر,ز,و,ط,ظ.)
2. 17 letters have the same form in frontal and middle and differ from the final position; they are (س, ش, ص, ض, ب, ت, ث, ن, ي, ح, ج, خ, ف, ق, ك, ل, م).
3. 3 letters have different form in each position they are (ع غ, هـ).

The first group is considered as the separated letters, while the second and the third groups are the connected letters which written connected to the previous or next letter or connected with both.

Table 8. Forms of Arabic letters

Arabic character	frontal	middle	final
ا,د,ذ,ر,ز,و,ط,ظ	ا,د, ذ, ر, ز,و,ط,ظ	ا,د, ذ, ر, ز,و,ط,ظ	ا,د, ذ, ر, ز,و,ط,ظ
ب, ت, ث, ن, ي, ح, ج, خ س, ش, ص, ض ف, ق, ك, ل, م	بـ,تـ, ثـ, نـ, يـ, حـ, جـ, خـ سـ, شـ, صـ, ضـ فـ, قـ, كـ, لـ,مـ	بـ, تـ,ثـ, نـ,يـ, حـ, جـ, خـ سـ, شـ, صـ, ضـ فـ, قـ, كـ, لـ,مـ	ب, ت, ث, ن, ي, ح, ج, خ س, ش, ص, ض ف, ق, ك, ل, م
ع, غ, هـ	عـ, غـ, هـ	عـ, غـ, هـ	ع,غ or غ ه or هـ

8. CHNG CODE

We have agreed according to the above to name the letter which represents (11111) as the change code (chng) as it appears in Table No. 8. The Arabic text maybe contains any other characters or symbols or numbers which are not an Arabic letters or diacritics and can be represented by ASCII table provided that each must be preceded by chng code.

9. GENERAL STRUCTURE OF THE CODING AND ENCODING ALGORITHM

The fundamental structure of the coding calculation is portrayed in Figures 1, 2 which clarify the flowchart of the proposed coding and encoding calculation individually. The application utilizes the Arabic letter set as a string called letter set = 'ابنحدرسصطعفلمهو ءيتجذزشضظغقكثةخ', which is requested in exceptional succession proposed beforehand in this proposition. The Arabic language is correctly supported; hence, the most right letter is in position 1, etc the last letter is the most left character. The diacritics characters are put away in the string variable called diacritic="ً ٌ ٍ َ ُ ِ ّ ْ" in the coding program. The calculation gives two choices: coding and encoding. The coding method chip away at blending the Arabic letter with its diacritic code in a solitary portrayal code, while the encoding methodology is used to isolate them for composing. Python 3.8.5 is utilized to construct the coding and encoding calculations.

Algorithm 1, Algorithm 2 depicted the essential steps to perform the coding and encoding algorithms.

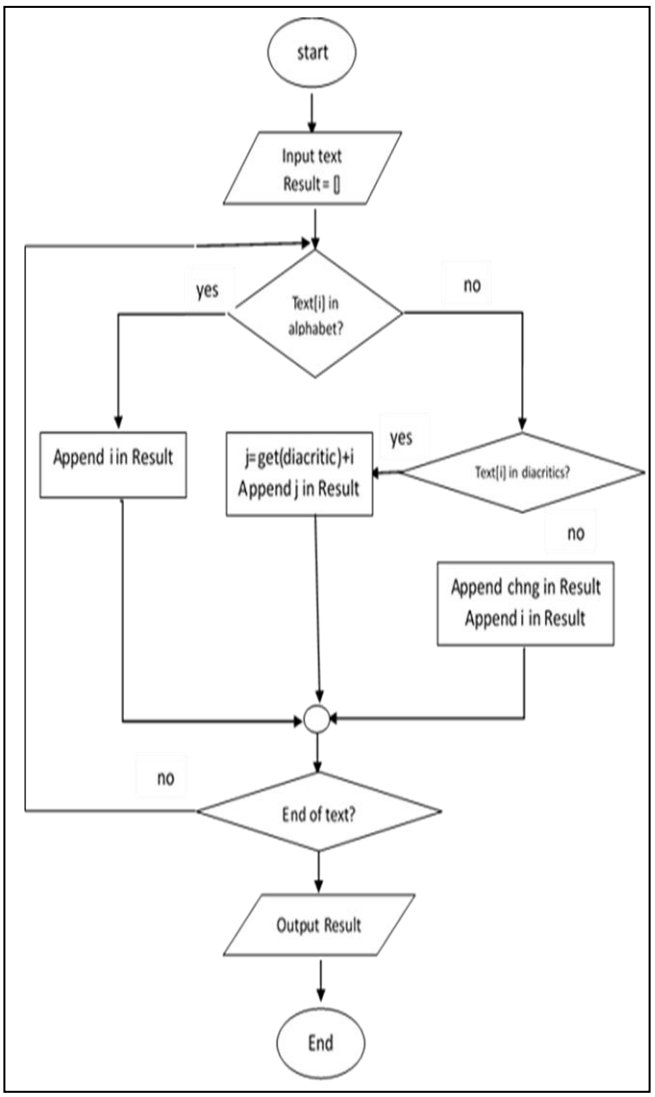

Figure 1. Flow chart of the Coding algorithm

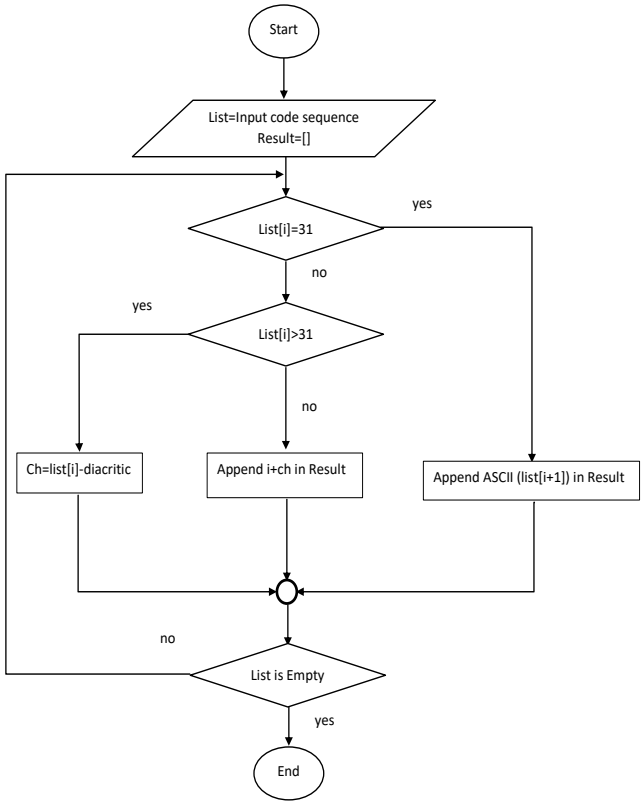

Figure 2: flow chart of the Encoding algorithm

10. EXPERIMENTAL RESULTS ANALYSIS

The principal objective of planning and building Arabic letters encoding plan is to utilize a strategy gotten from understanding the letter and diacritics in the Arabic language to locate a reasonable portrayal of the Arabic letter and its diacritic, and we have come to make a portrayal of the character and its diacritics in the size of one byte, while they were spoken to by the size of two bytes. In this manner, we got a pressure proportion identical to 50 percent. While, if Arabic writings were utilized without diacritics, the content would not be compacted. Table No. 9 demonstrating the pressure proportion for four records with a similar book, the main document has a 100% diacritical extent and the subsequent record contains a similar book, however at a proportion of 60% of diacritics the third record contains similar content at a proportion of 30% of diacritics and the fourth document contains similar content without diacritics.

Table 9. Encoding Results

Arabic Text File	Original File Size (KB)	Encoded File Size (KB)	Compression Percentage
File1.txt	10.0KB	3.4KB	66%
File2.txt	8.28KB	3.3KB	61%
File3.txt	5.69KB	3.2KB	44%

11. CONCLUSION

In this article, we have intended to develop a new compact and basic encoding system to represent the Arabic letters as an alternative to the ASCII system in an Arab operating environment, which is an important and basic part of the line of embarking on building Arabic operating environments (systems) and its application needs to alter the currently used keyboard or design a new model It fits with the new acting. Without that, the new representation system could be used in other applications such as a parsing system or a text compression system.

One of the most contributions of this article is to build a representation based on a deep understanding of the Arabic language and not as it was in the past, where the Arabic letters were represented in a similar way to represent the Latin language.

The second contribution of this article is in the field of data compression, we creating a compressed representation of at least 66%.

ACKNOWLEDGMENT

I would like to express my special thanks to my first teacher (**Assistant Professor Dr. AbdulKareem Ibadi**) who gave me the golden opportunity to do this wonderful project on the topic (**A new Arabic Coding Scheme**) as well as to my supervisor (**Assistant Professor Dr. Umut Inan**), who also helped me in conducting this research and forward I prepared it with advice and directions and introduced me to many new things for which I am grateful.

REFERENCES

[1] W. Helali, Z. Hajaiej, and A. Cherif, "Arabic corpus implementation: Application to speech recognition," 2018 Int. Conf. Adv. Syst. Electr. Technol. IC_ASET 2018, pp. 50–53, 2018, DOI: 10.1109/ASET.2018.8379833.

[2] M. Johnson et al., "Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation," Trans. Assoc. Comput. Linguist., vol. 5, pp. 339–351, 2017, DOI: 10.1162/tacl_a_00065.

[3] A. Awajan, "Multilayer model for Arabic text compression," Int. Arab J. Inf. Technol., vol. 8, no. 2, pp. 188–196, 2011.

[4] [4] R. Ayadi, M. Maraoui, and M. Zrigui, "A Survey of Arabic Text Representation and Classification Methods," Res. Comput. Sci., vol. 117, no. 1, pp. 51–62, 2016, DOI: 10.13053/RCS-117-1-4.

[5] M. Mehrnoush, B. J. Belzer, K. Sivakumar, and R. Wood, "EXIT Chart-Based IRA Code Design for TDMR Turbo-Equalization System," IEEE Trans. Commun., vol. 65, no. 4, pp. 1762–1774, 2017, DOI: 10.1109/TCOMM.2017.2662003.

[6] P. Update, "Unicode character encoding model," pp. 1–23, 2008.

[7] T. A. Hilal and H. A. Hilal, "Arabic text lossless compression by characters encoding," Procedia Comput. Sci., vol. 155, no. 2018, pp. 618–623, 2019, DOI: 10.1016/j.procs.2019.08.087.

[8] A. Ibady, "A new localization and compression system Dr. Abdulkareem Ibady Baghdad college of economic science 2010," 2010.

[9] S. K. Mukhopadhyay, M. O. Ahmad, and M. N. S. Swamy, "SVD and ASCII Character Encoding-Based Compression of Multiple

Biosignals for Remote Healthcare Systems," IEEE Trans. Biomed. Circuits Syst., vol. 12, no. 1, pp. 137–150, 2018, DOI: 10.1109/TBCAS.2017.2760298.

[10] S. S. Ismail, I. F. Moawad, and M. Aref, “Arabic text representation using rich semantic graph: A case study,” Recent Adv. Inf. Sci., pp. 148–153, 2013.