How to cite this paper:

Ashraf Al_Omoush, Norita Md Norwawi, Roesnita Ismail, Fauziah Abdul Wahid, & Ahmad Akmaluddin Mazlan. (2017). Storage optimization for digital quran using sparse matrix with hexadecimal representation in Zulikha, J. & N. H. Zakaria (Eds.), Proceedings of the 6th International Conference of Computing & Informatics (pp 167-174). Sintok: School of Computing.

STORAGE OPTIMIZATION FOR DIGITAL QURAN USING SPARSE MATRIX WITH HEXADECIMAL REPRESENTATION

Ashraf Al_Omoush[1], *Norita Md Norwawi[2] , Roesnita Ismail[3], Fauziah Abdul Wahid[4], and Ahmad Akmaluddin Mazlan[5]

[1]*Universiti Sains Islam Malaysia*, omoush23@yahoo
[2,3,4,5] Universiti Sains Islam Malaysia, Malaysia, {norita, roesnita, fauziah, aamja03}@usim.edu.my

ABSTRACT. Digital Quran is a common application today in smart devices such as hand phones and tablet. The verses are usually presented using images of verses on written in an Arabic font. This raise issues related to storage when representing the whole Quran text. This study proposed a hexadecimal digital representation technique for words in Arabic using UTF-8 for character encoding which is backward compatible with ASCII code. This paper will explain three approaches used in representing the digital Quran. They are the hexadecimal representation for each Quranic words, sparse matrix in representing verses of Al-Quran and double offset indexing for transforming the sparse matrix to efficiently use the space. This representation proof more than 50% reduction in the memory space for storage thus will increase the searching speed. The storage of words being optimized through the use of one memory space for that particular word rather than one memory space for each Arabic character in the words .This technique helps optimized memory management for the whole digital Quran. Due to the smaller size, the digital Quran can be conveniently installed in the future as standard applications in mobile devices.

Keywords: Digital Quran, Unicode, Mapping, Indexing, UTF-8.

INTRODUCTION

With the technology advancement, the digital Quran is a common and popular application nowadays in smart devices. In this paper, we present the use of sparse matrix for a new representation for Quranic verses using UTF-8 for character encoding formulation then inherit the Hexadecimal code of each verse by giving each word and verse a Unique ID (Uniqe.ID). Since verses of Al Quran has much duplication of words using this optimization to indexing of Al Quran words based UTF-8 character encoding yields more than 50% reduction in the memory space thus able to maximum searching speed.

Since the Holy Quran contains 114 chapters, 30 juzu, 6236 verses, 77439 words and 320015 letters. Due to this huge amount of words the proposed Digital Quran representation with sparse matrix will optimize the storage and safe the searching time. Thus the digital Quran can be used as a standard application in electronic devices such as smart phones, tablets, and other devices and help Muslims to retrieve and browse Al Quran faster and save storage on their devices.

This paper is organized as follows. In section 2, the review of literature related to our prosed Digital Quran and compare to other proposed techniques and other languages, in section 3, overview on UTF-8 character representation for holy Quran past and currently. The methods of constructing Digital Quran representation and algorithms based on hexadecimal and UTF-8 character encoding is presented in section 4, section 5 presents the results and findings. Finally, our work of this paper is summarized in the last section.

LITERATURE REVIEW

Study of literature have been done to the three languages to the word representation which are Chinese, Hindi and Arabic showing some techniques used to represent the words on each language. There are about 20,000 Chinese characters and 6,700 of them are considered to be the commonly used characters (Law & Chan, 1996). These characters form a web of compound words that are arbitrary in length. An interesting example is the Chinese words "海上" and "上海". The two characters used in the example are above (上) and sea (海). The word "海上" means above the sea while the word "上海" means Shanghai. The hidden Markov model HMM is extensively used in the word boundary discovering area (Law & Chan, 1996) & (Zhang et al., 2003). Most papers use Baum-Welch's Expectation Maximization (EM) approach to learn the probabilistic model and then apply some modified algorithm very similar to the Viterbi decoding algorithm to get the segmented text sequence.

Hindi language ranks third after English and Chinese as (Tripathi, 2012) mentioned on his paper that Unicode was function to retrieve from the web but due to a lake of understanding user's languages also lake of corpora and vastness of Hindi language in addition to limited patterns matching of literals cause a delay of develop a suitable algorithm for searching. (Sharma et al., 2012) proposed an algorithm to translate between English and Hindi language using different character encoding for target language in UTF and wx-notation by using phrase based statistical machine translation (SMT) technique shown that transliteration in wx-notation gives improved result over UTF-notation.

Recently Arabic language researches concentrate on modern standard Arabic; though they did not focus on Quranic Arabic despite the importance of Al-Quran for all Islamic word, (AlMaayah et al., 2014) proposed a method for Quranic Arabic WordNet that can do preprocessing by tokenization, removal of stop word, stemming and POS tagging then synonym sets by grouping word of similar meaning and part of speech, see Figure 1.

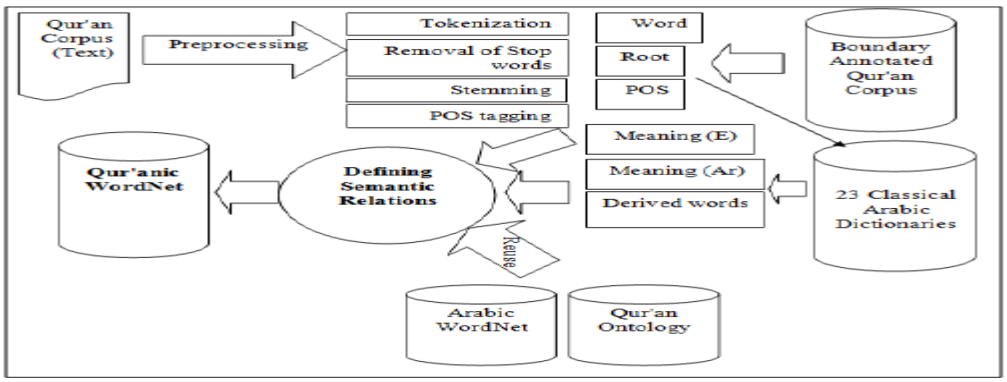

Figure 1. Quranic Arabic WordNet.

(Far & Jalil,2012) proposed an algorithm that compare each word from Arabic text which coded internally to a library by selecting the shortest code word possible then replace Unicode

representation which save space, the QR code known as Romanization cause Arabic character embedded in to QK code as 8-character instead of one character as shown in Table 1.

Table 1. Arabic character Representation Unicode.

Character	Representation in Unicode
ي	ى
ىو	وى

(Nazeeh, 2015) proposed an algorithm to segment Quran pages to text line images without changes by applying prepressing method which called binarization, a Quranic code for representing the holy Quran proposed by (Foda et al., 2013) which on character level, word level and phrase level and adding a new characters that have a symbol in Quran and don't have unicode for it; refer to Figure 2, in this model the authors use the presentation to represent chapter name, aya number and page number.

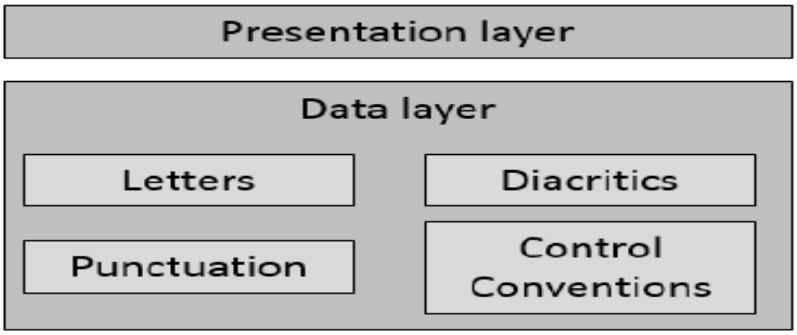

Figure 2. Qur'anic Model.

(Abdelhamid et al., 2013) and (El-Sakka, 2013) used the same criteria for Holy Quran text representation , the database was structured hierarchically as chapter, ID,…, page, index refer to Figure 3 .

```
<Chapters>
    <ID>
    <Arabic Name>
    <English Name>
    <Bismillah>
    <Verse Count>
    <Word Count>
    <Type>
    <Page No>
    <Index No>
</Chapters>
```

Figure 3. Definition of Holy Quranic Chapters.

OVERVIEW ON UTF-8 CHARACTER REPRESENTATION FOR HOLY QURAN

The Holy Quran is a corpus which consists of 30 juzu, 60 hizb, 114 chapters, 6236 verses, 77439 words and 320015 letters. Due to this huge amount, there's no such a method to classify all chapters thus make the searching and retrieving difficult due to Arabic language issues and challenges such as synonyms, hypernym, polysemous and semantic meaning for Arabic (Al_Omoush, 2015).

The results of the Unicode project, an effort to improve the architecture of multilingual text handling a method of encoding characters on computers that allows for efficient processing with a character set that covers all the world languages; Unicode transformation format (UTF) is the universal character code standard to represent characters, UTF-8 is an alternative coded representation form for all the characters in Unicode while maintaining compatibility with ASCII code (Gupta et al., 2010). Figure 4 show the Current UTF-8 Unicode Representation A&B of Arabic Letters which used in our formula for encoding of Arabic words in digital Quran For example the letter seen (س) equal to FEB1 in hexadecimal and the word ALLAH (ﷲ) is equal to FDF2. The idea of choosing UTF-8 character representation for Quranic words in Arabic which is backward compatible with ASCII code and hexadecimal representation which also integrated with our methods disused in next section.

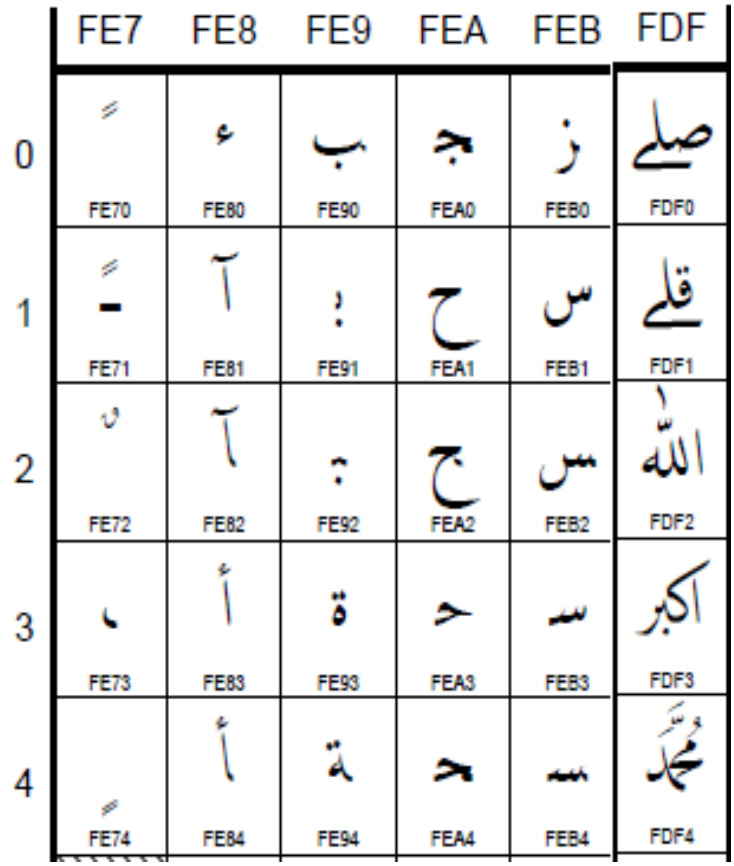

Figure 4. Unicode Standard 7.0, Copyright © 2014 Arabic Presentation Forms-A&B.

DIGITAL QURAN REPRESENTAION USING UTF 8 IN A SPARSE MATRIX

The first method was using UTF-8 character encoding formulation with The Unicode Standard as mention in above which is backward compatible with ASCII code to calculate the word in Arabic text representation will be in hexadecimal then calculate each verse using our formula , see Figure 5, after that compare the results which yields 64.41% of reduction fore Surat ALFATEHA (الفاتحة).

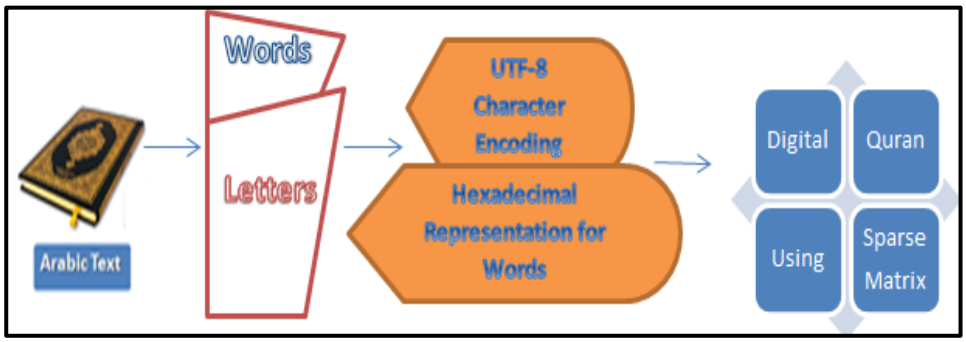

Figure 5. Unicode Hexadecimal Represent Methods for Quranic Words.

The main issue is to find an accurate way to classify or mapping all words of chapters of AL Quran in code criteria, existing source such as AL Quran words are mapped into array of letters then match each letter with its current place on the word either on the beginning, middle, isolate or last into hexadecimal Unicode for example the word (الرحيم) ALRAHIM will divide into six letters (ا لـ ر ح ـي ـم) then match each letter with its proper Unicode representation refer to Figure 6 which show the algorithm Used to decode Surat ALFATEHA (الفاتحة) in hexadecimal value which contains 7 versus the reduction will be 64.41%. Then calculate the word representation according to formula 1&2 followed by example of the word (الرحيم) ALRAHIM, hence the result of calculate the new hexadecimal representation was 5F893 which yield about 58.33% of reduction to the 148 times Occurrence in Quran.

Input: Quranic Words in Arabic
Output: Hexadecimal Code of Arabic Words

Start
Quranic Words ← W
Preprocessing Word into Letters,
$W = \{h_1, h_2, \ldots, h_n\}$
Check Look Up table for Hex. Cod
Letter Value in Hex. Cod
While (h_n != ∅)
Hex. Cod = Σ ($h_1, h_2, \ldots, h_n$)
End Loop
End

Figure 6. Words Conversion Algorithm in to Hexadecimal.

$$W(k) = L\ (h1, h2, \ldots hn) \quad (1)$$

$$k = \sum_i^n hi \quad (2)$$

Example of (الرحيم) ALRAHIM, (ا لـ ر حـ يـ م) according to the formula 1&2:

$$W(k) = L\ (h1, h2, h3, h4, h5, h6)$$

$$W(\text{الرحيم}) = L\ (\text{ا, لـ, ر, حـ, يـ, م})$$

$$k = \sum_1^6 hi$$

$$= \ 5F893$$

Look up Table created after calculate the word representation and trained to hold three fields; Uniqe.ID which refers to the position of the word to eliminate a repeated word which saves memory, the second field contains the Word and the last for Hex.Cod. The storage of words being optimized through the use of one memory space for that particular word rather than one memory space for each Arabic character in the words on Quran, after that Sparse Matrix take place which already contain the code ready to encode to the display of Arab characters Surat ALFATEHA (الفاتحة) refer to Figure 7&8, the reduction of the storage is around 81.19%.

```
public class DataSetCode {
    private HashMap <String, String> DataSet;

    public HashMap <String, String> insertData ()
    {
        DataSet = new HashMap<>();

        DataSet.put("3", "يمح الر نمح الر ه الل مهب");
        DataSet.put("4", "دمح ال");
        DataSet.put("2", "دلل ");
        DataSet.put("5", "بر");
        DataSet.put("6", "ينم الع ال");

        DataSet.put("7", "نمح الر");
        DataSet.put("8", "يمح الر ");
```

Figure 7. Screen Shots Represent Sparse Matrix.

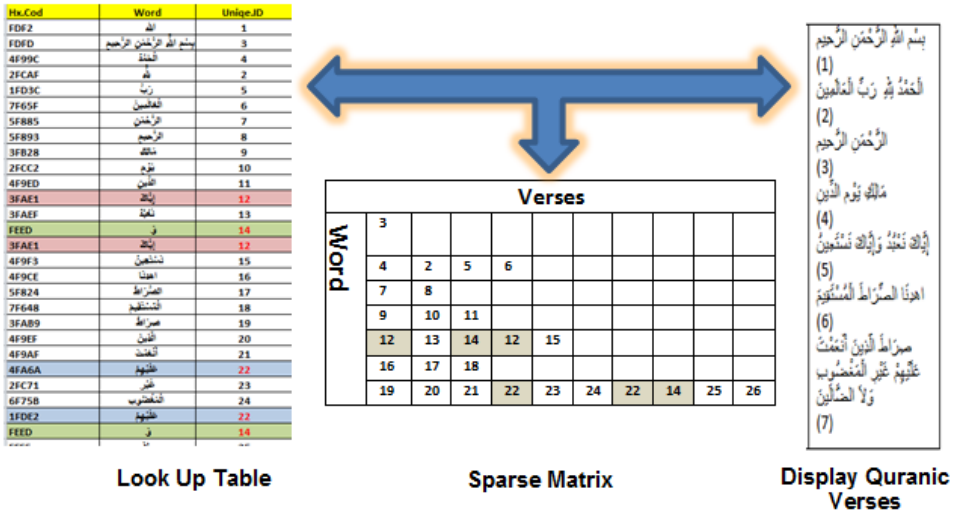

Figure 8. Flow Chart Represent Data Flow.

EXPERIMENTS AND FINDINGS

We found the proposed hexadecimal digital representation technique for words in Arabic using UTF-8 for character encoding is service as a space saving using the algorithm and formula 1&2 giving an example of (الرحيم) the word Occurrence in Quran is 148 times and the size of normal Arab characters is 12 bytes after applying our algorithm size reduced to 5 bytes which totally save about 58.33% of memory refer to Figure 9.

Criteria	Current presentation	Unicode presentation
Representation	الرحيم	5F893
Size(bytes)	12	5
Frequency in Quran	148	148
Total Size	1776	740

Figure 9. Example of (الرحيم) Mentioned in Quran 148 Times the Reduction is 58.33%.

The second result while comparing the size of input file and output file after applying the representation for Quranic verses using UTF-8 for character encoding for decode Surat ALFATEHA (الفاتحة) which contains 7 versus the reduction will be 64.41%. The third result using sparse matrix by generate a unique identification for each word in digital Quran (Uniqe.ID) which refer to the position of the word to eliminate repeated or duplicated words and swap the position by comparing the input file contains the unique code which equal to 104 bytes and the output file contains Quranic words is equal to 553 bytes which saves memory up to 81.19% for Surat ALFATEHA which excellent reduction of storage.

CONCLUSION

In this study we propose a new representation for digital Quranic using character encoding which is a hexadecimal digital representation technique using UTF-8 Arabic character encoding which is backward compatible with ASCII code. A sparse matrix was used to represent each verse of the Quran with double offset indexing method to efficiently use the matrix

space. The storage of words being optimized more than 50% through the use of one memory space for that particular word rather than one memory space for each Arabic character in the words on Quran. 81% improvement of space by using sparse matrix for representing verses in the surah. This technique helps optimized memory management for the digital Quran as a whole. One verse can be represent as one Uniqe.ID as a future work for this research which give more improvement.

ACKNOWLEDGMENTS

The authors gratefully acknowledge funding under research grant reference number (USiM-NRGS-P/FST/8404/52113) provided by the Research Management Centre, University Science Islamic Malaysia (USiM).

REFERENCE

Abdelhamid, Y., Mahmoud, M., & El-Sakka, T. M. (2013, December). Using Ontology for Associating Web Multimedia Resources With the Holy Quran. InAdvances in Information Technology for the Holy Quran and Its Sciences (32519), *Taibah University International Conference* on (pp. 246-251). IEEE.

AlMaayah, M., Sawalha, M., &Abushariah, M. (May 2014). A proposed model for Quranic Arabic Word Net.In *Proceedings of the 2nd Workshop on Language Resources and Evaluation for Religious Texts*, 31, Reykjavik, Iceland (pp. 9-13).LRA.

Al_Omoush, A., Norwawi, N. M., Ismail, R., Wahid, F., & Mazlan, A. A. (December 2015). Unicode Hexadecimal Representation for Digital Quranic Words.*FEIIC-International Conference on Engineering Education and Research*, Madinah, Kingdom of Saudi Arabia, 19-21

El-Sakka, T. M. (2013, December). Real-Time Interactive Verification of Quran Words in The Web Contents. In Advances in Information Technology for the Holy Quran and Its Sciences (32519), 2013 *Taibah University International Conference* on (pp. 12-17). IEEE.

Kamil, M. F. I., & Jalil, K. A. (2012, October). The embedding of Arabic characters in QR code. In Open Systems (ICOS), *2012 IEEE Conference* on (pp. 1-5). IEEE.

Foda, K. M., Fahmy, A., Shehata, K., & Saleh, H. (2013, December). A Qur'anic Code for Representing the Holy Qur'an (Rasm Al-'Uthmani). InAdvances in Information Technology for the Holy Quran and Its Sciences (32519), *2013 Taibah University International Conference* on (pp. 304-309). IEEE.

Gupta, R., Goyal, P., & Diwakar, S. (2010, September). Transliteration among Indian Languages using wx Notation, In *KONV ENS Germany* (pp. 147-150).

Law, H.C., Chan, C. (1996). "N-th Order Ergodic Multigram HMM for Modeling of Languages without Marked Word Boundaries," COLING Volume 1: *The 16th International Conference* on Computational Linguistics.

Nazeeh, L., Bany, J. (2015). Illumination Removal and Text Segmentation For Al-Quran Using Binary Representation, Faculty of Information and Communication Technology.Utem

Sharma, S., Bora, N., Halder, M., & Phrasal, S. (2012). *English-Hindi Transliteration using Statistical Machine Translation in different Notation*, Iccce, 12–14.

Tripathi, A. (2013). Problems and prospects of Hindi language search and text processing. *Annals of Library and Information Studies (ALIS)*, 59(4), 219-222.

Zhang, H.P., Liu, Q., Cheng, X.Q., Zhang, H., Yu, H.K. (July, 2003), "Chinese Lexical Analysis using Hidden Markov model" Second SIGHAN workshop affiliated with 41th ACL; *Sapporo Japan*, pp. 63-70.219–222.