ISSN 2347 - 3983
Volume 8. No. 1.1, 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter0981.12020.pdf
https://doi.org/10.30534/ijeter/2020/0981.12020

Sequential and Recursive Structure Searching Algorithms for Arabic and English Texts

Rostam Sadiqi, Mohd Zainuri Saringat, Aida Mustapha, Salama A Mostafa
Faculty of Computer Science and Information Technology, UniversitiTun Hussein Onn Malaysia, Parit Raja, 86400 BatuPahat, Johor, Malaysia
rostamsadiqi10@gmail.com, {zainuri, aidam, salama}@uthm.edu.my

ABSTRACT

The Holy Quran is a central religious book of Islam that is followed by all Muslims throughout the world. Recently, digital Quran indexing has helped Muslims in the entire world to search for specific content and index in the Quran. There are many methods of the searching amount of data to find one particular piece of information such as finding a specific word or a sentence in the Holy Quran. The search may proceed consecutively in every element until the query is found or recursively traverse the tree data structure but yet in a linear manner. This study is set to compare the performance of two searching techniques—sequential search as well as B+Tree search algorithms. A comparison analysis is conducted using two languages, which is Arabic from the Quran verses along with the English translation of each verse. The experimental result revealed that after the average times for each searching algorithm has been recorded, the sequential search required a lot of time during the search process, while B+Tree produced a significantly low retrieving time, which were 34ms and 1ms respectively. This can be concluded that B+Tree performed faster searching than sequential methods. In the future, it is hoped that more searching algorithms are open for testing with this dataset such using linked list.

Key words :B+Tree Search, Sequential Search, Quran Search Engine.

1. INTRODUCTION

Searching is a process of finding a piece of data from given datasets [1]. Given a list of data, sequential search or linear search is the most natural method where the algorithm consecutively checks for every element in the list until the query is found. On the other hand, a recursive structure searching is a recursive implementation of linear search in a tree, whereby the tree is recursively traversed from top to bottom starting from the root. B+Tree index structure has been generally employed in database management [2]. B+Tree is a variant of B-Tree that first sorts keys for sequential traversing but keeps the index balanced with a recursive algorithm. B+Tree stores copies of the keys in the internal nodes; the keys and records in the leaves; as well as a pointer to the next leaf node to speed up the sequential or linear search [3]. The rebalancing operations are also different for B+ trees because every rotation is different due to copy of the key in the parent.

A significant amount of literature has been published on the searching algorithms. While looking into a large and growing body of literature, it appears that searching algorithm has been proven to be successful in searching large dataset. The work in [4] compared three searching algorithms, which were sequential, binary and hashing algorithm on data structure with different problems based on time complexity and space complexity. They concluded that the sequential search is best used for small data while binary and hashing for large data. The experiments also showed that the hashing algorithm was faster than the other algorithms. In more recent work, [5] presented a search engine that was designed to crawl and index large datasets efficiently and produce relevant results for the user query. The index is the pointer to the documents stored in the actual database and designed with the purpose of finding relevant results in optimized time. The research concluded that the performance of a search engine depends on indexing schema for fast retrieval of results.

However, search queries in Arabic are different from English because the Arabic language consists of 25 consonants that are written from right to left. The morphology changes shapes according to their position in the word. In addition to three long vowels, Arabic has short vowels that are called diacritics, which are written above or under a consonant to associate the word with a particular sound with a particular meaning [6, 7]. In the last few years, the Quran has become a target of interest for the researcher in the field of Computer Science, for exploring the divine knowledge encapsulated in it. Processing Arabic text is also more sensitive, hence in need of extra natural language processing steps to deal with the diacritical scripts. For example, different number of dots at a different position (above or below a word) may change the meaning of the whole verse. This scenario does not happen in English, where a character does not affect the meaning of the word [8].

According to [9], on top of its glaring uniqueness, as compared to the English text, processing Quranic text is also different from processing regular Arabic corpora extracted from newspapers or speeches. Due to this uniqueness,

extracting topic from the holy Quran using the generative model revealed that a topic in the Quran can begin in one verse and the pattern sporadically changes in other verses. In addition, one verse may also well contain many topics, hence the importance of understanding the effect of keywords based on the total retrieve and relevant query results in an Information Retrieval (IR) process [10]. Besides, Quranic text has also been a subject to other research types such as in ontology processing and cross-language Machine Translation (MT). Back in 2012, [11] proposed a formal method for processing natural language from the Quran with a specific notation to express formal specification of the Arabic language. Three search techniques were investigated, which were text-based, stem-based and synonyms-based. The research also applied light stemming algorithm to find the stem word and analyze the formal specification.

In 2014, [12] presented a Quran-related search based on cross-language MT, which enables users to search a body of text in one language but using another language. For example, the Quranic text is in Arabic but the query term is written in English. Such ability is made possible with the availability of the English translation in the Quran. [13] explored searching methods in ontology processing based on given keywords to retrieve the knowledge of the keyword. The proposed searching method was evaluated using the recall and precision measurements and the experiments returned high accuracy. Other research related to the Arabic language includes hadith authentication [14], hadith ontology [15], and tweets [16].

This paper is set to compare the performance of two searching algorithms, which are sequential search and B+Tree search on both Arabic and English text in the Quran domain. The Quran is a sacred scripture of Islam that is believed to contain the commandment of the word of Allah, hence providing instructions and guidance to humankind in achieving happiness in life in the world and hereafter. As the Holy Quran contains rich knowledge and scientific facts, humans have difficulty in understanding the Quran content. The importance of the Holy Quran in Muslim daily life and references is unquestionable.

The Quran is written in Arabic but has been widely translated and transliterated into many languages across the world. It consists of 114 chapters (surah) and 6,236 verses (ayah). The 114 chapters, the Quran is further divided into 7 stages (manzil) and 30 parts (juz). Al-Baqarah (The Cow) is the longest chapter in the Quran with 286 verses while Al-Kauthar (The River of Abundance) is the shortest chapter with only 3 verses. The dataset understudy will cover all the 6,236 Arabic verses in the Quran along with the equal number of translations in English.

The remaining of this paper proceeds as follows. Section 2 presents the materials and methods to achieve the paper's objective, Section 3 presents the comparative results, and finally, Section 4 concludes the paper with future work.

2. MATERIALS AND METHODS

This research focuses on two types of searching techniques; sequential vs. recursive structure search and compares the searching performance when applied on both English and Arabic texts [17], [18]. The proposed research involved three steps, namely data resource, calculating the complexity and comparative analysis as shown in Figure 1.

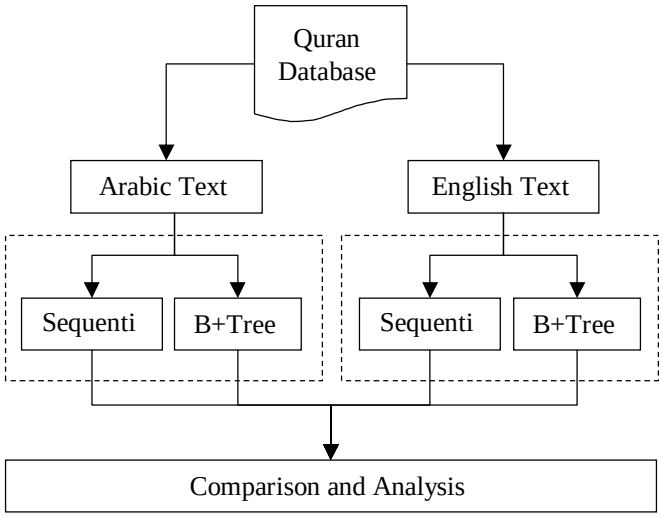

Figure1: Research Steps

The first step is preparing data, which consists of two stages (1) sourcing the data and (2) creating a database. The second step is calculating the complexity of these two search algorithms namely B+Tree and sequential. The third step is comparing and analyzing these searching techniques with performance measurement in terms of their efficiency based on the complexity, execution time per second and size of input data, the accuracy of searching. The Quran database is approximately 6236 records. In order to do a comparison among these search techniques this work will examine:

(i) The estimated value for each searching techniques for the Quran database.
(ii) The average for the estimation value of two searching techniques.
(iii) The average time complexity of two search techniques namely B+Tree and sequential.
(iv) The search accuracy of input based on searching techniques.

2.1Searching Algorithms

Two searching algorithms implemented in this study are the B+Tree searching algorithm as well as a sequential searching algorithm. These algorithms assume the existence of a key search field. They must be modified appropriately for the case of a B+Tree on a non-key field searching for record with search key field value *k*.In a B+Tree, data pointers are stored only at the leaf nodes, therefore the structure of the leaf nodes vary from the structure of the internal (non-leaf) nodes.

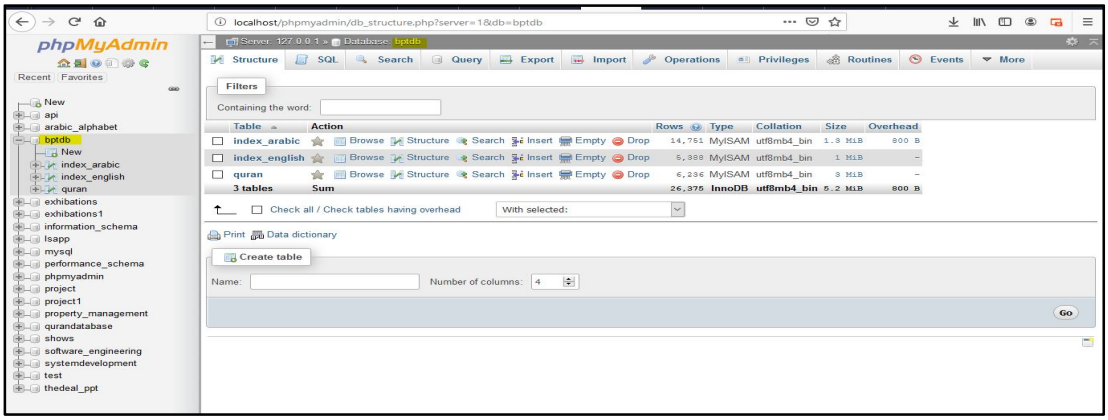

Figure 2: bptdb Database

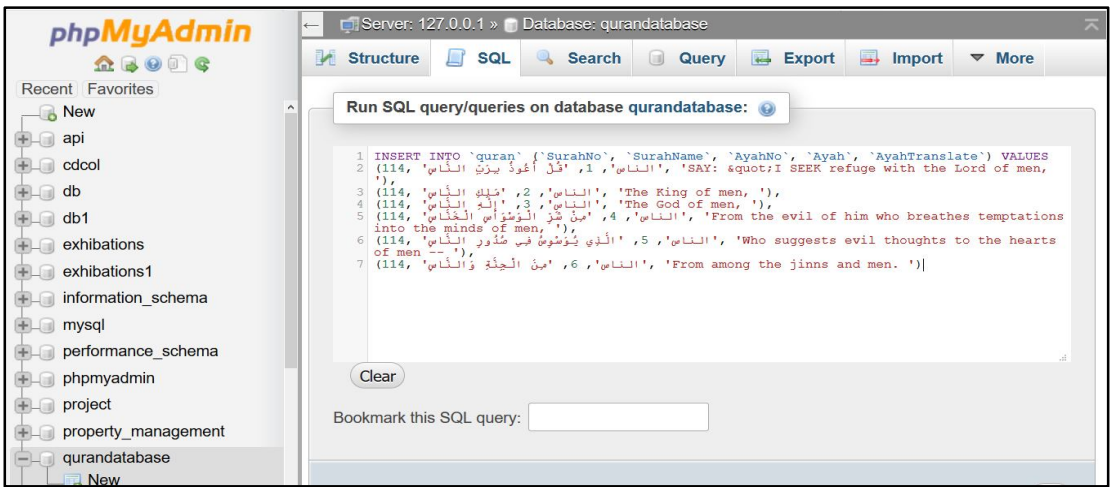

Figure 3: Data Insertion

If the search field is a key field, the leaf nodes have a value for every value of the search field, along with the data pointer to the record or block. If the search field is a non-key field, the pointer points to a block containing a pointer to the data file records, creating an extra level of indirection. The leaf nodes of the B+Tree are linked to providing order access on the search field to the record. The first level is similar to the base level of an index. Some search field values in the leaf nodes are repeated in the internal nodes of the B+Tree, in order to guide the search. There are some steps for searching in B+ Tree as follows. Let the key to be searched be *k*, then:

(i) Start from the root and recursively traverse down.
(ii) For every visited non-leaf node, if the node has a key, simply return the node.
(iii) Otherwise, we recur down to the appropriate child of the node.
(iv) If we reach a leaf node donot find *k* in the leaf node, return null.

A sequential search is a method for finding a particular value in a list that consists of checking every one of its elements, one at a time and in sequence until the desired one is found. It is suitable for small data not for large data. It is very simple and easy to understand. Searching sequentially is applicable to a table organized either as an array or as a linked list. The searching algorithm requires five parameters:

(i) The list.
(ii) An index to the last element in the list.
(iii) The target.
(iv) The address where the found element's index location is to be stored.
(v) The address where the found or not found Boolean is to be stored.

2.2Dataset

The dataset used in this research comprised of all verses in Arabic from the Quran along with the individual translation in English. In order to measure the performance of searching algorithms under study, the data is stored as a structured database using Wamp has as the local server. Wamp is a web-based development environment that supportsthe development of web applications with Apache, PHP, and MySQL databases. MySQL database is open-source software and therefore is freely accessible. It is very fast, reliable and easy to use. Figure 2 shows the Quran database named bptdb and used utf8_unicode_ci to accept Arabic and English data.

The database has 3 tables, named Quran, index_arabic, and index_english. In the Quran database, there are 5 columns, which are *SurahNo*, *SurahName*, *AyahNo*, *Ayah*, *AyahTranslate* with their data types.After creating the table, in this phase, the data in the Quran table the size of data is 2.3 MB is inserted. The record of data in the table is a complete Quran with English translation approximately 6236 rows. Figure 3 shows the data insertion in Quran table.

2.3Queries

The queries used in this project will be differentiated into two languages which are Arabic and English with two searching algorithms, which are sequential search and B+Tree recursive structure search. Figure 4 shows the interface of entering a query in the text field and specifying the query language (English or Arabic with/without diacritic) and the searching algorithm (sequential vs. B+Tree).

The proposed interface allows for cross-language Information Retrieval (IR), where the query can be made in Arabic text using English text, which is the Quran translation. The query can also be performed vice versa.

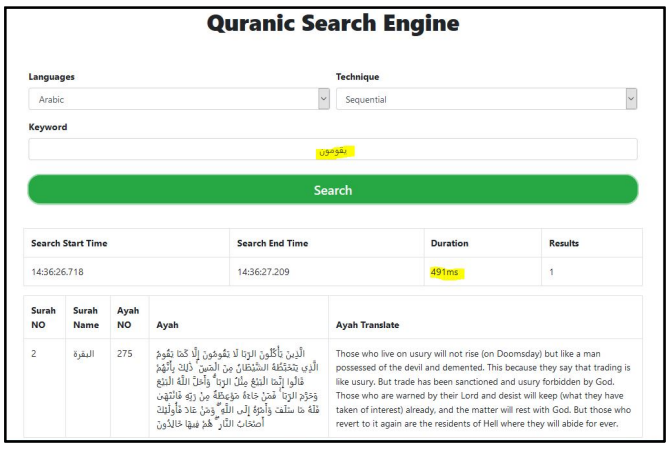

Figure 4: Query Search Interface

The results include the time complexity of searching the input query along with the location of the verses as well as their occurrence counts from the entire Quran. In this figure, it is shown that the query returned 1 search result with the time taken of 491ms. Table 1 and Table 2 show the average time taken in 10 selected query inputs.

Table 1: Results for Query Search in Arabic

Query	Arabic	Count	Sequential Search		B +Tree Search	
			Duration (ms)	Accuracy (%)	Duration (ms)	Accuracy (%)
Q1	الله	1567	10	100	1	100
Q2	الملك	26	12	97	1	97
Q3	ٱلكريم	3	9	88	1	99
Q4	الإِنسَانُ	59	3	94	1	94
Q5	جَنَّةٌ	18	97	97	1	97
Q6	نَّارِ	99	13	91	1	98
Q7	يصلون	4	485	100	1	100
Q8	كاتب	1	303	98	1	98
Q9	الناس	172	14	96	1	93
Q10	ذلك	270	131	100	1	100
Average		**221.9**	**107.7**	**96.2**	**1**	**97.6**

Table 2: Results for Query Search in English

Query	English	Count	Sequential Search		B +Tree Search	
			Duration (ms)	Accuracy (%)	Duration (ms)	Accuracy (%)
Q1	King	15	42	95	1	95
Q2	Allah	3	1	98	1	100
Q3	God	1889	19	90	1	90
Q4	Victory	14	71	97	1	97
Q5	Will	1572	79	94	1	94
Q6	Glorify	12	20	89	1	98
Q7	Punishment	234	67	99	1	99
Q8	Forgiving	87	14	94	1	94
Q9	Heaven	18	27	100	1	100
Q10	Daughters	19	4	98	1	98
Average		**386.3**	**34.4**	**95.4**	**1**	**96.6**

3. EXPERIMENTAL RESULTS

The performance of both sequential and B+Tree recursive searching algorithms are compared based on the time taken (ms) for each query using the Quran dataset. The dataset was divided into two parts; Arabic and English and was tested different searching algorithms, separately. As shown in Table 1 and Table 2, the queries covered both Arabic and English text using B+Tree and sequential search algorithms. Based on the implementation of the search algorithms, it is found that B+Tree search is faster than the sequential search.

By comparing these techniques, this research is able to find out the fastest time taken to access the data, and evaluate the results based on search time, search accuracy and identify the best searching algorithm for text [19], [20]. A summary of the comparison between B+Tree and Sequential searching techniques is projected in a chart form shown in Figure 5. Based on the results, it is shown that B+Tree is faster than Sequential based on search time, performance and accuracy.

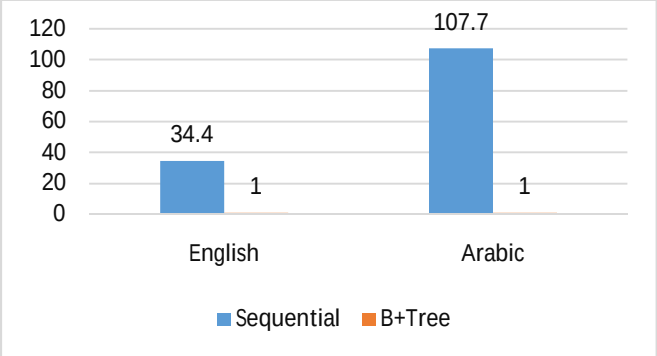

Figure 5: Results for Searching Time

4. CONCLUSIONS

This research described a comparative experiment between sequential and B+Tree recursive structure searching algorithms on Quranic text. The Quran dataset has two languages; Arabic to represent the original verses and English to represent the translation. The searching performance was evaluated based on time complexity and accuracy of searching results with different query inputs. The results showed that B+Tree performed faster searching than sequential technique. In the future, this research is hoped to extend the Quran dataset to cater to other translation languages such as Malay, Persian, and Urdu. Other than the English translation, the dataset should include hadith or tajweed to better represent the content of the Quran. The testing interface can also be extended to be a more comprehensive search engine. In this way, more searching algorithms are open for testing with this dataset such using linked list or other types of trees.

ACKNOWLEDGEMENT

This paper is supported by Research Fund E15501, Research Management Centre, UniversitiTun Hussein Onn Malaysia.

REFERENCES

1. S. A. Azzam and M. Qatawneh.**Parallel Processing of Sorting and Searching Algorithms Comparative Study**, *Modern Applied Science*, vol. 12, no. 4, pp. 143, 2018.
2. G. S. Choi, B. On and I. Lee.**PB+-Tree: PCM-Aware B+-Tree**, *IEEE Transactions on Knowledge and Data Engineering*, vol. 27, no. 9, pp. 2466–2479, 2015.
3. A. Eshtewi and S. Giuma.**Comparative Study on Data Searching in Linked List, B-Tree and B+Tree Techniques**, *Journal of Applied Microbiology*, vol. 119, no. 3, pp. 859-867, 2015.
4. K. K. Pandey and N. Pradhan.**A Comparison and Selection on Basic Type of Searching Algorithm in Data Structure**, International Journal of Computer Science and Mobile Computing, vol. 3, no. 7, pp. 751–758, 2014.
5. B. A. Jantkal and S. L. Deshpande. **Hybridization of B-tree and HashMap for Optimized Search Engine Indexing**, in *Proc. 2017 International Conference On Smart Technologies For Smart Nation*, 2017, pp. 401-404.
6. B. H. Hammo. **Towards Enhancing Retrieval Effectiveness of Search Engines for Diacritisized Arabic Documents**, *Information Retrieval*, vol. 12, no. 3, pp. 300–323, 2009.
7. A. A. Alzand and R. Ibrahim.**Diacritics of Arabic Natural Language Processing (ANLP) and its Quality Assessment**, in *Proc. 5th International Conference on Industrial Engineering and Operations Management*, 2015, pp. 227–231. https://doi.org/10.1109/IEOM.2015.7093716
8. J. M. Atwan, M. H. Rashaideh and G. Kanaan.**Semantically Enhanced Pseudo Relevance Feedback for Arabic Information Retrieval**, *Journal of Information Science*, vol. 42, no. 2, pp. 246-260, 2016.
9. M. Alhawarat.**Extracting Topics from the Holy Quran Using Generative Models**, *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 12, 2015.
10. M. A. M. Yunus, A. Mustapha, and N. A. Samsudin.**Query Translation and Quran Result in TreeMap**, *MATEC Web of Conferences*, vol. 135, 2017, p. 00072.
11. B. S. Lucas, B. Fisher, L. R. McGee, S. H. Olson, J. C. Medina and E. Cheung.**An Expeditious Synthesis of the MDM2-p53 Inhibitor AM-8553**, *Journal of the American Chemical Society*, vol. 134, no. 30, pp. 12855–12860, 2012.
12. Z. A. Adhoni, H. A. Hamad, A. A. Siddiqi and Z. A. Adhonid.**A Cloud-Based Cross Language Search Engine for Quranic Application**, in *Proc. International*

Conference on Advanced Communication Technology, 2014, pp. 218–221.

13. Q. A. Abed.**Ontology-based approach for retrieving knowledge in Al-Quran**, Ph.D. dissertation, Universiti Utara Malaysia, 2015.
14. E. T. Luthfi, N. Suryana, and A. H. Basari.**Digital Hadith Authentication: A Literature Review and Analysis**, *Journal of Theoretical and Applied Information Technology*, vol. 96, no. 15, 2018.
15. A. Abdelkader, M. Najeeb. M. Alnamari, H. Malik.**Creation of Arabic Ontology for Hadith Science**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 6, pp. 3269–3276, 2019. https://doi.org/10.30534/ijatcse/2019/96862019
16. I. S. Makki and F. Alqurashi.**An Adaptive Model for Knowledge Mining in Databases EMO_MINE for Tweets Emotions Classification**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 7, no. 3, pp. 52–60, 2018. https://doi.org/10.30534/ijatcse/2018/04732018
17. M. A. Algburi, A. Mustapha, S. A. Mostafa and M. Z. Saringat.**Comparative Analysis for Arabic Sentiment Classification**, in *Proc. of the International Conference on Applied Computing to Support Industry: Innovation and Technology*, 2019, pp. 271–285, Springer, Cham.
18. M. A. Mohammed, S. S. Gunasekaran, S. A. Mostafa, A. Mustafa and M. K. A. Ghani. **Implementing an agent-based multi-natural language anti-spam model**, in*Proc. of the 2018 International Symposium on Agent, Multi-Agent Systems and Robotics* (ISAMSR), 2018, pp. 1–5, IEEE.
19. M.Hassan andM. W. Hussain. **Header based spam filtering using machine learning approach**. International Journal of Emerging Technologies in Engineering Research, 5(10), 133-140,2017.
20. S.SaradhaandP. Sujatha.**Analysis and Significance Study of Clustering Techniques**. International Journal of Emerging Technologies in Engineering Research, 4(9), 31-33, 2016.