Indian Journal of Science and Technology, *Vol 7(10), 1643–1649, October 2014*
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

An Efficient Indexing Approach to Find Quranic Symbols in Large Texts

Vahid Rafe and Morteza Nozari

[1]Department of Computer Engineering, Malayer Branch, Islamic Azad university, Malayer, Iran; nozari_st@yahoo.com, v-rafe@araku.ac.ir

Abstract

The study presents an algorithm for precise index-based multiple pattern matching which detects Quranic verses in a text and pinpoints them. To be sufficiently precise, Arabic diacritical symbols are removed from the input text, and then a unique algorithm changes the detected strings into indices and detects Quranic verses by focusing on indices consecutiveness. To accelerate the function, the stored strings in databanks were decreased from 84845 to 13362 strings; therefore, the search speed increased. The proposed Quranic algorithm is used for text analysis, and information retrieval criteria such as recall and precision and F criteria have been used to evaluate it. The results suggest that they had a profound impact on the efficiency of the algorithm.

Keywords: Indexing, Information Retrieval, Strings Matching, Text Analysis

1. Introduction

Nowadays, access to information sources has been very easy. The Internets, large digital libraries, journals, books, etc. eliminated the problem of access to information, but the main problem remains: efficient methods to search and use the information.

Traditionally, texts used to be books, journals, and so on which followed clear syntactical and diacritical rules; however, today it is not the case, as texts and books are turned digital, and Internet sources are available. The content of websites is mostly unstructured; yet there have been attempts to introduce languages like XML to structure the content of websites.

The Holy Quran, Mulsims' most important book, has always been investigated by researchers who attempted to study this divine book from a different perspective; nevertheless, despite human-made books, the Quran contains various intellectual layers and has the potentiality to be investigated anytime and anywhere. Therefore, the present paper aims to design a system which could detect and highlight Quranic verses in different texts and provide necessary and practical statistics. The proposed algorithm in our system makes use of text analysis techniques and ensures precise multiple pattern matching to detect verses in various texts properly and precisely. This algorithm finds Quranic verses as fast as possible and at the same time it maintains the sufficient recall and precision. It is one the most important challenges which has to be encountered.

2. Related Applications

The proposed algorithm is used in text analysis and information retrieval. Text analysis is generally the process of extracting the desired information from an unstructured text. It is not something like search engine on the Web. When searching, we are attempting to find what others

*Author for correspondence

has already prepared for us; meanwhile, in text analysis we are trying to find new pattern, pieces of information which may not be obvious[1,2].

Patterns matching algorithms are one of the practical methods in text analysis and information retrieval, which are used in a wide range of content filtering and text processing systems. The main problem in pattern matching is to search and find all the occurrences of a certain pattern in an input text. There are various pattern matching algorithms. The present paper uses an index-based multi-string matching algorithm.

Information retrieval: data are not ambiguous; still, information need to be interpreted, hence ambiguous. The systems designed for data retrieval do not need to remove ambiguity; however, in information retrieval systems, information need to be modeled as much as possible to decrease ambiguity.

Precision, recall, and F criteria are used in information retrieval systems, which will be discussed later. In the next section, related applications are categorized and explained based on their techniques.

2.1 Classification Based on String Matching

There are two main techniques for string matching:

Exact String Matching: The pattern is exactly found in the text, i.e. it finds all the patterns which are totally matched against the input text.

Approximate String Matching: All the similar patterns are found in the text.

Classification based on the Number of Patterns:

Based on the number of patterns to be found in a text, string matching algorithms are classified into two important categories:

Single Pattern Matching: only one pattern is searched in a text.

Multiple Pattern Matching: more than one pattern is searched in a text. It can search several patterns at the same time[3–5].

2.2 Classification of Exact Multiple Pattern Matching Algorithms

Exact multiple pattern matching algorithms are generally divided into two groups:

Shift Table-based Algorithms: They make use of several pre-determined tables to spot the next location that the pattern may occur.

Automaton-based Algorithms: They make use of an NFA or DFA-based display to show regular expressions. Since NFA display is influenced by information storing, it is usually slower than DFA[6].

The proposed algorithm in this paper is inspired by the shift table-based algorithms. This algorithm has the potentiality to be used in texts other than the Quran, and no algorithm has ever been used for a wide range of data.

Most applications function like search engines to detect verses in a text. A verse or part of it is fed into the application and it highlights the occurrences in the text.

Pattern matching algorithms go through two stages:

Pre-process stage: Complete information to optimize the number of comparisons is retrieved about the pattern.

Process/search stage: Patterns are spotted using the information retrieved in the previous stage[1,2].

In a study on the Quran, a combined method is used to classify the Quran based on the chronological order of Suras, which is a combination of statistical and information-based methods. This classifying application is fed with the date of Suras revelation and then the revelation date of the input Sura is estimated by the application[7].

In another research, an efficient program called DataQuest is proposed to model and retrieve information from Quran-related distributed information resources, which presents an ontology based on information extraction from Quran-related documents using techniques like Web semantics, information extraction and natural language process. These documents are explained by ontology and then a smart meaning-based search engine filters user's queries and extracts the related information[8].

Another research presented a method to categorize the Quran based on Prophet Mohammad's (pbhu) life phases. This method makes use of machine learning techniques. Initially, Suras are divided into Meccan and Medinan and then each Sura in each section is clustered by single link fuzzy clustering method[9].

Quranic literature is focused on how to store information and how to get access to verses, and few studies have been conducted into large input data. The present paper attempts to introduce a new method in information storing and retrieval.

3. Statement of the Problem and Challenges

In a nutshell, this study aims to retrieve Quranic verses in different texts. It may seem easy; however, when

the challenges are stated, the differences from similar functions will become clear.

Muslims' holy book, the Holy Quran, enjoys unique features which distinguish it from other books and written texts. The Quran contains 84845 words each of which have a different meaning and diacritical symbols. Some challenges that distinguish the Quran from other books are as follows:

3.1 Magnitude of Patterns Data

In order to perform precise searching and information retrieval operations exclusively for the Quran, every word have to be stored in a databank which contains 84845 words. When such operations have to be done on large books like a 600-page book and the algorithm has to match patterns with 84845 patterns, the time of search increases dramatically.

3.2 Different Forms of Patterns

One of the distinguishing features of the Quran is the way of writing, i.e. a single pattern is written differently in two books. It is known as "a form of writing" in religious books. As an example, the word "باتکلا" is written as "باتکلا" in Iranian form of writing (Tahir Calligrapher); yet in Egyptian form of writing (Uthman Taha) it is written as "بتکلا". Now this writing problem adds to the previous problem that is 84845 patterns in the databank.

3.3 Diacritical Problems of Patterns in a Text

Another challenge is that the input text might have diacritical problems. For example, there is a Quranic verse in the input text, but because vowel diacritics are not properly placed (It is typed نمُ الذینَ instead of نمِ الذینَ) information retrieval will be problematic or sometimes the input text might have no Arabic diacritics at all.

3.4 Searching Consecutive Patterns

Each verse is composed of some patterns (words). Not only should they be from the Quran, but they also have to be consecutive in order for the algorithm to detect them as Quranic or not?

4. The Proposed Solution

The present paper aims to propose a method to search and retrieve Quranic verses from different texts. Each verse is composed of several patterns (words). Therefore, we have to ensure that a certain pattern is present in the databank and we should also check the consecutive patterns, and finally it can be said whether it is a Quranic verse or similar to Arabic words or not at all in the Quran.

Therefore, an algorithm has been designed which uses "Exact Index-based Multiple Patterns Matching" to achieve our goals. The reasons why this name is chosen are briefly explained:

Exact Multiple Patterns Matching: we are looking for Quranic verses and each verse is composed of several patterns, that is, several patterns have to be checked simultaneously. On the one hand, the consecutiveness of any patterns is not important, since there might be several patterns in a text which match the databank. On the other hand, they may/may not be consecutive, yet not a Quranic verse. Thus, we are looking for patterns (multiple patterns) which are consecutive and also match the patterns in the Quran (exact matching).

Index-based: The less the searching algorithm is dependent on a text, the better results it can provide, as it can search a wider range of texts and the searching time reduces dramatically. The algorithm, detecting Quranic verses in different texts, goes through three stages:

Preparing: Building databanks for the Quran.

Pre-processing: Processing the input text.

Retrieving Quranic words and providing index-based output.

Processing: Searching the consecutive patterns and detecting Quranic verses.

4.1 Building Databanks for the Quran

In this stage, the following necessary information are built and implemented: single-word, double-word, triple-word, and writing forms databanks.

In the single-word databank, every word in the Quran is stored one by one with a unique index. The stored patterns numbered to 84845 with unique indices. Given the magnitude of input text, the searching time of these patterns may increase. The problem is that a Quranic word has several unique indices, so the databank has a surplus. To overcome this problem, initially we built a new databank and remove the Arabic diacritic symbols such as.....ّ ٌ ً from the present strings in the single word bank. Therefore, words like نمِ is not different from نمَ or نمَ all these forms are stored as نم with no Arabic diacritic symbols with a unique index in the new databank. Therefore,

the patterns in the main Quranic databank were reduced from 84845 to 13362.

In this study, it is assumed that if four or more searched patterns are in the Quran and consecutive, they are identified as a Quranic verse. Therefore, we may miss many single-word or double-word verses of the Quran, e.g. الهی کم التی اکثر.

Another problem lies in the language of the Quran, i.e. Arabic. The exact or similar words of many Quranic words are found in the colloquial language of Arabic-speaking countries (Lahjah Arabic); thus, any two words cannot be detected as a verse of the Quran. Figure 1 displays how double-word or triple-word databank is formed.

If m is the total number of words in a text and n is the repetition of double-word patterns, m/n indicates the frequency of a certain phrase in a text. If this proportion exceeds a threshold, here 1000, it is detected as a double-word verse.

To reduce errors, output verses are reviewed by a person after the above stage. A similar criterion is used for the triple-word databank. In the writing forms databanks, a pattern has to be defined.

Pattern: a set of strings and each string is made of a set of symbols.

There might be different forms of a word, hence different patterns, for example the word الکتاب in the verse 2 of Sura al-Baqara is written as"ذلک الکتاب بی ریب فیه" in Iranian writing form but "ذلک الکتب بی ریب فیه" in Uthman Taha writing form.

The problem that distinguishes the Quran from other books lies in the writing forms. It means that the words have the same pronunciation but different spellings, e.g. the word الکتاب in the above example. To overcome this problem, we go through the following stages:

First, the difference of writing forms are fed into another application and then the Quran is fed to the application to find the differences with the Iranian writing

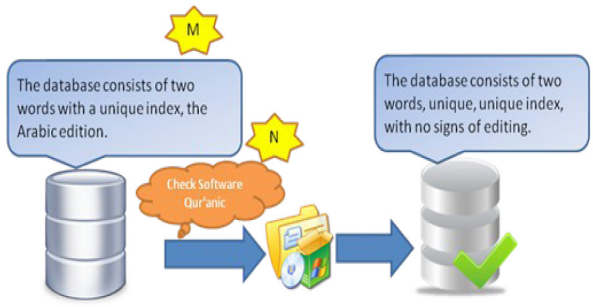

Figure 1. Create a database of two-word and three-word.

form which is the basic form. To enhance precision, the output is carefully reviewed by some experts. This is done for recognized writing forms.

The application might find a writing form which is quite new or not recognized. In this case, Arabic words, which may not be recognized as Quranic verses due to lack of a writing form, are highlighted to warn the user that they might be from the Quran.

4.2 Input Text Processing

Firstly, Arabic diacritic symbols ..... are removed. The reasons are explained as follows. The algorithm processes different types of texts, and verses might be written in different forms, for example, this verse might be written in two forms: "ذٰلِکَ الْکِتٰبُ لَا رَیْبَ فِیهِ" or without diacritic symbols: "ذلک الکتاب لا ریب فیه" . If the search is done with diacritic symbols on, a lot of words might be lost and the result of the search is not valid; however, if it is done without the symbols, we will get better results.

The input text might have incorrect diacritic symbols and they are not put in the correct place on the words; for example, if it is written "ذٰلِکَ الْکَتٰبُ لَا رَیْبَ فِیهِ", the search results will not be acceptable.

Consequently, the first stage is to remove Arabic diacritic symbols from the input text.

Secondly, the algorithm refers to a single-word databank (of the whole Quran) in which all the unique patterns are stored without Arabic diacritic symbols (13362 patterns stored in this databank), and the pattern is matched against the input pattern; if the pattern does not match, it goes to the next word in the text.

Thirdly, after the pattern is matched, the algorithm retrieves the pattern index form the databank. The searching and pattern index retrieving operations proceed until a mismatch occurs.

The three stages explained above are summarized below:

Arabic diacritic symbols are removed from the input text.

The words in the text are matched against the single-word databank.

If patterns match in the single-word databank, its index is retrieved from the databank.

The previous stage continues until a mismatch occurs in the input text.

Retrieving Quranic Words and Providing an Index-based Output

When a pattern match occurs, its index is stored in a file and pattern matching operations proceed until a mismatch occurs. If so, a list of reviewed indexes and word are sent to the next stage. For example, if

"... ختم الله على قلوبهم" is in the input text, the following output goes from pre-process stage to the next stage after the above operations. (Figure 2)

4.3 Searching Patterns Consecutiveness and Detecting Quranic Verses

The proposed algorithm is designed to search and retrieve information as fast as possible in addition to having sufficient recall and precision. These two important goals are realized in this stage.

To reach maximum search speed, multiple strings are used. Given the type of hardware and experimental results, at most ten strings are included in the application to process the output as fast as possible when it goes through the pre-process stage. Therefore, the system functions quickly and when the pre-processed output is prepared, it looks for consecutiveness in the processing stage.

This algorithm is designed so that when ten processes are being executed and a new request is received, it goes to the waiting list. When a process is executed, it frees the memory and the processor automatically and waits for a request. To achieve the recall and precision and also to enhance the search speed, every process is executed according to the following algorithm:

First Stage:

The subcategories of the indices which were sent to the processing stage- the previous stage- will be opened. (Figure 3)

Second Stage:

Algorithm has to answer these two questions:

1-Is the first index of the second pattern larger than the last index of the first pattern + 1?

If so, there is no consecutive pattern. As an example, the last index of the first pattern in the above figure is 73493. If the first index of the second pattern equals 73495, then there is no consecutiveness of patterns and it is quite logical.

2-Is the last index of the second pattern larger than the last index of the first pattern +1?

If not, we move to the next stage. We have to narrow down the search by removing all the indices smaller than the first index of the first pattern and larger than the last index of the first pattern +1 from the next pattern. (Figure 4)

After removing the indices, we move to the next pattern and the same removing process continues as above. The full stage is illustrated below (Figure 5)

Then, the algorithm searches for consecutiveness in the remaining indices; therefore, the only consecutive indices in this stage are 90, 91, 92, and 93. It should be noted that since indices are ordered from the smallest to the largest, binary searching algorithm is used to find consecutiveness.

Ten processes carry out the above stages simultaneously. They receive the output from the pre-process stage and search for consecutiveness and verses. It should be noted that it is done at the same time as the pre-process stage. The processes refer to a file and if it is empty, the pre-processes go to the waiting list; otherwise, they search for consecutiveness in verses based on the information in the pre-process stage.

Figure 6 and 7 show examples of applications output in various situations.

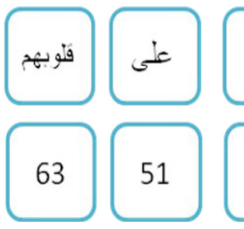

Figure 2. The output of the preprocessing stage.

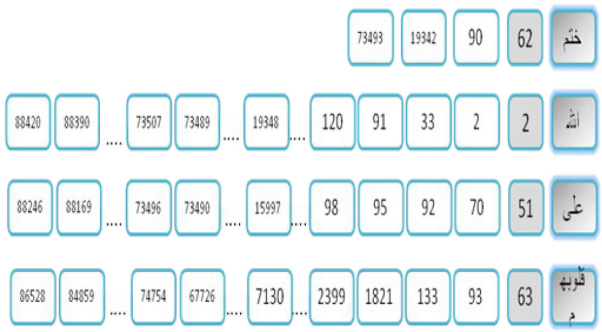

Figure 3. The pre-processing sub-index was delivered.

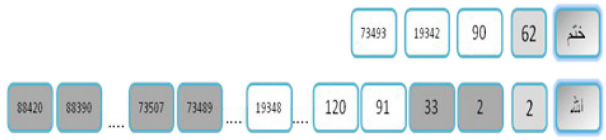

Figure 4. First, remove the index that are not within the scope of the search.

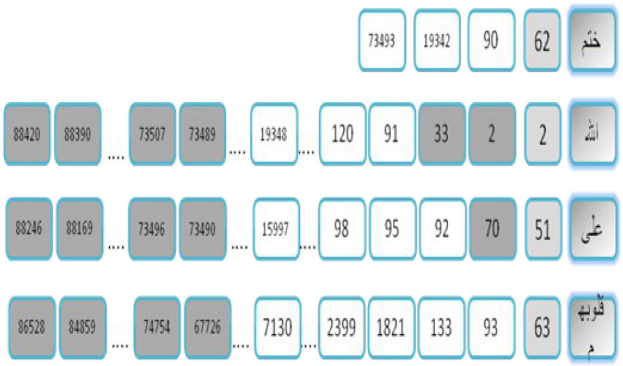

Figure 5. The third step is to remove indexes that are not within the scope of the search.

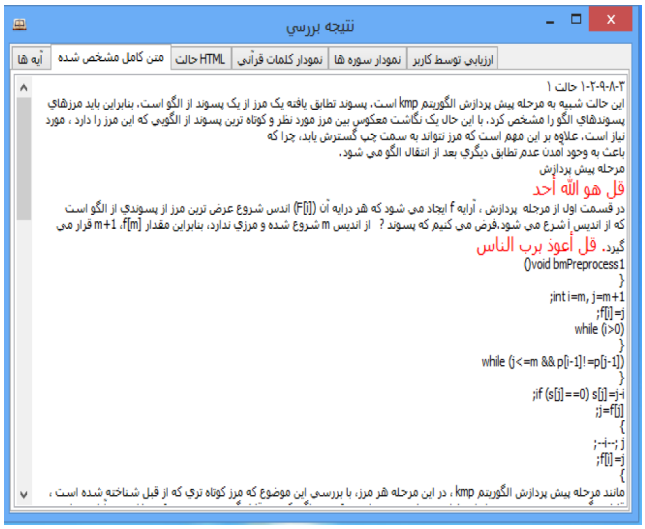

Figure 6. The software outputs when the verse and the verse addresses the show.

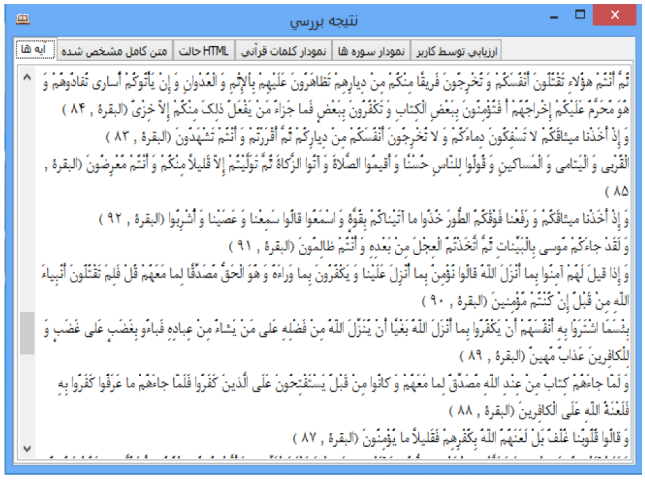

Figure 7. Found in this verse is distinguished from other texts.

5. Evaluation and Results of the Experiments

To measure the efficiency of the proposed algorithm, information retrieval criteria are used. The most important criteria are recall, precision, and F criterion.

Recall is the proportion of the correctly detected Quranic words to all the words in the Quran.

Precision is the proportion of the correctly detected Quranic words which also have consecutiveness (verses) to all the detected Quranic words.

F criterion is the average of the two above criteria, i.e. recall and precision, which shows their efficiency at the same time.

To apply these criteria on the proposed algorithm, we went through the following stages:

Each word in a text has four forms: it is Quranic and the system detected it so (TP); it is Quranic but the system detected it not (FP); the word is non-Quranic but the system detected it as Quranic (TN); it is non-Quranic but the system detected as Quranic (FN).

To evaluate, first the input text is divided into two sections: words detected as Quranic and as non-Quranic by the system. These two sets of words will be compared with the Quranic verses fed to the application. These verses are considered as yardsticks. First, the verses detected as Quranic are compared with the yardstick to evaluate the two forms (Quranic, Quranic) and (Quranic, non-Quranic) and measure the recall, precision and F criterion. Second, the words detected as non-Quranic will be compared with the yardstick to evaluate the two forms (Quranic, Quranic) and (Quranic, non-Quranic) and put the results in the relevant formula to measure the recall, precision, and F criteria.

Finally, the average will be calculated as the result of the evaluation.

The formulae of recall (1), precision (2) and F criteria (3) are as follows.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$F = 2 * \frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3}$$

To ensure the results to be as precise as possible, we choose the input text that contains Persian, Arabic and

Table 1. Table evaluate different frequency

Type of input	Number of words	Frequency Quran	Frequency of non-Qur'anic	Average accuracy	Average precision	Average F
Persion, Arabic English	1520	Low	High	0.883	0.876	0.879
	1390	High	Low	0.973	0.943	0.957
Arabic Text	6391	Low	High	0.892	0.889	0.891
	6520	High	Low	0.936	0.917	0.926
English Text	5513	Low	High	0.982	0.982	0.982
	5642	High	Low	0.964	0.957	0.961
Average				0.938	0.927	0.932

English words, a text with only Arabic words, and another with only English words. In table 1, we have changed the number of Quranic words, as the frequency of Quranic and non-Quranic words. The frequency of Quranic words means the number of Quranic verses in the input text. The frequency of non-Quranic words means the number of non-Quranic words in the input text.

To evaluate the application for other texts, an Arabic text without Quranic words was fed to the system. The output was successful; no Quranic verses were detected. Another text, i.e. an English text, was also fed to the application with the same results as the Arabic text.

6. Conclusion

This study proposed an index-based multiple pattern matching algorithm exclusively for the Quran. This paper tried to provide answers to the following questions:

1. Users may know what percentage of a book contains Quranic verses and which Suras are included and whether there is a desired verse in a certain book. In fact, the algorithm provides a Quran-related summary of the text of a book.

2. Recall and precision: If the Quran is stored word by word, there are 84845 patterns; and if writing forms are added, databank volume will increase. To overcome this problem, Arabic diacritic symbols are removed to reduce the patterns to 13362; thus, such symbols will not affect the search results.

So far, in the searching and information retrieval operations for the Quran, a verse was fed to the system and it could detect if a verse is from the Quran or not. The proposed algorithm is applicable on a lot of large books in various languages and provides complete reports on what percentage of the book is Quranic or non-Quranic, which Suras are referred to, and also a Web-based output.

7. References

1. Bhukya R, Somayajulu D. An even odd multiple pattern matching algorithm. IJEST. 2011 Mar; 3(3):2118–26. ISSN: 0975-5462
2. Bhukya R, Somayajulu D. Index Based Sequential Multiple Pattern Matching Algorithm Using Pair Indexing. International Conference on Life Science and Technology IPCBEE; 2011; Singapore. IACSIT Press; Singapore:3:1100–4
3. Bhukya R, Somayajulu D. Index based multiple pattern matching algorithm using DNA sequence and pattern count. International Journal of Information Technology and Knowledge Management. 2011 Jul-Dec; 4(2):431–41.
4. Bhukya R, Somayajulu D. Exact Multiple Pattern Matching Algorithm using DNASequence and Pattern Pair. Int J Comput Appl. 2011 Mar; 17(8):0975–8887.
5. SaiKrishna V, Rasool A, Khare N. String Matching and its Applications in Diversified Field"s. IJCSI. 2012 Jan; 9(1):219–26. ISSN (Online): 1694-0814.
6. Kandhan R, Teletia N, Patel JM. SigMatch: Fast and Scalable Multi-Pattern Matching. Proceedings of the 36th International Conference on Very Large Data Bases; 2010 Sep; Singapore.
7. Nassourou M. A Knowledge-based Hybrid Statistical Classifier for Reconstructing the Chronology of the Quran. Accepted in WEBIST/WTM, The Netherlands, 2011.
8. Ain QU, Basharat A. Ontology driven Information Extraction from the Holy Qur'an related Documents. 26th IEEEP Students' Seminar 2011 Pakistan Navy Engineering College National University of Sciences & Technology. 2011.
9. Nassourou M. Using Machine Learning Algorithms for Categorizing Quranic Chapters by Major Phases of Prophet Mohammad's Messengership. Department of Computer Philology & Modern University of Würzburg; 2011.