Searching and Analyzing Arabic Text Using Regular Expressions e–Quran Case Study

Ahmed ARARA, Adel SMEDA, and Ismail ELLABIB

Abstract**—** With the advancement of the social networks which became an essential media of communication, new web techniques are needed. This work aims to investigate the usage of new web programming techniques for e-services in Islamic studies. As a case study, the holy Quran as an electronic document is stored, and manipulated as XML form. The application uses PHP as a programming language and Joomla! content management system as a tool for managing information sourcess. It applies XML technologies for manipulating and processing e-books. The process of manipulation is intended to extract the holy Quran verses' interpretations based on multiple interpretation e-books. In addition, searching techniques based on regular expressions designed and developed for good results of searching and understanding the holy Quran. Regular expressions are exploited to query Arabic textual e-books. The project uses other web programming tools such as HTML, CSS, JavaScript, and AJAX.

Keywords**—** e-services, Text processing, pattern matching, XML text representation, regular expressions.

I. Introduction

IN e-society, the social networks (Facebook, Twitter,..etc) have become the essential media of communication. The Muslim society is no exception with respect to technology use and adaptation. Internet access by Muslims in their daily life has become necessary. Hence, developing a portal by Muslims and for Muslims will enforce screening and filtering of its contents and moreover the adoption of technology to the needs of Muslim users. In fact, our aim is to allow the user to navigate efficiently the voluminous data sources. Queries processing is made via XML query and/or SQL query based on MySQL database. An Islamic web portal, as a central place for making all types of information accessible to an audience of varying range, is developed to provide information in "one stop shop". processing information over the web.

As a matter of fact, semi-structured data (in XML format) can be the savior of representing, exchanging, and processing data sources in the web. Structured data (databases) can be easily manipulated as semi-structured data when stored in XML format.

II. Related work

There are numerous works regarding the issue of searching and content management of Arabic text. Here are some relevant related works that are closer to our work:-

A. Search engines and Arabic text processing

The popular search engines, like Google and Yahoo!, with their powerful crawlers that keep track of web pages to find matches between their contents and the user's searching words and patterns via regular expressions. Archie, one of the first search engines, used regular expressions exclusively to search through a database of filenames on public FTP servers[9]. Regular expressions were chosen for these early search engines because of both their power and ease of implementation [10]. The long history of experiences for such popular search engines [9] makes improvement to their searching techniques. In fact, knowledge bases are used to support key words search in multiple languages. In Arabic Language processing, regular expressions allow collocations, syntagms and word associations to be easily extracted from an Arabic text by using time-saving, semi-automatic queries.

Mansour [5] is an Arabic application for analyzing Arabic text, the application features as written in the website are:- Part of speech tagging, morphological analysis, word diacritization, and syntactic analysis. The application is programmed in C# language, used MS Access database and General Public License version 3.0 (GPLv3).

B. Islamic sites

"Islamweb" [3] is an Islamic website that offers some e-services. The site gives an ease to navigate screen for moving through its components. IslamWeb.net, host databanks containing other historical texts. This sometimes involves digitizing old manuscripts, an expensive and time-consuming project that not all sites can afford. "quran.al-islam.com" is another well interpretation website that provides the basic interpretation books, navigated by Sura name and aya (verse) numbers. There are shortcomings of this site that it provides a text box instead of selection box to select the starting verse in a Sura which causes an ambiguity to read. The site also offers indices for the holy Quran Suras. The way indices are displayed shows that they are indices of static HTML pages. The system does not own or have an access to a database and consequently queries are not possible.

IslamOnline (IOL) portal [3,4] provides counseling

Ahmed ARARA , and Ismail ELLABIB, College Of engineering /Tripoil Universisty Libya

Adel SMEDA, College of accounting / University of Aljabel Algharbi Libya.

services in several subject areas. At IOL, legal opinions or fatwas are searched, processed, edited or issued and published by the Fatwa department. one of the most-visited Arabic/English Islamic web portals which issue fatwas. IOL is intended for Muslims as well as non-Muslims audiences. It aims to discuss news, receive advice and communicate, but it also aims to correct or complicate the often simplistic image of Islam that other media present to non-Muslims.

It should be noted that IslamOnline invites not only religion scientist (sharia experts) to give advice, but also academics from fields including sociology, political science, psychology, medicine and economy, and sometimes even from literature or the arts. This is due to a belief among IOL founders that muftis cannot often give answers to questions which require special knowledge outside the framework of Islamic jurisprudence and theology.

The above portals offer users information on religious practice and its respective Contemporary interpretation. Hence, through legal opinions (fatawa), and an established genre of Islamic jurisprudence (fiqh), users can search online databases for previously issued fatwas by subject or the name of the issuing mufti, or use an online form to request a new fatwa tailored to their personal situation. In fact, a search tab giving results in a form of a table of hyperlinks. However, the Arabic language is neither analyzed well nor used formal patterns in their search engines. Consequently results are not always precise.

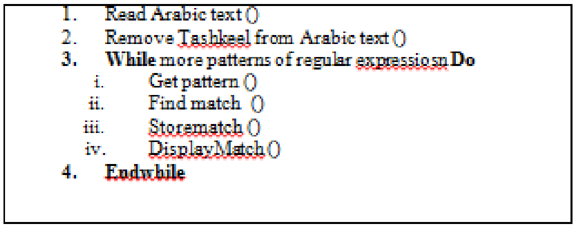
III. Arabic Text Searching

A regular expression is string of characters coupled with some special meta-characters (i.e. ^…, […], $, *…) that, when applied to another text string, provides a concise and flexible means to specify and recognize strings within the text, such as particular characters, words, or patterns of characters.

There are several flavors of regular expressions [6]. In this paper we are talking about PHP Regular expressions. PHP supports two types of regular expressions namely:- POSIX-extended and Perl-Compatible Regular Expression (PCRE). PCRE is more powerful and has more functions. The PCRE functions are classified as:- matching, replacing, filtering and validating. Our focus will be on searching functions of Arabic text. Patterns are constructed to search either through the whole Quran book or specific Suras or a range of Ayat (verses). The searching mechanism is extended to the well-known Quran interpretation books (Tafseer books). The general search algorithm of Arabic text is as shown in table I below:-

Table I

Arabic text Searching algorithm



```
1.  Read Arabic text ()
2.  Remove Tashkeel from Arabic text ()
3.  While more patterns of regular expressiosn Do
      i.   Get pattern ()
      ii.  Find match  ()
      iii. Storematch ()
      iv.  DisplayMatch ()
4.  Endwhile
```

Patterns are built based on regular expressions rules. A database of patterns is developed and a corresponding XML document is generated and consequently queried to provide answers to users from diversified multiple XML document.

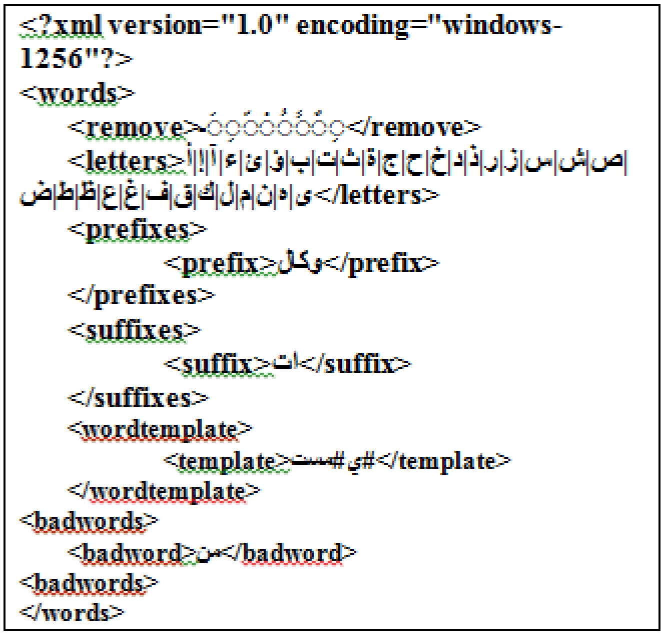
IV. Implementation of Searching Mechanism

A. XML as a container of Arabic words

XML is used to build a structure of the templates that will help in abstracting the Arabic words. Table 2 shows the main structure of the "word.xml" document that holds all these templates.

Table II

XML file as a container of patterns



```
<?xml version="1.0" encoding="windows-
1256"?>
<words>
    <remove>ًٌٍَُِّْ</remove>
    <letters>ا|أ|إ|آ|ء|ئ|ؤ|ب|ت|ث|ة|ج|ح|خ|د|ذ|ر|ز|س|ش|ص|
ض|ط|ظ|ع|غ|ف|ق|ك|ل|م|ن|ه|ى</letters>
    <prefixes>
            <prefix>وكل</prefix>
    </prefixes>
    <suffixes>
            <suffix>ات</suffix>
    </suffixes>
    <wordtemplate>
            <template>ـــست#ي#</template>
    </wordtemplate>
<badwords>
    <badword>من</badword>
<badwords>
</words>
```

The "remove" tag contains the characters that always removed from the text before editing (Tashkeel), while "letters" tag contains all the Arabic letters (used in regular expressions since the Arabic letters don't have collating sequence in the ASCII code). The "prefixes", "suffixes", and "wordtemplate" are patterns that will be used to make regular expressions too, where prefixes object contains all the Arabic language words' prefixes, and suffixes object contains all the Arabic language words' suffixes, while "wordtemplate" object holds patterns used to find the roots of the words, it is used after removing the prefixes and suffixes from it.

The "badwords" object holds the Arabic words that are much repeated and affect the speed and efficiency of the search and without affecting its meaning, like "the" in English language. The "prefixes", "suffixes", "wordtemplate", and "badwords" are collected from the access database of "Mansour" software, since it is open source software, and then converted into xml format.

B. Abstracting Arabic words using regular expressions

To abstract an Arabic word, first of all we must remove the

Tashkeel from it, we can do this using the function “RemoveTashkeel()” which uses the object “remove” from the XML document to do this function as follows:

TABLE III
FUNCTION TO REMOVE TASHKEEL

```
Function RemoveTashkeel( txt )
{
     Reader= XMLReader::open(words.xml)
     Tashkeel=Reader->read(Tashkeel)
     Tashkeel=explode(“ “ , Tashkeel)  //Tashkeel is an
array contains all the Tashkeel
     txt= str_replace ( Tashkeel , "" ,txt)
}
```

The second step is to remove suffixes and prefixes from the word,the following function shows how to remove the suffixes of the word.

TABLE IV
FUNCTION TO REMOVE PREFIXES

```
Function RemovePrefixes(word)
{Reader = XMLReader :: open(“word.xml”)
//A while loop is reading all prefixes from the xml and
store them  in an array
While (! Reader-> read (prefix))
{ prefixes [ ]= Reader->value}
Foreach (prefixes as prefix){
newWord =
ereg_replace("^(".prefix."){1}", "", word);
                  If(strcompare(newWord, word) !=0)
                  { Searchwords[ ] = newWord}
 //add the new word to the searchwords array
      }
Return Searchwords
}
```

The RemoveSuffixes function is as the RemovePrefixes function except the regular expression form, it will be as follow:-

```
newWord = ereg_replace("(".suffix."){1}$", "", word)
```

The “ereg_replace” is a PHP function that search the regular expression pattern in a string and remove it if exists, if not the origin word will be returned. The “^”symbol in a regular expression means that the string must start with this pattern, while the “$”symbol means to end with this pattern.

The third step to abstract a word is to find the roots of the words; here we will use the “wordtemplate” object in the XML document to do that. The function is the responsible of this process.

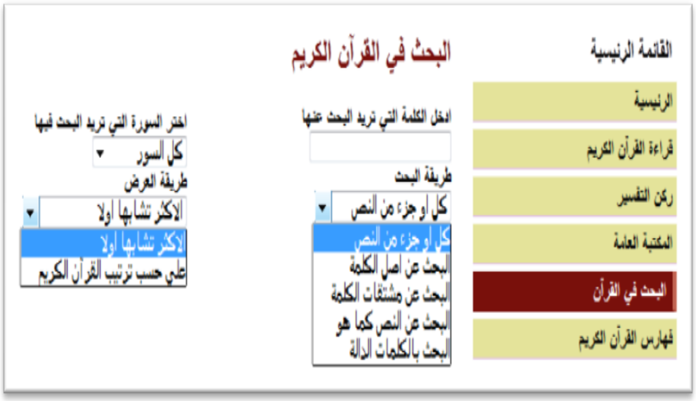
TABLE V
FUNCTION TO BUILD WORD TEMPLATES OF ARABIC LANGUAGE

```
Function wordtemplates (words)
{
Reader = XMLReader :: open(“word.xml”)
Reader-> read (letters)
letters= Reader->value
While (! Reader-> read (template))
{Template= Reader->value
 patterns[]="^".str_replace("#","(".letters."){1}", Te
mplate)."$"; }
Foreach (words  as  word ){
      Foreach (patternsas  pattern){
                  If(ereg(pattern, word)
                  { newWord = ereg_replace (pattern,
word)
      Searchwords[ ] = newWord}  //add the new word
to the searchword array
      }
    }
returnSearchwords
}
```

Since the products of the RemoveSuffixes and RemovePrefixesfunctions were arrays then the wordtemplates function input is array that holds each word as an array element, then apply the abstraction process over it. The returned value “Searchwords” is an array holds all the possibilities to the root of the origin word.

C. *Searching a string in the Holly Quran*

To search a string in the holy Quran a form with four inputs is used; the four inputs are the string to search, the search region, the search type and the displaying type.



البحث في القرآن الكريم

ادخل الكلمة التي تريد البحث عنها

طريقة البحث: كل او جزء من النص

اختر السورة التي تريد البحث فيها: كل السور

طريقة العرض: الاكثر تشابها اولا

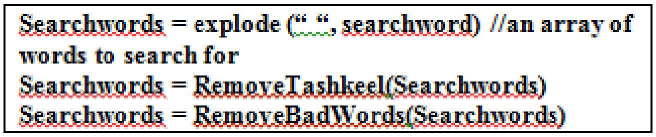
Fig. 1 GUI of search engine of Arabic Text

The search region can be one of the Suras specified by its ID, or all Suras by setting the ID to (-1).

The search type field offers four searching types with four searching algorithms.

The first type of search is the “part or all string searching”, the main idea of search is to split the string into words searching for each word separately, after removing the bad words. So in this kind of search there is no need to abstract the words. The algorithm of this search type will start first by preparing the searching words like this:-

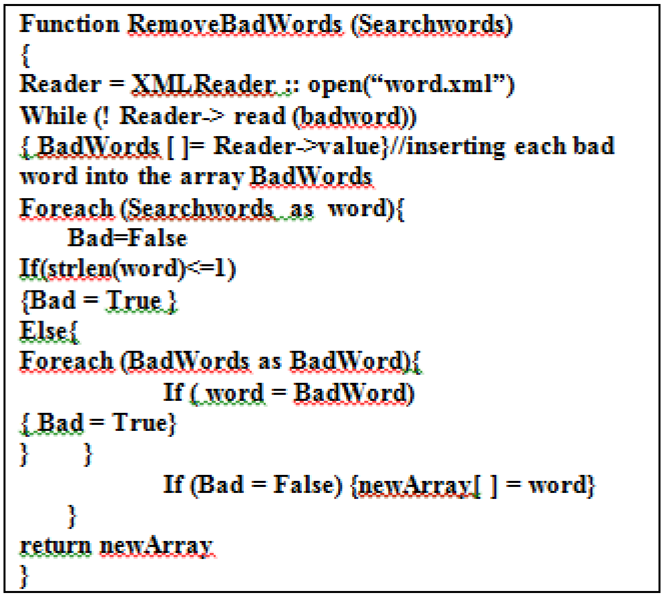
TABLE VI
FUNCTION TO REMOVE PREFIXES



```
Searchwords = explode (“ “, searchword) //an array of
words to search for
Searchwords = RemoveTashkeel(Searchwords)
Searchwords = RemoveBadWords(Searchwords)
```

The RemoveTashkeel function, which was discussed previously, removes the Tashkeel from the text, while the RemoveBadWords function removes the unwanted words from the array of searching words; it works according to the following algorithm.

TABLE VII
FUNCTION TO REMOVE COMMON WORDS

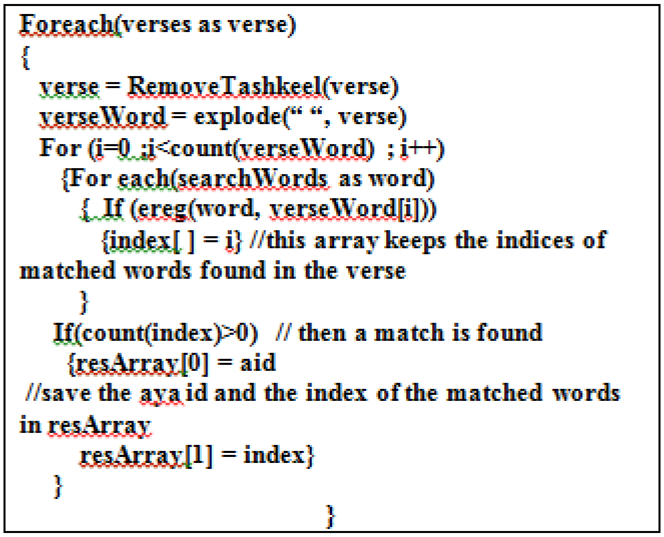


```
Function RemoveBadWords (Searchwords)
{
Reader = XMLReader :: open(“word.xml”)
While (! Reader-> read (badword))
{ BadWords [ ]= Reader->value}//inserting each bad
word into the array BadWords
Foreach (Searchwords  as  word){
      Bad=False
If(strlen(word)<=1)
{Bad = True }
Else{
Foreach (BadWords as BadWord){
                  If ( word = BadWord)
{ Bad = True}
}      }
                  If (Bad = False) {newArray[ ] = word}
      }
return newArray
}
```

The result of this function will be an array of words that are ready to be searched.

The search can be done by reading each verse in the holy Quran, in case the search region is a specific Sura then the Sura ID will be checked first, and if “all Suras” is selected then it will search all verses in the Quran, the search started by removing the Tashkeel from the verse then split it into words searching word by word using regular expression, and if matched then the index of the word (in the verse) is saved in the index[ ] array, to help in displaying colorized searching words, So distinguishing the matched words. The purpose of searching word by word is to allow searching the inserted words in different sequences, which improve the efficiency of the search.

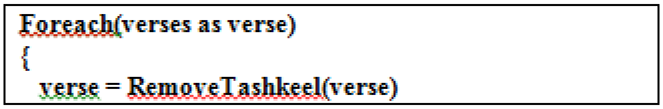
TABLE VIII
OPTIMIZED SEARCH ACCORDING TO QURANIC VERSES



```
Foreach(verses as verse)
{
  verse = RemoveTashkeel(verse)
  verseWord = explode(“ “, verse)
  For (i=0 ;i<count(verseWord)  ; i++)
    {For each(searchWords  as word)
      { If (ereg(word, verseWord[i]))
         {index[ ] = i} //this array keeps the indices of
matched words found in the verse
      }
    If(count(index)>0)  // then a match is found
     {resArray[0] = aid
 //save the aya id and the index of the matched words
in resArray
      resArray[1] = index}
    }
                        }
```

The second searching type is the “search for roots”, first removing the Tashkeel from the entered words, and then the searching words are abstracted from the prefixes and the suffixes of the Arabic language, and finally the roots are brought on. The result of the three functions will be an array of words entered by the user plus the abstracted words, the process of abstraction sometimes produces unwanted results such as the bad words or empty words, to exceed this problem it calls the RemoveBadWords( ) function over the array of words before executing the search to ensure the absence of these mistakes.

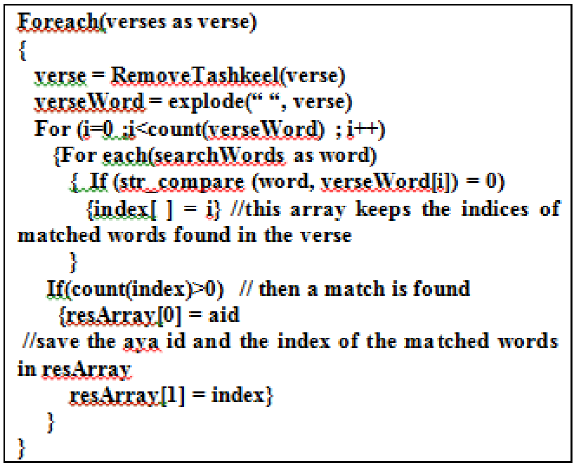
TABLE IX
SEARCHING FOR ROOTS OF WORDS



```
Foreach(verses as verse)
{
  verse = RemoveTashkeel(verse)
```

The search is then done with completely matching words instead of using regular expressions. So the str_compare( ) function is used instead of the ereg( ) function.

TABLEX
SEARCHING BY EXACT MATCH

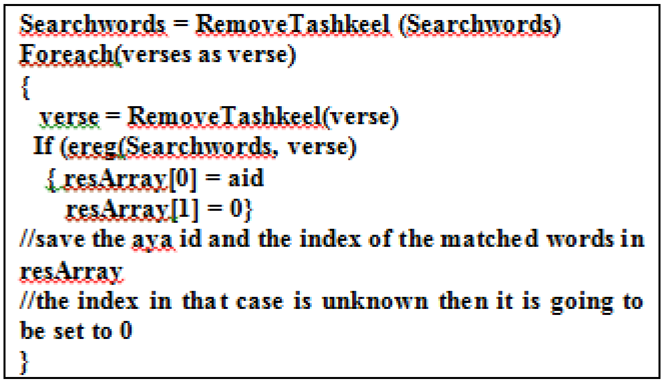


```
Foreach(verses as verse)
{
  verse = RemoveTashkeel(verse)
  verseWord = explode(“ “, verse)
  For (i=0 ;i<count(verseWord)  ; i++)
    {For each(searchWords  as word)
      { If (str_compare (word, verseWord[i]) = 0)
        {index[ ] = i} //this array keeps the indices of
matched words found in the verse
      }
    If(count(index)>0)  // then a match is found
     {resArray[0] = aid
 //save the aya id and the index of the matched words
in resArray
      resArray[1] = index}
    }
}
```

The third searching type is “searching by the derived words”, so the search executes by preparing the search word as the same as the algorithm of the second searching type by replacing the str_compare( ) function with the ereg( ) function, this will results the words abstracted and the words derived from the abstracted words.

The fourth type of search is “searching typical matching” where the words must match typically, so there is no need to split the entered string into word and search them separately, the main idea of this conversion is to remove Tashkeel from the entered string then search it in the distinct verses.

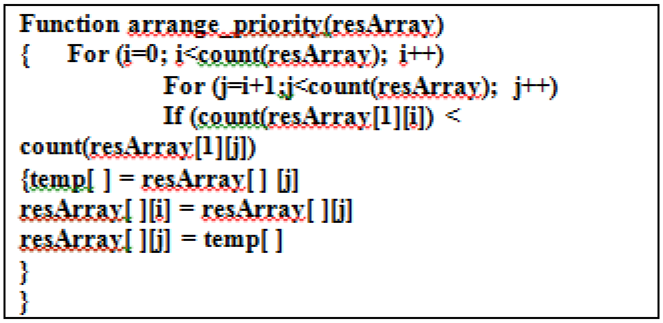
TABLE XI
SEARCHING AFTER TASHKEEL REMOVAL



```
Searchwords = RemoveTashkeel (Searchwords)
Foreach(verses as verse)
{
  verse = RemoveTashkeel(verse)
 If (ereg(Searchwords, verse)
   { resArray[0] = aid
     resArray[1] = 0}
//save the aya id and the index of the matched words in
resArray
//the index in that case is unknown then it is going to
be set to 0
}
```

The final step before displaying the result is checking the displaying type which offers two types of displaying (i) the most matched first, (ii) and according to the Quran sequence, the first improves the efficiency parameter of the searching, while the second improves the speed parameter of searching, since the search result is already arranged by the Quran sequence. The arrangement is according to the most matched first depends basically on the “index[ ]” array of each verse which contains the indexes of the matched words in the verse, by counting elements in this array we can know the number of matching words in the verse. Then arrange them in descending order using bubble sort algorithm as follows:-

TABLE XII
SEARCHING BY BUILDING INDICES



```
Function arrange_priority(resArray)
{     For (i=0; i<count(resArray); i++)
                For (j=i+1;j<count(resArray);  j++)
                If (count(resArray[1][i]) <
count(resArray[1][j])
{temp[ ] = resArray[ ] [j]
resArray[ ][i] = resArray[ ][j]
resArray[ ][j] = temp[ ]
}
}
```

Finally displaying the result in a table in a HTML page with linking the verses to their interpretation pages.

V.CONCLUSION AND FUTURE WORK

With the advancement of the social networks which became an essential media of communication, new web techniques are needed. In this work we investigated the usage of new web programming techniques for e-services. As a case study the work presents an Islamic portal for the Muslim society. It uses XML to hold the templates which is a suitable choice because of the flexibility it offers and the using of tree structural hierarchy. The xmlReader is used since it’s the best reading xml parser that deals with huge amounts of data, as in the “word.xml” document. Using such techniques for web content management showed very interesting results. Despite the fact that our search techniques of e-Quran (as an Arabic text) based on lexical analysis are efficient but we feel that semantics issue was not considered. Hence, for future work, ontologies of Quran should be explored and developed. Domain ontologies can be accumulated to build a conceptual model that can be queried to obtain more efficient results. In fact, semantic web has been used as platform to support Semantic Web searching techniques. Hence, building a knowledge base as domain ontology will not only enhance the searching technique but it will allow the system to make inferences.

ACKNOWLEDGMENT

We express our thanks and gratitude to the computer engineering of their support and encouragement to the inception of this work as an undergraduate BS project. We also thank our student Menna who implemented the search engine via a site using Apache and PhP language as well as other web tools.

REFERENCES

[1] Kevin T. Smith. Professional Portal Development with Open Source Tools. Wiley puplishing;2004.

[2] Robert Richards. Pro PHP XML and Web Services. 1st ed. Apress; 2006.

[3] Bettina Gräfi , IslamOnline.net: Independent, interactive, popular http://www.islamweb.net/mainpage/index.PHP

[4] IOL,“Media Kit“, http://www.IslamOnline/English /mediakit/index. shtml (accessed January 22, 2007) or “*Iclan macana*“, (accessed January 22, 2007). http://www.islamonline.net/ Arabic/MediaKit /index.shtml

[5] Alghamdi, Mansour, "Arabic and Islamic Contents on the Internet", Third National Information Technology Symposium King Abdullah Initiative for Arabic Content, Riyadh, Saudi Arabia. 06-07 March 2011.

[6] Eberhardt, Peter and Qin Xiao Jin “ISO 101: A SAS® Guide to International Dating”, Proceedings of the SAS® Global Forum 2013 Conference.

[7] Lee Babin. Beginning Ajax with PHP, From Novice to Professional. Apress; 2007.

[8] Corporate Portal Framework for Transforming Content Chaos on Intranets. Intel Technology Journal, Vol.11.

[9] Wall,Aaron,*Search Engine History*, October 3, 2007. <http://www.searchenginehistory.com/.

[10] “Lexical analysis,” *Wikipedia*, <http://en.wikipedia.org/wiki/Lexical analysis>, October 3, 2007.