# RAX System to Rank Arabic XML Documents

Hesham Elzentani
Faculty of Informatics and Computing
Singidunum University
Belgrade, Serbia

Mladen Veinović
Faculty of Informatics and Computing
Singidunum University
Belgrade, Serbia

Goran Šimić
Military Academy of the Ministry of Defense
Belgrade, Serbia

***Abstract*—This paper describes an RAX System designed for ranking Arabic documents in information retrieval processes. The proposed solution basically depends on the similarity of textual content. The model we have designed can be used for documents stored in the different formats and written in Arabic language. Due the complex lingual semantics of this language the proposed solution uses a pure statistical approach. The design and implementation are based on existing text processing frameworks and referent Arabic grammar. The main focus of our research has been the evaluation of different similarity measures used for classifying Arabic documents from different domains and different document categories based on query criteria provided by the user.**

***Keywords*—*Text similarity measures; Text classification; Processing Arabic documents***

## I. Introduction

Arabic is a widely spoken Semitic language. It has morphology, vocabulary and vowels. Like other Semitic languages an Arabic statement consists of a (Subject-Verb-Object) or (Verb-Subject-Object) chain. The Arabic word is structured by adding infixes, prefixes and/or suffixes as well as diacritics to the root. The Arabic language has 28 letters which are written from right to left, unlike Latin based languages which are written from left to right. The shape of letters changes according to their positions in the words. Arabic words are divided into nouns and verbs. Nouns include adjectives and adverbs while verbs include prepositions, pronouns and conjunctions. Nouns are masculine or feminine and singular, dual or plural. Verbs are derived from roots [1]. This will be described in further detail in the Related Works section.

In recent years the growth of Arabic content and numbers of users on the Internet has greatly increased as can be seen from the table of top ten languages in the Internet (Table I). Arabic is a widely spoken language with more than 375 million speakers and over 155 million, or over forty percent of these Arabic speaking people use the Internet. This represents nearly five percent of all the Internet users in the world. The number of Arabian speaking Internet users has grown by a factor of sixty in the last fifteen years (2000-2015). This growth in usage has outpaced the growth in information retrieval systems, summarization of Arabic text (such as documents and web pages), query processes and natural language processors [2].

TABLE I. Number of Internet Users by Language

| Top Ten Languages In The Internet | Internet Users by Language | Internet Penetration (% Population) | Users Growth in Internet (2000 - 2015) | Internet Users % of World Total (Participation) | World Population for this Language (2015 Estimate) |
|---|---|---|---|---|---|
| English | 851,623,892 | 60.9 % | 505.0 % | 26.0 % | 1,398,277,986 |
| Chinese | 704,484,396 | 50.4 % | 2,080.9 % | 21.5 % | 1,398,335,970 |
| Spanish | 245,150,733 | 55.5 % | 1,248.4 % | 7.5 % | 441,778,696 |
| Arabic | 155,595,439 | 41.5 % | 6,091.9 % | 4.8 % | 375,241,253 |
| Portuguese | 131,615,190 | 50.0 % | 1,637.3 % | 4.0 % | 263,260,385 |
| Japanese | 114,963,827 | 90.6 % | 144.2 % | 3.5 % | 126,919,659 |
| Russian | 103,147,691 | 70.5 % | 3,227.3 % | 3.2 % | 146,267,288 |
| Malay | 93,915,747 | 32.7 % | 1,539.0 % | 2.9 % | 286,937,168 |
| French | 92,265,199 | 23.9 % | 669.0 % | 2.8 % | 385,389,434 |
| German | 83,738,911 | 87.8 % | 204.3 % | 2.6 % | 95,324,471 |
| TOP 10 LANGUAGES | 2,576,501,025 | 52.4 % | 768.2 % | 78.8 % | 4,917,732,310 |
| Rest of the Languages | 693,989,559 | 29.6 % | 980.6 % | 21.2 % | 2,342,888,808 |
| WORLD TOTAL | 3,270,490,584 | 45.0 % | 806.0 % | 100.0 % | 7,260,621,118 |

An Arabic word has different forms of syntax and morphologies with different meanings. Grammatically, documents contain different forms of words including derivations. This causes problems in text processing, document summarization and information retrieval systems. Furthermore, there is a high level of information loss during the processes of querying, document summarizing and information retrieval, especially with large documents, as information loss is directly proportional to the size of documents during these processes.

This paper describes an RAX System which is designed for ranking Arabic documents stored in the different formats in information retrieval processes. It consists of the following sections: (1) Introduction, which introduces the Arabic language and current internet statistics; (2) Related works that have informed the research; (3) Arabic document management, which introduces the XQuery and Sedna XML database management systems; (4) Proposed solution, which describes

the processing and ranking of Arabic documents; (5) Conclusions.

## II. Related Works

There is currently a high level of interest within the research community for text processing of Arabic documents as well as queries, stemmers, ranking, keyword extraction. The retrieval of formal Arabic language, as used in media such as news domains, as well as the retrieval of Arabic dialects is among the problems that face information retrieval systems. Natural language processing of Arabic information to enable retrieval is considered in [1].

XML documents and querying XML data and databases using XQuery and XML indexing (which summarize large XML data structure into a tree) are discussed in [3] and [4].

Different Arabic text stemmers, as well as constructed Arabic stopwords lists used in information retrieval systems, are described in [5] and [6].

Stemming methodologies and query terms affect the information retrieval systems according to the word and stem. In contrast, term importance can be computed according to term frequency and inverse document frequency as described in [7].

The use of similarity measures in a vector space model, according to term frequency (TF) and inverse document frequency (IDF) of documents and structural of terms, is described in [8].

Automatic keyword extraction according to candidate keywords (that are extracted from a document and selected based on term frequency of words within these documents), word degree and ratio of degree to frequency are covered in [9].

Arabic natural language processing techniques have used linguistic resources such as Corpora and Lexicon to develop parser and POS-tagger. This has enabled the creation and evaluation of a framework for use in Islamic sciences written in the Arabic language. This framework could adapt the theories, resources, tools and applications of other NLPs such as English and French as described in [10].

Three vector space models (Cosine, Dice and Jaccard coefficients) for classifying Arabic text using the K-Nearest algorithm and the IDF term are compared in [11].

Finally, there is currently a high level of activity in the production of tools that provide automatic annotation and translation of Arabic texts. The linguistic difference of the Arabic language to western culture languages results in complexity of implementation. In [NP Subject Detection in Verb-Initial Arabic Clauses] the focus is on the words-in-sentence ordering problem and the different way Arabic phrases are formed. For example the sentence in Fig. 1, which in English is ordered from left-to-right compared to the Arabic phrase which is ordered right-to-left, illustrates the ordering problem [12].

### A. Standard Arabic reference background

The following is a brief introduction to standard Arabic based on [5] and [13].

The Arabic language is sematic language. It consists of masculine and feminine and includes grammatical cases (nominative, genitive and accusative) as well as morphology. Arabic nouns in the nominative case have a root (***stem***) which is the standard word in a list or the ***base form*** in a dictionary. For instance **خريطة** means ***a map***. The definitive noun of ***a map*** is created by adding the prefix article **ال** to the beginning of the noun to create the feminine Arabic word **الخريطة** which means ***the map***. One can also attach a preposition such as **ل** (*to)* or **بـ** (*by)* to the front of the definitive article of the noun. Thus the masculine plural of the Arabic word **بالخرائط** means by ***the maps***.

A possessive pronoun can take prefixes and suffixes. For example the Arabic word **بسيارتي,** meaning ***by my car,*** could be resolved into **ب + سيارة + ي** (remembering that writing in Arabic is in the opposite direction to western culture languages). This contains a prefix **ب,** meaning ***by,*** and a pronoun suffix **ي,** meaning ***my***.

In the Arabic language plural has a regular (sound) plural form and irregular (broken) plural form. To create the sound plural for feminine nouns one adds suffix **ات** while for masculine nouns one adds **ون** in the nominative and **ين** in the genitive and accusative. For example the Arabic word **مدرّسة** means ***teacher*** (feminine). The plural is **مدرّسات** and means ***teachers*** (feminine) in nominative, genitive and accusative. The Arabic word **مدرّس** means ***teacher*** (masculine). The plural in the nominative is **مدرّسون,** which means ***teachers*** (masculine), while the plural in the genitive is **مدرّسين ,** which also means ***teachers*** (masculine).

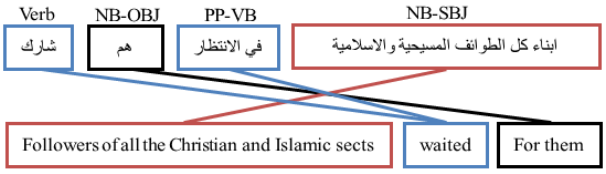


Fig. 1. Phrase reordering [12]

Moreover, the Arabic word **رجل** means ***man.*** The broken plural is **رجال** meaning ***men***, which is created by adding infix **ا**. The plural form of the noun **غرفة** , meaning ***room*** is **غرف** , meaning ***rooms***, which is created by stripping out the suffix **ة**. Another familiar example of broken plural is the Arabic word **امرأة** , meaning ***woman***, while the plural is a completely different word **نساء** , meaning ***women***.

Root is the main characteristic of Arabic language. Every root has many derivative forms. So regarding the problems discussed above, Arabic text in documents must be stemmed to get the root for every word in the text, and then rank these documents (stemmed text) using similarity measures.

### B. Preliminaries to XML trees and paths

Arabic documents are written within Arabic character encoding formats such as ISO 8859-6, Windows-1256 and UTF-8. Listing 1 shows the XML tree of an Arabic document and its translation. Hierarchically structured XML documents are the result of these transformations. A document tree consists of a set of nodes which form the root of a tree [14,15,16] and a set of edges including attributes, tags and strings (#PCDATA).

XML is a tree $T = (r_T, N_T, E_T, F_T)$, where $N_T \subseteq \mathbb{N}$. This means that every element of $N_T$ is also an element of a natural numbers nodes set. $r_T \in N_T$ is root of $T$, which is an element of $N_T$, $E_T \subseteq N_T \times N_T$. This means that all element of $E_T$ are elements of the set of edges. $F_T : N_T \mapsto \alpha$ means that the function $F_T$ maps the element $N_T$ to $\alpha$, where $\alpha$ is *attributes* $\cup$ *tags* $\cup$ *strings*.

An *XML path p* is a sequence from the tree root element to a specific node, which is $p = s_1.s_2....s_m$ symbols of nodes in $\alpha$, where $s_1$ is the tag name of root element and $s_m$ is a tag name of the specific node including attributes and strings. An XML path has two types, the incomplete path which is *tag path* and the *complete path* including $\alpha$. This paper will focus on the complete path #PCDATA (string) content.

### C. Text similarity measures

There are many methods for measuring text similarity according to query and document terms such as Dice's coefficient and Cosine similarity. The following is a brief introduction to these methods.

Dice's coefficient, defined in [17], is a statistical method used to measure the similarity between two sets or two strings, or to measure the similarity between queries and documents in terms of common n-grams. An n-gram is an adjacent section of letters in the string. Dice's coefficient is given in "(1)". The similarity values vary between 0 and 1.

$$\text{Dice}(Q, D) = \frac{2 \times \text{n-grams}(Q) \cap \text{n-grams}(D)}{\text{n-grams}(Q) + \text{n-grams}(D)} \quad (1)$$

where *n-grams(Q)* are a multi-set of letter n-grams in Query and *n-grams(D)* is a multi-set of letter n-grams in Document.

Furthermore, the main idea is breaking a string to n-grams. For example, the string "right", the set of bigrams would be {"ri", "ig", "gh", "ht"}. Likewise, the string "write" would break down into {"wr", "ri", "it", "te"}.

However, after bigrams have been created, the "(1)" can be applied. So, the set Q = {"ri", "ig", "gh", "ht"}, then |Q| = 4 and D = {"wr", "ri", "it", "te"}, then | D | = 4. The intersection of the bigram sets (Q ∩ D) is {"ri"}, only one element exists in the set. The union of the bigram sets (Q ∪ D) is {"ri", "ig", "gh", "ht", "wr", "it", "te"}, only seven elements exit in the set. So, according to "(1)", the similarity measurement between "right" and "write" is $\frac{2}{7}$.

Listing 1. XML tree of Arabic document and its translation.

```
<?xml version="1.0" encoding="UTF-8"?>
<articles>
  <article id="25">
    <title>التكاليف الاقتصادية للمشكلات البيئية وأهم طرق التقييم البيئي المستخدمة</title>
    <author>سلمى عائشة كيحلي, سليمة غدير أحمد, يوسف قريشي</author>
    <subject>بيئة</subject>
    <keywords>بيئة, المشكلات البيئية, التلوث</keywords>
    <contents>
      الملخص: ينطوي التطور الاقتصادي والاجتماعي على تكما ينعكس على البيئة ...
    </contents>
  </article>
</articles>
```

```
<?xml version="1.0" encoding="utf-8"?>
<articles>
  <article id="25">
    <title> The economic costs of environmental … </title>
    <author> Salma Aisha Kehli, Salema Ghadeer Ahmad, Yousef Qureshi </author>
    <subject> environment </subject>
    <keywords> environment, environmental problems, pollution </keywords>
    <contents> Abstract: Involves economic and social … </contents>
  </article>
</articles>
```

A collection of XML documents can be represented by a vector space model in which each document is represented by a vector of terms and their weights. A query (an expression that requests information from database) is represented as terms with weights to represent the importance of query terms. Term frequency (TF) and Inverse document frequency (IDF) are used for the weighting of terms [7,8,18,19]. This commonly used statistical measure uses:

- The frequency of a term *j* in a document *i* ($tf_{i,j}$)
- The frequency of the term *j* in the whole collection ($df_j$)
- The inverse document frequency of term *j* in document *i* ($idf_j$)

Equation (2) gives the inverse document frequency of term *j* in the collection and "(3)" gives the weight of the term *j* in the document *i*.

$$idf_j = \log_e\left(1 + \frac{N}{df_j}\right) \quad (2)$$

$$w_{i,j} = tf_{i,j} \times idf_j \quad (3)$$

where *N* denotes the number of documents in the collection.

Term weights using TF IDF for measuring the similarity between Query and Document.

Cosine similarity is used to calculate the angle between Query and Document. If a vector is considered in a *V*-dimensional Euclidean space, the angle between Query and Document represents their mutual similarity. A smaller angle means greater similarity. Equation (4) defined the similarity between a document $D_i$ and a query *Q*.

$$sim(Q, D_i) = \frac{\sum_{j=1}^{V} w_{Q,j} \times w_{i,j}}{\sqrt{\sum_{j=1}^{V} w_{Q,j}^2 \times \sum_{j=1}^{V} w_{i,j}^2}} \quad (4)$$

where $w_{Q,j}$ is weight of query term *j*, and $w_{i,j}$ is weight of term *j* in document *i* as mentioned in "(3)", (Example of cosine similarity is illustrated in section 4.1 and 4.2).

## III. Arabic Document Management

Arabic documents represented in different formats are used as information resources. They are structured in different ways and for information retrieval purposes their content should be preserved regardless of the processing necessary for information retrieval. Therefore they must be stored and manipulated in a non-relational content management system. XML data management systems, as well as filtering technologies as XPath and XQuery, were recognized as being suitable for this purpose.

### A. XQuery

The RAX System developed during the course of this research ranks documents based on criteria given by the end user (or client application). XQuery is utilized for this purpose as the documents are represented in XML format. XQuery is a query language for querying collections of XML documents as introduced by the World Wide Web Consortium (W3C). XQuery uses XPath expression to address specific nodes on XML document including FLWOR expression (FOR, LET, WHERE, ORDER BY and RETURN) [20]. The example in listing 2 illustrates XQuery expression for getting terms that appear in the text by using iteration (FOR clause) and criteria (term frequency > 0 in WHERE clause).

XQuery runs many operations to access XML documents including selecting information based on identified standards, filtering, seeking, joining data from multiple documents or collections, sorting, clustering, restructuring XML data into another XML structure and performing arithmetic calculations [21].

### B. Sedna XML database management system

Many database management system producers offer support for the management of data stored in XML formats (IBM DB2, MS SQL Server, Oracle DB, PosstgreSQL, etc.). For example Sedna XML DBMS can be used for managing XML documents [22]. Sedna is an XML DBMS with full database functionality. Sedna gives flexible XML processing capability including W3C XQuery implementation and integration of XQuery with full-text searching. The Sedna client application programming interfaces (APIs) can access databases of the Sedna DBMS and treat data using XML database query languages (e.g. XPath and XQuery). API enables access to Sedna from other client systems programmed in high-level languages such as Java APIs.

## IV. Proposed Solution

The aim of this paper is to find a suitable solution to the problems mentioned in the introduction section i.e. the rapid growth in demand for Arabic language content; the complexity of the language and its differences to existing tools based on western culture/Latin based languages causing problems in text processing, document summarization and information retrieval systems; and the high information loss rates especially for larger documents. To achieve this following steps were followed:

- Collect data from Arabic PDF documents from different domains and categories (agriculture, sciences, geography, ecology, engineering, development, energy, industry, administration, accounting, education, information technology and computers).
- Use an Arabic normalizer to normalize the Arabic text extracted from Arabic PDF documents.
- Remove Arabic stopwords from normalized text.

Listing 2. Example of XQuery with FLWOR expressions.

```
for $article_text in // contents /text()
let $tokenized_text := tokenize($article_text, ' environmental ')
let $term_freq:= count($tokenized_text) where $term_freq>0
return $term_freq
```

- Use an Arabic stemmer to stem every normalized word in the text and get the base form (dictionary word).
- Create XML documents according to the stemmed text because XML is used to exchange and represent semi-structured data on the Internet.
- Load the XML documents to the XML database management system.
- Apply queries and weight and rank XML documents to define an appropriate concept of similarity between the XML documents and queries.

To carry out the above steps the *RAX system* was developed and used to rank Arabic documents via an XML database

management system. Basically, processing of Arabic documents is performed in two stages; document preparation stage and implementation stage. Fig. 2 illustrates the overall system architecture and dataflow through its steps. Portable document format is used as the input format due to fact that there are many tools and functional libraries designed for conversion of different document formats to PDF. These include Apache OpenOffice [23] and documents4j library [24]. Text extraction from PDF is also well supported e.g. via Apache PDFBox [25] and iText library [26].

Document preparation is represented with a dataflow from PDF input to XML DBMS while the document implementation stage is represented with a dataflow from XML DBMS to the user. More details about these two stages are given in the next two sub-sections.

### A. Document preparation stage

The first stage begins with the loading of PDF documents into a PDF Box library. This process is described in the following seven steps:

*1)* Apache PDFBox is used to extract of pure text and metadata from PDF files. PDFBox represents a class library written in Java and used in many advanced content management tools (e.g. Alfresco, Lucene, Apache Tika and REWOO Scope). It is an open source tool for dealing with PDF documents. The *RAX* system used the Apache PDFBox library to test a set of 100 Arabic PDF documents from different domains and categories [27].

*2)* An Arabic normalizer performs the normalization process in which Arabic diacritics, punctuation, non-letters and stretching letters are removed and different versions of a letter are converted into the standard letter. For instance the letters أ , إ , آ , ٱ are all forms of the letter ا (letter ***A*** in the English language). These various forms would each be converted into letter ا. Other examples are the normalization of the letters ى and ه which are transformed into ي and ة A diacritic word مَدْرَسَةٌ meaning ***school*** will normalize to مدرسة without diacritics. The stretched word كِتَـــــــاب meaning ***book*** will normalize to كتاب without stretching. The word أَحَمَدْ contains diacritics and one form of letter ا. This normalizes to احمد with the standard form of letter ا and without diacritics.

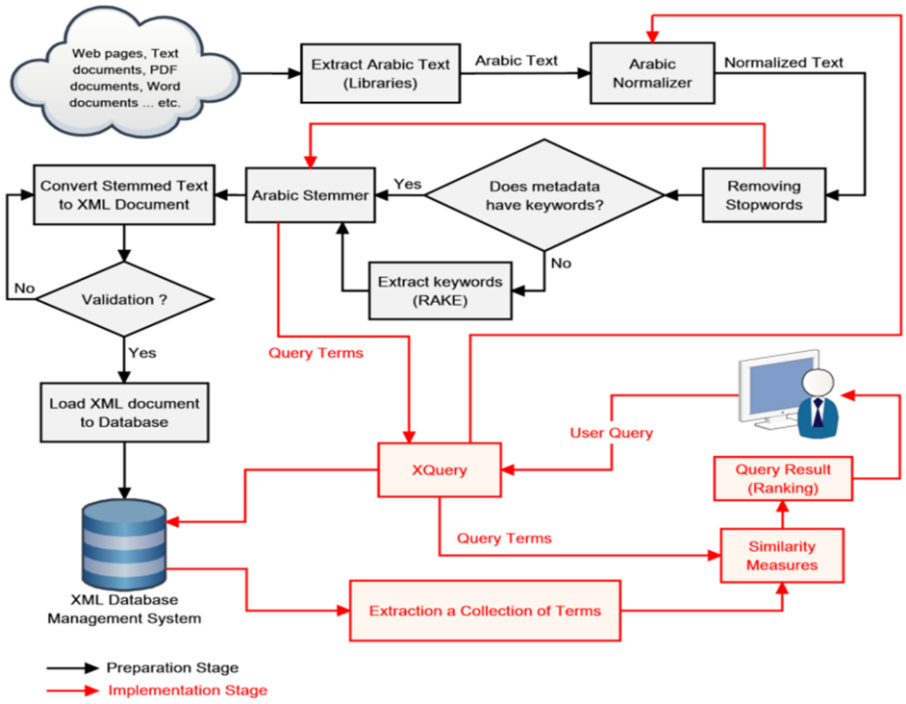


Fig. 2. RAX system model

*3)* The *RAX* system then removes Arabic stopwords, such as و, ان and في. A list of Arabic stopwords has been created consisting of 168 stopwords including pronouns and prepositions [13].

*4)* Following normalization and removal of stopwords. If there are no keywords in the document's metadata the RAX system uses rapid automatic keyword extraction (RAKE) technology to extract keywords from text. RAKE contains a list of stopwords, phrase and word delimiters that is used to identify candidate keywords – a series of words by priority of occurrence in the text. Each candidate keyword is scored according to the ratio of word degree to word frequency. The top scoring candidates are selected as keywords, which calculated as $\frac{1}{3}$ number of words [9].

*5)* Since it is difficult to process Arabic language in summarization and information retrieval due to its complex

morphology, an Arabic stemmer is used to reduce derivational forms of a word to a stem or a root word (base form). Each root gives rise to many different words, such as nouns, adjectives, and verb stems. For example the words **مَكْتَبْ** (maktab) *office*, **كُتُبْ** (kutub) *books*, **كِتَابْ** (kitAb) *book*, **كَتَبَ** (kataba) *he wrote*, and **نَكْتُبُ** (naktubu) *we write* all come from the root **كتب** (ktb). The *RAX* system uses its own stemmer to strip off prefixes such as **ل ل ا و س ب ي ن م ت** and suffixes such as **ف هما تما كما ان ها و ا تم كم تن كن نا تا ما ون ين هن هم ته تي ني ن ك ه ة ت ا ي ات.** To illustrate the processes mentioned above, fragments of three documents are used:

- **$D_1$**: "الأنظمة الحكومية هي الأنظمة الموثوق بها والتي تحتفظ بمعلومات عن المواطنين ..." – "*Government systems are the trusted systems that hold information about citizens….*"
- **$D_2$**: "قواعد البيانات هي الجوهر لنظم المعلومات ..." - "*Data bases are the core of information systems. ….*"
- **$D_3$**: "قواعد البيانات في أنظمة الحكومة التي تحتفظ بمعلومات عن المواطنين وهم جوهر انظمة المعلومات ..." - "Data bases in government systems hold information about citizens and they are the core of these information systems. …."

In the first step the system eliminates stop words and the sentences are modified as follows:

- **$D_1$**: الأنظمة الحكومية الأنظمة الموثوق تحتفظ بمعلومات المواطنين - *Government systems trusted systems hold information citizens*.
- **$D_2$**: قواعد البيانات الجوهر لنظم المعلومات - *Data bases core information systems.*
- **$D_3$**: قواعد البيانات أنظمة الحكومة تحتفظ بمعلومات المواطنين جوهر انظمة المعلومات - *Data bases government systems hold information citizens core information systems*.

In the next step (stemming) all of the words are transformed into normal form. In this way the sentences are put into their final form as follows:

- **$D_1$**: نظم حكم نظم وثق حفظ علم وطن - Government system trust system hold information citizen.
- **$D_2$**:قعد بين جوهر نظم علم - Data base core information system.
- **$D_3$**: قعد بين نظم حكم حفظ علم وطن جوهر نظم علم - Data base government system hold information citizen core information system.

*6)* Document Creation; XML is a simple textual data, which supports different Unicode standards for different languages and well as benefiting from simplicity and usability over the Internet. XML syntax is widely used as a default format to represent data structure and create documents e.g. in Microsoft Office, OpenOffice.org, and web services. By this stage RAX has initialized the XML document and converted normalized stemmed Arabic text originating from PDFs to well-formed Arabic XML documents using Java API for XML Processing [28] and Simple API for XML [29]. Listing 3 shows the fragment of the first document ($D_1$) in this step represented in XML form.

Listing 3. XML form of fragment $D_1$.

```
<?xml version="1.0" encoding="UTF-8"?>
<articles>
   <article id="1">
      <title>الانظمة الحكومية</title>
      <author>محمد نصر</author>
      <subject>سياسة</subject>
      <keywords>نظام الحكم</keywords>
      <statistics>
         <percentstemmed>100.0</percentstemmed>
         <stemingtime>0.30</stemingtime>
         <stemmedwords>7</stemmedwords>
         <nonstemmedwords>0</nonstemmedwords>
         <stopwords>4</stopwords>
         <punctuationwords>0</punctuationwords>
         <nonletterwords>0</nonletterwords>
         <totalwords>11</totalwords>
         <totalstemmedwords>7</totalstemmedwords>
      </statistics>
      <stemmedtext>وطن حفظ علم نظم حكم نظم وثق </stemmedtext>
      <notstemmedwords/>
   </article>
</articles>
```

The XML database management systems enable storage of XML documents and transfer of data between relational databases. These documents can be queried, transformed, transported and returned to a calling system. So after the documents are preprocessed and XML formed they are ready to be stored in an XML DBMS The Sedna DBMS, which has full ability of database services and gives flexible XML processing facilities including W3C XQuery accomplishment with full-text search, is used. This is the end of preparation stage and *RAX* system is ready for implementation.

### *B. Implementation stage*

One of the obvious facts about information retrieval systems, as opposed to sorting and searching algorithms, is that the more documents are stored into the database the better it performs. Next is a description how the system works during implementation:

*1)* When the end user enters a query the *RAX* system performs its processing in the same way as with documents (normalization, the removing of stopwords and stemming). As a result the query expression is transformed into vector of terms.

*2)* Next the *RAX* system executes XQuery on the document base in the XML DBMS (Sedna DBMS) expression which includes the vector of query terms. XQuery uses XPath syntax for accessing different nodes of

XML documents. A set of XML documents is returned as a result of the query. Listing 4 shows an XQuery expression; this query returns a collection of documents including the term frequency for each document that contains the query term.

Listing 4. XQuery expression used by *RAX* system.

```
let $query_term :='نظم'
for $document_terms in //article, $id in $ document_terms/@id
let $output_terms := tokenize($document_terms/stemmedtext/text(), $query_term)
let $freq := count($output_terms)where $freq>0
return <document id="{$id}">{<frequency>{$freq}</frequency>}</ document>
```

*3)* To rank documents the *RAX* system calculates weights for each particular term in the document. Term frequency and inverse document frequency are used for this purpose ("(2)" and "(3)"). This means that the vector space model in which the documents are transformed is enriched with additional information: each document's vector is represented by an array of term-weight pairs. The user query is processed in the same way. Final comparison between these two is performed using cosine similarity as a measure of the documents' ranking ("(4)"). The following example shows how the documents' samples fragments (described in section 4.1) are used in this process. Transformation in the improved vector model is the most crucial and processor intensive phase. After this each fragment of document being considered for ranking is represented with two vectors. The original XML document consists of a vector of terms. The terms are collections of words and each word is represented by its weight (TF*IDF) value. In this way the documents are represented with two vectors. Original words are filtered and transformed into normal form and for convenience are labeled terms. For clarity, TF and IDF are represented separately i.e. ($term_1$, $tf_1$, $idf_1$), … , ($term_n$, $tf_n$, $idf_n$), where *n* is the full number of terms in the document set. Thus there are 9 different terms for our example and the vector space model (VSM) for each document should contain these. TF is represented by row frequency, which represents the number of occurrences of a specific term in the document, and IDF is calculated according to "(2)". See table II.

TABLE II. TF AND IDF CALCULATIONS FOR SAMPLES USED IN EXAMPLE

| **Document: $D_1$** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ***Term*** | ***نظم System*** | ***حكم Government*** | ***وثق Trust*** | ***حفظ Hold*** | ***علم Information*** | ***وطن Citizen*** | ***بين Data*** | ***قعد Base*** | ***جوهر Core*** |
| TF | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| IDF | 0.30102 | 0.39794 | 0.60206 | 0.39794 | 0.30102 | 0.39794 | 0 | 0 | 0 |
| $W_{D1}$ | 0.60204 | 0.39794 | 0.60206 | 0.39794 | 0.30102 | 0.39794 | 0 | 0 | 0 |
| **Document: $D_2$** | | | | | | | | | |
| ***Term*** | ***نظم System*** | ***حكم Government*** | ***وثق Trust*** | ***حفظ Hold*** | ***علم Information*** | ***وطن Citizen*** | ***بين Data*** | ***قعد Base*** | ***جوهر Core*** |
| TF | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |
| IDF | 0.30102 | 0 | 0 | 0 | 0.30102 | 0 | 0.39794 | 0.39794 | 0.39794 |
| $W_{D2}$ | 0.30102 | 0 | 0 | 0 | 0.30102 | 0 | 0.39794 | 0.39794 | 0.39794 |
| **Document: $D_3$** | | | | | | | | | |
| ***Term*** | ***نظم System*** | ***حكم Government*** | ***وثق Trust*** | ***حفظ Hold*** | ***علم Information*** | ***وطن Citizen*** | ***بين Data*** | ***قعد Base*** | ***جوهر Core*** |
| TF | 2 | 1 | 0 | 1 | 2 | 1 | 1 | 1 | 1 |
| IDF | 0.30102 | 0.39794 | 0 | 0.39794 | 0.30102 | 0.39794 | 0.39794 | 0.39794 | 0.39794 |
| $W_{D3}$ | 0.60204 | 0.39794 | 0 | 0.39794 | 0.60204 | 0.39794 | 0.39794 | 0.39794 | 0.39794 |

The next step is to determine the cosine similarity between the query and the previous collection which is represented in table II. Let us consider a query which contains two words: **نظم علم - *information system*** *(*in stemming form*)*. Table III shows that the query has transformed into VSM.

TABLE III. TF AND IDF CALCULATIONS FOR QUERY

| **Query: $Q_1$**<br>**نظم علم - Information System** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ***Term*** | ***نظم System*** | ***حكم Government*** | ***وثق Trust*** | ***حفظ Hold*** | ***علم Information*** | ***وطن Citizen*** | ***بين Data*** | ***قعد Base*** | ***جوهر Core*** |
| TF | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| IDF | 0.30102 | 0 | 0 | 0 | 0.30102 | 0 | 0 | 0 | 0 |
| $W_{Q1}$ | 0.30102 | 0 | 0 | 0 | 0.30102 | 0 | 0 | 0 | 0 |

Finally, table IV shows the calculated cosine similarity according to "(4)". The document $D_3$ is the best fit to the query and the ranking is $D_3$, $D_1$, $D_2$.

TABLE IV. COSINE CALCULATIONS

| Cosine Similarity | |
|---|---|
| Cosine($Q_1$,$D_1$) | 0.52230 |
| Cosine($Q_1$,$D_2$) | 0.45894 |
| Cosine($Q_1$,$D_3$) | 0.64904 |

*4)* Practical Evaluation and Comparison

Regarding the system complexity and hardware limitations the collection of 100 of random documents was found to be optimal for the different scenarios used in the research. The random documents are obtained from different categories [27]. The next table (table V) illustrates these categories:

TABLE V. CATEGORIES OF INTEREST

| Category | No. of documents |
|---|---|
| General reference | 1 |
| Culture and the arts | 2 |
| Geography and places | 1 |
| Health and fitness | 3 |
| Mathematics and logic | 5 |
| Natural and physical sciences | 9 |
| People and self | 11 |
| Philosophy and thinking | 3 |
| Society and social sciences | 21 |
| Technology and applied sciences | 44 |
| Total | 100 |

First of all, the RAX stemmer is compared with others. Fig. 3 represents the comparison of summarization process over the collection between RAX stemmer and other stemmers, such as Khoja and Light10. It's clear from Fig. 3 that RAX system is much powerful than the others, because RAX System has used wider list of stopwords.

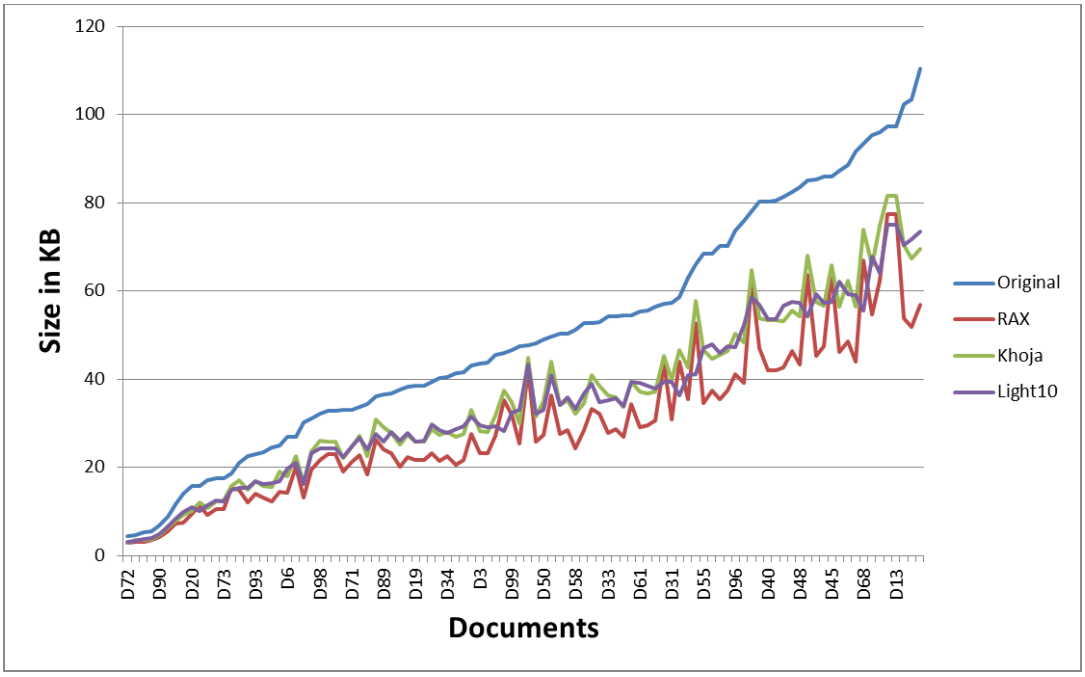


Fig. 3. Summarization comparison between RAX stemmer and other stemmers in ascending order according to original XML documents size

Following we mentioned just two queries from Computer Science and Ecology domains. These queries are used in the experiment in order to cover all of the documents. The *RAX system* is used for measuring similarity by using TF, IDF and cosine similarity as previously described as well as compares results with Dice coefficient measurements:

- $Query_1$= {نظم المعلومات – *Information Systems*}, after stemming process has taken place, $Query_1$={نظم علم– Information System}. $Query_1$ has two terms; the (*Information*) term which occurred once (TF=1) and the (*System*) term which occurred also once (TF=1). So the inverse document frequency and the weight of $query_1$ were determined from "(2)" and "(3)". As a result of $query_1$ we have found 77 documents contain the term (*Information*), and 92 documents contain the term (*System)*. The cosine similarity measures are calculated between $query_1$ terms and document terms according to "(4)". The top ranked documents which contain both terms are $D_{78}$, $D_{84}$, $D_{46}$, $D_{23}$, $D_{59}$, $D_{18}$ and $D_{61}$ (see Fig. 4).

  RAX system has used N-grams=3 to calculate Dice Coefficients similarity measures according to "(1)" because the whole roots in the Arabic language have three letters. Fig. 5 represents the percentage of similarity measurements. The most ranked documents are $D_{51}$, $D_{87}$, $D_{20}$, $D_{46}$, $D_1$, $D_5$ and $D_9$ respectively.

- $Query_2$= {التنمية البيئية المستدامة - *Sustainable Environmental Development* }, after stemming

process has taken place, $Query_2$={نمي دوم بيئية - *Sustain Environment Develop* }. Query2 has three terms; the (*Sustain*) term which occurred once (TF=1), the (*Environment*) term which occurred once (TF=1) and the (*Develop*) term which also occurred once (TF=1). So the inverse document frequency and the weight of $query_2$ were determined from "(2)" and "(3)". We have found 46 documents contain the term (*Sustain*); no document contains the term (*Environment*), and 82 documents contain the term (*Develop*). As none of the collection match the term (*Environment*) it is impossible to calculate IDF due to the denominator dfj being zero, so the RAX system has excluded the term (*Environment*) from further consideration. The cosine similarity measures are calculated between $query_2$ terms and document terms according to "(4)". We conclude that the RAX system will exclude terms which are not matched. The top ranked documents which contain both terms are $D_{71}$, $D_{44}$, $D_6$, $D_{36}$, $D_1$, $D_9$ and $D_{17}$ (see Fig. 6).

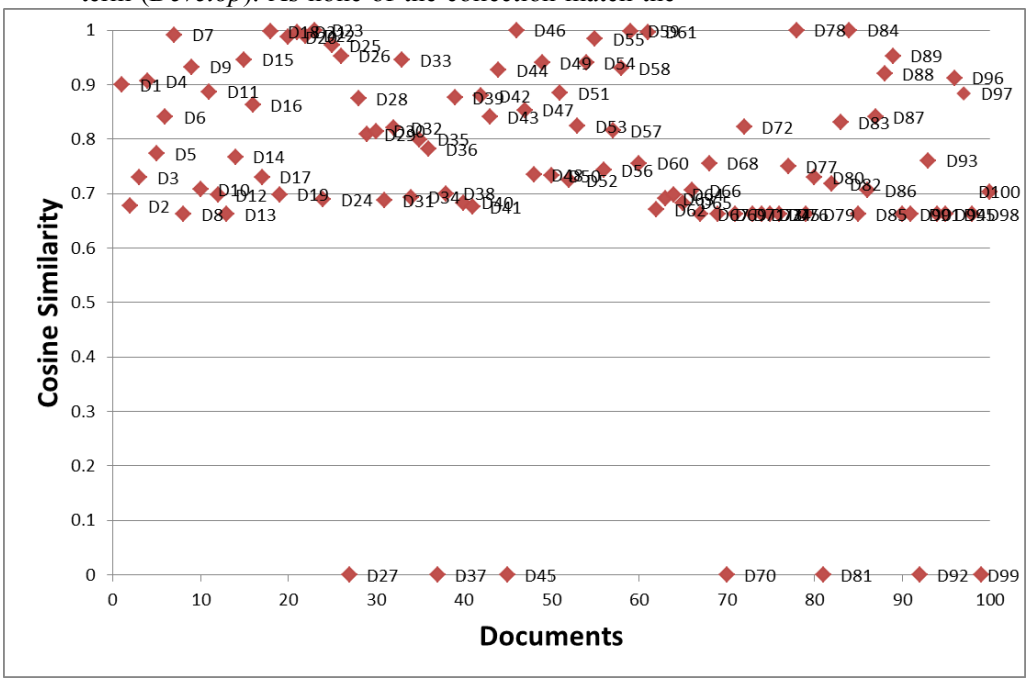


Fig. 4. Similarity measures of query

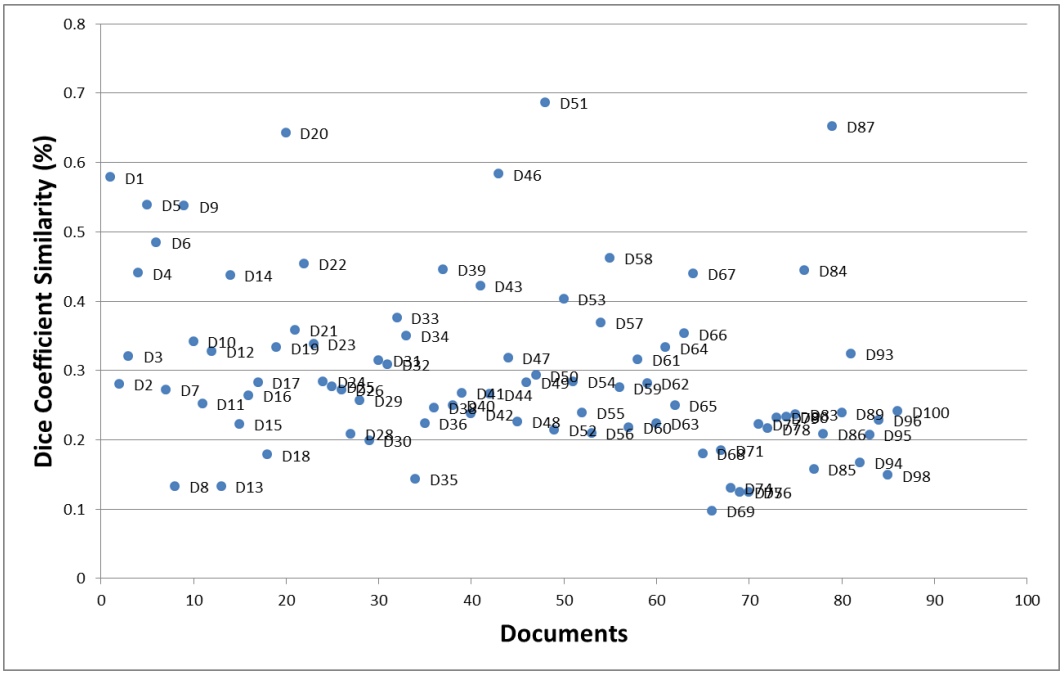


Fig. 5. Dice coefficient measures of documents that match $Q_1$ terms

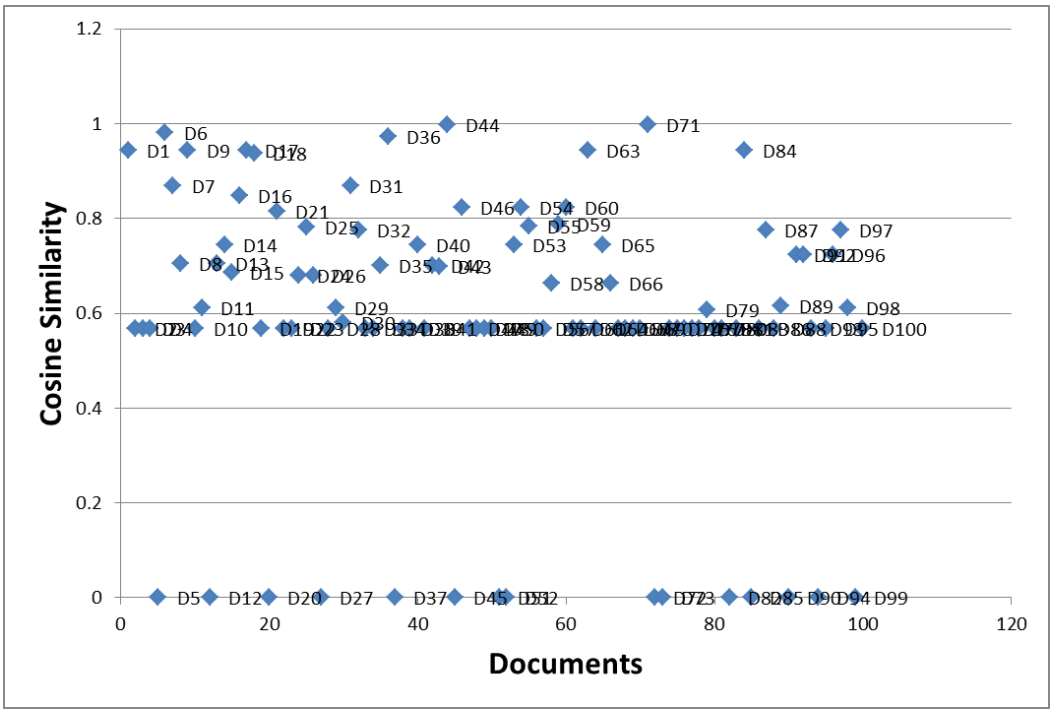


Fig. 6. Similarity measures of query$_2$

RAX system also has used N-grams=3 to calculate Dice Coefficients similarity measures according to "(1)". Fig. 7 represents the percentage of similarity measurements. The most ranked documents are $D_{87}$, $D_1$, $D_{67}$, $D_{84}$, $D_6$, $D_{58}$, and $D_9$ respectively.

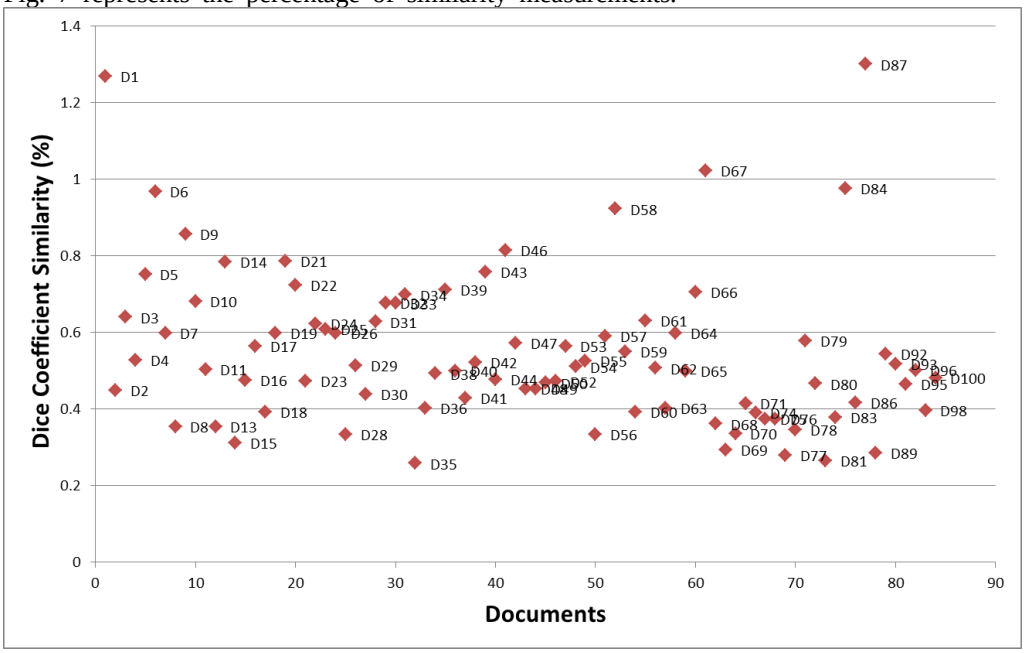


Fig. 7. Dice coefficient of the resulted documents and the $Q_2$ as three terms

From above results, we noticed that there was difference in the documents ranking between Cosine and Dice because cosine similarity depends on term frequency and inverse document frequency. In contrast, Dice coefficient similarity depends on n-grams, every time the n-grams is changed the ranked document will change too. So, Cosine similarity is much accurate than Dice similarity.

## V. CONCLUSION

This paper described the RAX System which has been designed for ranking Arabic documents based on content similarity. Our model was applicable to documents stored in different formats and written in Arabic language. The design and implementation were based on existing text processing

frameworks and referent Arabic grammar. The main focus of the research was on evaluating different similarity measures used for classifying Arabic documents from different domains and different categories.

In the preparation stage the *RAX* system was used to process Arabic text taking in account the character encoding for the Arabic language (UTF-8, Windows-1256 etc). In the implementation stage the *RAX* system managed XML documents via an XML database management system using Xpath and XQuery languages. The *RAX* system uses cosine similarity to measure the similarity metric in n-dimensional space. This is based on the finding that when two vectors are similar in rate and direction from the origin to their end points, they will be close to each other in the vector space, with a small angular separation, and vice versa. The cosine value lies between 1 and -1. Therefore the cosines of small angles are close to 1, which means high similarity, while the cosines of large angles are close to –1, which means low similarity.

The preparation stage of the processing of Arabic text was established in 4 steps: extraction of full text from documents; normalization (remove diacritics, remove non-letters and remove punctuation marks); removal of stopwords from the normalized text and stemming (remove prefixes, remove suffixes and finally extract roots or stems words). The well-formed Arabic XML document was created from the stemmed text and loaded into XDBMS which manages end user queries over a collection of XML documents. The Arabic text in queries was processed in 3 steps: normalization, removal of stopwords and stemming (Implementation stage).

When were no documents in the collection which match one of the terms i.e. *Environment.* In this case it was impossible to calculate IDF due to the denominator $df_j$ being equal to zero. In this case the *RAX* system excluded the term *Environment* from further consideration.

We conclude that the Arabic text was fully represented in the processing of Arabic documents.

There was a proportional relationship between the number of terms of a query and its result. The *RAX* system excludes terms which are not matched. Some factors such as the position of nodes in the XML tree and the query expressions (structure of expressions) could affect the operation of the *RAX* system. System performance could be improved by changing the type of stemmer.

There are two main advantages of the *RAX* system. Firstly, the query results are more comprehensive and wider when using the roots of words or stems. Secondly, the similarity measures are calculated after the completion of the query process.

As regards future work the RAX system could be improved in various ways. We plan to work on making it more efficient. This will mean that the stemmer will need to be improved and enhanced in capabilities and effectiveness to deal with the huge volume of Arabic roots in large data sets (stopword list, compatibility between prefixes and suffixes in stemming process, etc). We also aim to use DTD and XML schema to create XML documents as well as to enhance their summarization. Finally, we plan to upgrade the *RAX* system to find and replace any query term which has a zero term frequency.

## References

[1] K. Darwish and W. Magdy, “Arabic information retrieval,” Foundations and Trends in Information Retrieval, vol. 7(4), pp. 239-342, 2013.

[2] Internet world stats, retrieved on June 30, 2015, http://www.internetworldstats.com/stats7.htm.

[3] L. Zhang, X. Shen and W. Xiong, “The query and application of XML data based on XQuery,” 4th International Conference on Computational and Information Sciences, 2012.

[4] Q. Zou, S. Liu and W. Chu, “Using a compact tree to index and query XML data,” Proceedings of the 13th ACM International Conference on Information and Knowledge Management, pp. 234-235, 2004.

[5] A. Chen and F. Gey, “Building an Arabic stemmer for information retrieval,” 11th Text Retrieval Conference (TREC 2002), 2003.

[6] L. Larkey, L. Ballesteros and M. Connell, “Light stemming for Arabic information retrieval,” in Arabic Computational Morphology: Text, Speech and Language Technology, vol. 38, pp. 221-243, Springer, 2007.

[7] H. Abu-Salem, M. Al-Omari and M. Evens, “Stemming methodologies over individual query words for an Arabic information retrieval system,” Journal of the American Society for Information Science, vol. 50(6), pp. 524–529, 1999.

[8] T. Shlieder and H. Meuss, “Querying and ranking XML documents,” Journal of the American Society for Information Science and Technology, vol. 53(6), pp. 489-503, 2002.

[9] S. Rose, D. Engel, N. Cramer and W. Cowley, “ Automatic keyword extraction from individual documents,” in Text Mining: Applications and Theory, edited by Michael W. Berry and Jacob Kogan, 2010.

[10] M. Najeeb, A. Abdelkader and M. Al-Zghoul, “Arabic natural language processing laboratory serving Islamic sciences,” International Journal of Advanced Computer Science and Applications, vol. 5(3), pp. 114-117, 2014.

[11] J. Ababneh, O. Almomani, W. Hadi, N. El-Omari and A. Al-Ibrahim, “Vector space models to classify Arabic text,” International Journal of Computer Trends and Technology, vol. 7(4), pp. 219-223, 2014.

[12] S. Green, C. Sathi and C. Manning, “NP subject detection in verb-initial Arabic clauses,” Proceedings of the Third Workshop on Computational Approaches to Arabic Script-based Languages, 2009.

[13] K. Ryding, A reference grammar of modern standard Arabic. Cambridge University Press, 2005.

[14] A. Tagarelli and S. Greco, “Toward semantic XML clustering,” Siam Conference on Data Mining, 2006.

[15] M. Hachicha and J. Darmont , “A survey of XML tree patterns,” IEEE Transactions on Knowledge and Data Engineering, vol. 25(1), pp. 29-46, 2013.

[16] S. Flesca, F. Furfaro, S. Greco and E. Zumpano, “Repairs and consistent answers for XML data with functional dependencies,” In Z. Bellahsene, A. Chaudhri, E. Rahm, M. Rys, R. Unland (Eds), Database and XML Technologies, First International XML Database Symposium, XSym 2003, Berlin, Germany, September 8, 2003, Proceedings. Lecture Notes in Computer Science 2824, pp. 238–253, Springer 2003.

[17] G. Kondrak, “N-gram similarity and distance,” String Processing and Information Retrieval, Lecture Notes in Computer Science, vol. 3772, pp. 115-126, 2005.

[18] G. Šimić, Z. Jeremić, E. Kajan, D. Randjelović and A. Presnall, “A framework for delivering e-government support,” Acta Polytechnica Hungarica, vol. 11(1), pp. 79-96, 2014.

[19] D. Lee, H. Chuang and K. Seamons, “Document ranking and the vector-space model,” IEEE Software, vol. 14(2), pp. 67-75, 1997.

[20] W3C Recommendation, “XQuery 3.0: An XML Query Language,” 2014, http://www.w3.org/TR/xquery-30/.

[21] P. Walmsley, XQuery, O'Reilly Media Inc, 2007.

[22] Sedna, "Native XML database system," 2015, http://www.sedna.org/.

[23] Apache OpenOffice, retrieved on December 24, 2016, https://www.openoffice.org/.

[24] Documents4j library, retrieved on December 24, 2016, http://documents4j.com/#/.

[25] Apache PDFBox® - A Java PDF library, retrieved on December 24, 2016, http://pdfbox.apache.org/.

[26] iText library, retrieved on December 24, 2016, http://itextpdf.com/.

[27] Wikipedia, portal:contents/categories, retrieved on December 24, 2016, https://en.wikipedia.org/wiki/Portal:Contents/Categories.

[28] Java API for XML processing (JAXP), retrieved on December 24, 2016, https://docs.oracle.com/javase/tutorial/jaxp/.

[29] Simple API for XML (SAX), retrieved on December 24, 2016, http://www.saxproject.org.