American Journal of Applied Sciences 2 (11): 1520-1525, 2005
ISSN 1546-9239
© 2005 Science Publications

Designing and Building an Automatic Information Retrieval System for Handling the Arabic Data

[1]Ibrahiem M.M. El Emary and [2]Ja'far Atwan
[1]Faculty of Engineering, Amman Al Ahliyya University, Jordan
[2]Faculty of Science & IT, Al Balqa Applied University Amman, Jordan

Abstract: This paper aimed to design and build an Automatic Information Retrieval System to handle the Arabic data. Also, this paper presents some type of comparison between the retrieval results using the vector space model in two different indexing methods: the full-ward indexing and the root indexing. The proposed Automatic Information Retrieval system was implemented and built using a traditional model technique: Vector Space Model (VSM) where the cosine measure similarity was used. The output results indicate and show that the root indexing improved the retrieval performance more than the full-ward indexing on the Arabic documents; furthermore it reduces the size of stored data and minimizes the time of system processing.

Key words: IR, VSM, word indexing, stem indexing, root indexing

INTRODUCTION

Information Retrieval (IR) deals with the representation, storing, organizing and accessing the information items that match the user needs. The primary goal of the IR system is to retrieve all documents, which are relevant to the user query while retrieving as few non-relevant as possible. The user-task of retrieval system is to translate his information need into a query of the language provided by the system, where the query is a set of words that convey the semantics of the information need [1]. Working with IR in the Arabic language in a new area of research compared to the work done on the other languages. Arabic-IRS uses both the Arabic and English language.

Related works: Vector model is the most popular model among the research community in information retrieval. Most of this popularity is due to the long term search of Salton. Most of this research revolved around the SMART retrieval system developed at Cornell University[2]. Term weighting for the vector model has also been investigated thoroughly. YU and Salton[3] studied the effect of the term weighting in the final ranking. In[4], a comparison between three similarities (Cosine, Dice, and Jacard) for binary weight vector model was done and it is found that they produced the same ranking of the vector model. Yats and Neto[5] found out that the vector space mode retrieval system has various advantages as:

a. Its term weighting scheme improves retrieval performance.
b. Its partial matching strategy allows retrieval of documents that approximate the query conditions.
c. Its cosine ranking formula sorts the documents according to their degree of similarity to the query.

Van Rijsbergen[6] found that the major way in reducing the size of index term and achieving a high degree of relevancy for the retrieved document is using stemming techniques. Also, he found that stemming process would reduce the size of the document representation by 20-50% compared with the full words representation.

An overview of the information retrieval system: The information retrieval system is functionality consisting of input, processor and output. The input is a text or query. The output is a set of references. The processor sometimes classifies the stored information in some structured type and does the matching work between the input queries with the stored information to respond the user.

In order to handle the IRS well, we should clarify two important subjects; the first one related to the logical view of the document and the second one related to the retrieval process.

Regarding the first one, we say that modern computers are making it is possible to represent a document by its full set of words. In this case, the retrieval system adopts a full text logical view (or representation) of the documents with very large collections and high cost. Modern computers also might have to reduce the set of representative keywords. This can be accomplished by the elimination of stop words (such as articles and connectives) by using of stemming (which reduced distinct words to their common grammatical root) and by identification of noun groups (which eliminate adjectives, adverbs, and verbs). Furthermore, compression might be employed. These operations are called text operations (or transformation). Text operations reduce the complexity of the document representation and allow moving the logical view from that of a set of index.

Corresponding Author: Ibrahiem M.M. El Emary, Faculty of Engineering, Amman Al Ahliyya University, Jordan

Regarding the second one, we say that it is necessary to define the text database which usually done by the manager of the database. Then the logical view of the documentation is defined, and the database manager (using the DB manager module) builds an index of the text. The user first specifies user needs, which is then parsed and transformed by the same text operations applied to the text. Query operations might be applied before the actual query, which provides a system representation for the user need. The query processing is made possible by the index structure previously built. Before been sent to the user, the retrieved documents are ranked according to a likelihood of relevance. The user then examines the set of ranked documents in the search for useful information. At this point, he might pinpoints subset of the documents seen as definitely of interest and initiates a user feedback cycle. This modified query is a better representation of the real user need.

Models of information retrieval, weighting and indexing process: There are three classic models in the information retrieval given by: (1) Boolean model in which the documents and query are represented as sets of index terms; this model is a set theoretic. (2) Probabilistic model in which the framework for modeling documents and query representation is based on probability theory; this model is probabilistic. (3) Vector space model in which the documents and query are represented as vectors in t-dimensional space. Thus, this model is algebraic and it is of main concern of our study.

The vector model recognizes that the use of binary weights is too limiting and proposes a framework in which partial matching is possible. This is accomplished by assigning non binary weights to index term in query and in documents. These term weights are ultimately used to compute the degree of similarity between each document stored in the system and the user query. By sorting the retrieved documents in decreasing order of this degree of similarity, the vector model takes into consideration documents which match the query terms only partially.

With regard to the models of weighting, we have three methods; in the first one the term will be more important to a document if it occurs frequently in that document and the importance of that document decreases as the term is assigned to more documents in the collection. This scheme is called the inverse document frequency weight, which is also based on the availability of each term in the text. Here, the weight is equal to the frequency of occurrences of term in document and inversely proportional to the total number of documents to which each term is assigned.

The second method in based on calculation of the signal to noise ratio in analogy with Shannon's communication theory. It depends on Fj. (the frequency of term in document) and TFj (the total frequency of term j in the collection).

The third method is called to term distribution value which is defined by Slaton in[6]. This value is given as a difference between the average similarity of documents with $term_j$ deleted and the average similarity measure which is obtained by the cosine measure[6,8].

In view point of indexing process; we say that the indexing represents one of the most crucial techniques in an information retrieval system and it consists of choosing the appropriate term to represent the documents[5]. The inverted file is created to handle a quick access to retrieve documents by using index term[7]. They reduce the memory size because most processing uses only the index and the abstracts file can be left on the disk most time. There are three types of indexing process given by:

1. Word indexing
2. Stem indexing and
3. Root indexing.

In word indexing, before starting any indexing process, we must take care that there is no spelling errors in the documents by checking it one by one and making the spelling correction.

In stem indexing, at first a stemming dictionary should be created (by using the look-up-table) because when the stem indexing process started with keyword list file, for each key word in the keyword list, this keyword will be searched in the dictionary file after checking if it is not word. By using stemming, the relevancy of the retrieved documents will be rectified and their number will also be increased. In rot indexing, the root can be defined as a word after removing all attached affixes (prefixes, infixes and suffixes). As the stemming process for each term in the document, it must be checked against the root dictionary file, after checking if this term is not a stop word.

Proposed algorithm for implementing the information retrieval system: In this section, we describe our novel algorithm of information retrieval system. This system is capable of performing an automatic information retrieval to handle the Arabic text. We have constructed an automatic full word and root indexing using inverted file technique. Our system aims to retrieve the data which may more relevant to the user demand. By using vector space model with cosine similarity, the system retrieves the data in a descending similarity order depending on the most relevant documents to the user demand.

Also, we add new features to our system in order to improve the retrieval performance as:

1. The stop words
2. Root
3. Calculating the similarity

4. Ranking the retrieved documents in a descending order according to the similarity. Our study will concentrates on making a comparison between two types of indexing methods: the full word indexing and the root indexing using vector space model.

Depending on the inverted Table, when the user enters a certain query and asks for all the most relevant documents to that query, the system will calculate the similarity between the query and all the documents, which is already implemented in the system.

We use vector space model with cosine similarity which improves the retrieval process compared with the Boolean model that calculate the similarity depending on set-theory which makes full matching without ranking whereas the similarity using vector space model make a partial matching and returns the results in descending order. The most relevant document has a highest similarity and less relevant have less similarity value. Also, using vector space model can be more helpful for the user in which it doesn't require a highly skilled user in writing a query.

The proposed algorithm that is used for implementing the information retrieval system can be described as shown in the following steps.

1. Select the number of documents as search data.
2. Build the stop word table.
3. Build the Inverted table

3.1 Read the documents term by term.

3.2 Apply the stop word test to check if this term is a stop word, if the term is a stop word, discard it. Otherwise, if you are using full word indexing go to step 3.2.2. In case of root indexing go to step 3.2.1.

3.2.1 Get the root of the term, go to step 3.2.2.

3.2.2 Add term to the inverted table.

3.3 For each term, we execute the following:

3.3.1 Add document number where you get the term.

3.2.2 Calculate the number of times the term occurred in that document (frequency).

3.3.3 Calculate the number of documents where the term occurs (ni).

3.3.4 Calculate the inverse document frequency (IDF) measurement where:

Idf = log (N/ni)

N: Total number of documents.

3.3.5 Calculate the weight of the term:

W = tf * idf

Tf = (freq.) / maxfreq.

Maxfreq.: the max term , Frequency Appeared in that document.

Most of the existing Arabic stemming algorithms depend on existing pattern and roots files, and this require too many spaces to store these field and it is a time consuming. In our study, we use a novel approach which is completely different than the previous algorithms which does not depends on any numeric values of each word letter. Also, it makes some calculations on these values to extract the roots. The proposed and used algorithm was tested using a set of ten abstracts chosen randomly from a corpus of 242 abstracts from the proceedings of the Saudi Arabian National Computer Conferences. The results showed that this novel algorithm extract the correct root with an accuracy rate reached up to 95%. Roots can be used in many applications such as compression of data; in spell checking in IR systems where many studies showed that using roots as an index word in IR give must better results than using full words.

EXPERIMENTAL RESULTS

Regarding experiment of full word indexing method, we examined the system by using 10 queries against 242 Arabic documents using the vector space model with cosine similarity measurement, the system then retrieve documents in descending order according to their similarity values as shown in table 1.

Table 1: The output of a query search in full word indexing method

Doc. Name	Sim
d 47. txt	0.347877448
d177. txt	0.136976681
d 53. txt	0.099864103
d 211. txt	.0045161852
d 45. txt	0.038065734
d 26. txt	0.032986595
d 55. txt	0.031694289
d 71. txt	0.022761926

On the other side, regarding the experiment of root indexing method, we examined the system by using the same 10 queries against the 242 Arabic document using the vector space model with cosine similarity measurement, the system then retrieve documents in descending order according to their similarity value as shown in Table 2.

Table 2: The output of query 2 search in root indexing method

Doc. Name	Sim
d 47. txt	0.398656
d 210. txt	0.294367
d 211. txt	0.275109
d 177. txt	0.196763
d 49. txt	0.154029
d 58. txt	0.149503
d 55. txt	0.123651
d 53. txt	0.07157

Our experimental results also concerned with the evaluation of retrieval efficiency and its effectiveness by comparing the results of full word indexing and root indexing based on a recall and precision measurement and representation of the interpolating in Excel charts.

When considering retrieval performance evaluation, first we should consider the retrieval task that is to be evaluated. The retrieval task could consists simply of query proceed in batch mode (i.e., the user submit a query then receives an answer back) or of a

whole interactive session (i.e., the user specifies the information needed through a series of interactive steps with the system).

In view point of recall and precision which characterize the main performance measure of IR efficiency, consider an example information request 1 (of a test reference collection) and its set R of relevant documents. Let $|R|$ be the number of documents in this set. Assuming that a given retrieval strategy (which is being evaluated) processed the information request I and generates a document answer set A. Let $|A|$ be the number of document in this set. Fig. 1 illustrates these set:

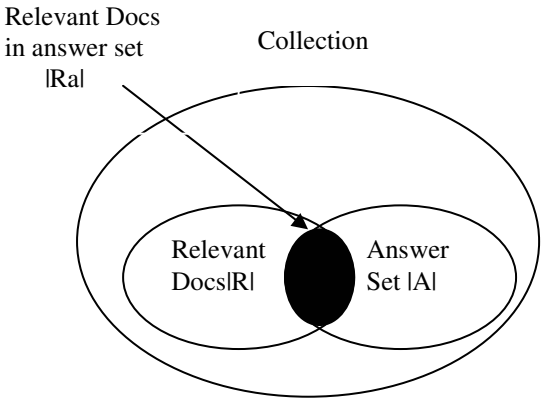


Fig. 1: Precision and recall for a given example information request

Recall is the function of the relevant documents (the set R) which has been retrieved:

$$\text{Recall} = |Ra| / |R| \quad (1)$$

Precision is the function of the retrieved documents (the set A) which is relevant:

$$\text{Precision} = |Ra| / |A| \quad (2)$$

Here, recall and precision assume that all the documents in the answer set A have been examined or seen. But usually, the user is not presented with all documents in the answer set A at ones. Because the documents in A are first stored according to a degree of relevance (i.e., a ranking is made). Then, the user can examine this ranked list starting with top document. At this situation, the measurement of recall and precision will be varying depending on the user examination on the answer set A. So, the proper evaluation requires plotting the precision versus recall curve.

Recall and precision are binary measurements, where an object is either relevant or non-relevant (true or false). Relating this to IR, measurement can be relevance and retrieval. Integrating this with the binary measurement creates 2*2 possible states:

1. Retrieved and Relevant.
2. Retrieved and Not Relevant.
3. Not Retrieved and Relevant.
4. Not Retrieved and not Relevant.

Table 3 gives our results concerning the precision and recall on a query using full word indexing method while table 4 illustrate Recall and Precision for the queries using VMS on full word Indexing method and Table 5 illustrate Recall and Precision for the queries using VSM on root indexing method.

Table 3: Result of precision and recall on a query using full word indexing method

Doc Name	Sim	Precision	Recall
d 26.txt	0.032987	1	0.375
d 55.txt	0.031694	1	0.4375
d 71.txt	0.022762	0.875	0.4375
d 69.txt	0.022006	0.777778	0.4375
d 105.txt	0.018282	0.7	0.4375
d 46.txt	0.018003	0.727273	0.5
d 66.txt	0.0175	0.666667	0.5
d 212.txt	0.016125	0.692308	0.5625
d 73.txt	0.015003	0.0642857	0.5625
d 176.txt	0.014103	0.6	0.5625

Table 4: Recall and precision for the queries using VMS on full word Indexing method

Average Precision	Recall
0.866666667	0.0
0.870389610	0.1
0.844139194	0.2
0.691585973	0.3
0.657498731	0.4
0.634941054	0.5
0.525144994	0.6
0.460220366	0.7
0.390006343	0.8
0.303366923	0.9
0.20379771	1

Table 5: Recall and precision for the queries using VSM on root indexing method

Average Precision	Recall
0.900000000	0.0
0.884230769	0.1
0.82000000	0.2
0.785227273	0.3
0.725335775	0.4
0.717325369	0.5
0.673046295	0.6
0.587089479	0.7
0.479946120	0.8
0.406774439	0.9
0.270060729	1.0

When we make a comparison between the full word indexing and the root indexing, we made an interpolating on these two indexing methods. We compute the precision and recall on an Arabic set of queries (10 queries). Then, we divide the recall values into suitable intervals and read the related precision values which varies in each interval. After that, we select the maximum value of precision in each recall's interval. Their, we fixed the intervals of recall to the tested ten queries and took the average of precession which is ready to be drown. We made the interpolating for each full word and root indexing methods. By drawing the recall versus precision for these two methods of indexing, we obtained the results. Figure 2

is the chart (we choose the most compatible intervals of recall that most fit our results for the selected 10 queries).

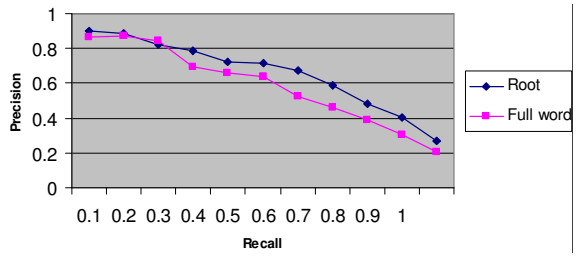


Fig. 2: Interpolating for full word and root indexing methods

From Fig. 2, we see that the precision evaluation of root indexing is better than full word in which root indexing gives higher values than the full word indexing. Also, from this Fig. 2, we can conclude to the following points:

1. In general, precision usually decreases in value through the intervals of recall for one indexing method.
2. The decreasing in precision is due to the sequential increasing of retrieved documents while not all retrieved documents are relevant.
3. The values of precision in root are always greater than the full word indexing, while the range of intervals of recall is the same for both.
4. Although the precision simultaneously decrease, the loss in precision will be smaller than the profit in recall.

Diagrams shown in the appendix A illustrate a comparison between full word and root indexing in view point of Execution Time of Retrieving Process. According to our experimental results, we summarize some notes as follow:

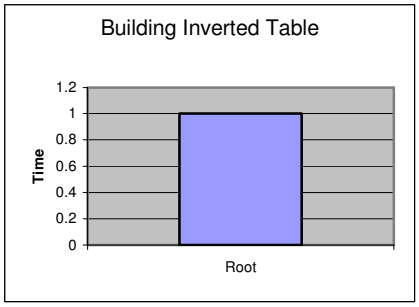
1. When the weight equal zero, this means that the term is existed in all the documents, so N = ni, and idf = 0, so weight = 0.
2. When the similarity equals zero, this indicates that there is not match between any term of the query and any documents.
3. The term frequency (tf) measurement (freq/maxFreq) gives a ratio of a term frequency to the maximum frequency in that document, this improves the results rather than using only frequency, which doesn't distinguish the term in document, because there can be two terms or more with the same frequency.
4. The inverse term frequency (idf) measurement (log N/ni) is constant for the term in all documents and gives an indication about the term to all documents.
5. We deal with the query as a document, and make indexing to its term, calculate the frequency for them, get the value of (idf) from the inverted table because the (idf) is constant for the term any where used, and calculate the weight for the query terms.

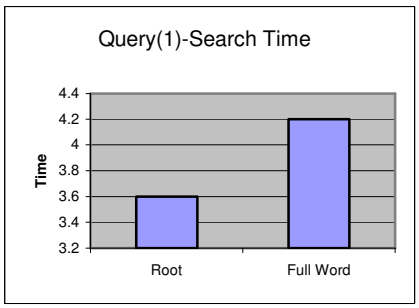
CONCLUSION

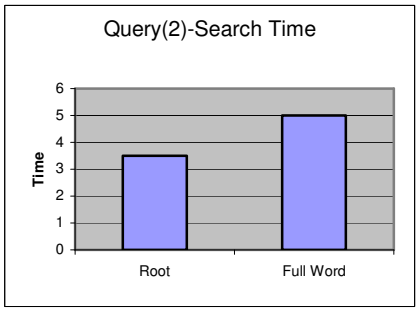
The interest of this paper lies in making a comparison between the full word indexing method and the root indexing using vector space model with cosine similarity an Arabic texts. Our experimental results showed that using root indexing method will give better results than using full word indexing method.

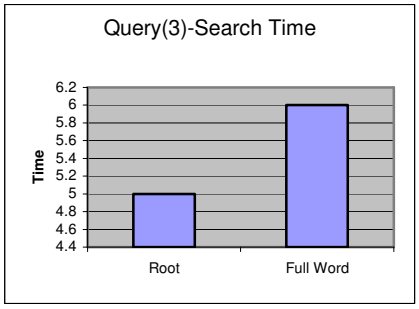
Using vector space model with cosine similarity provides better retrieval performance by ranking the retrieved data descending according to the similarity which gives indication to the user which may be suites his need better. Root indexing is very useful in the system because it minimizes the size of storage area, reduces the time of system processing, and gives wider amount of retrieved data which may more relevant to user query.

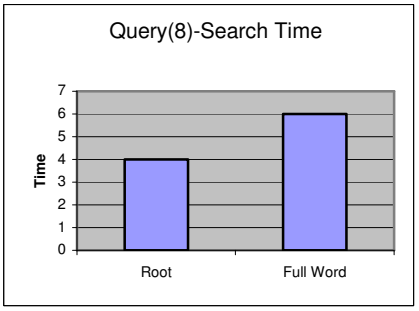
Appendix A: A comparison between full word and root indexing in view point of execution time of retrieving process.

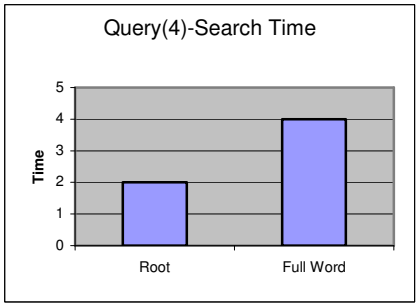


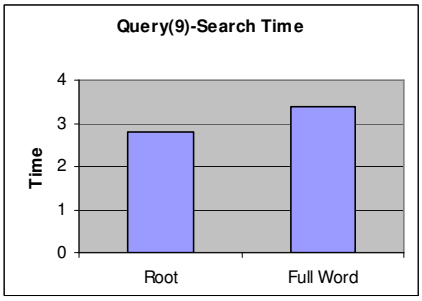


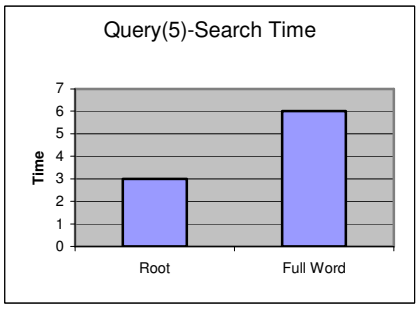


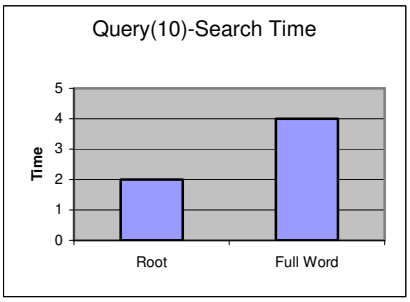


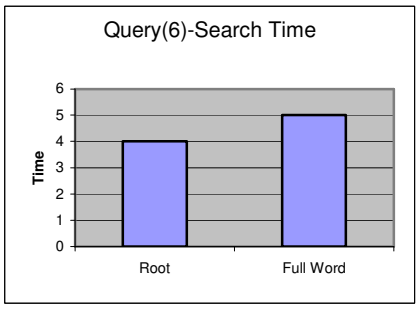


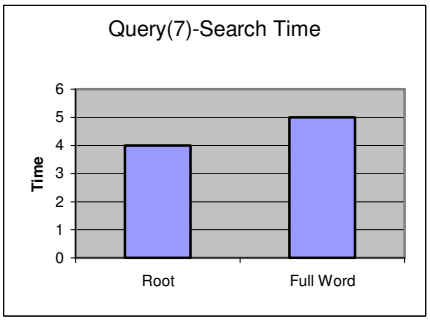












REFERENCES

1. Ricardo, B.Y., Santiago, C.B.-R. Neto and B. Horizonte, 1999. Modern Information Retrieval.
2. Van Rijsbergen, C.J., 1979. Information Retrieval. Computer Laboratory, University of Cambridge, Sec. Edn.
3. Salton, G. and M.J. McGill, 1983. Introduction to Modern Information Retrieval. McGraw-Hill Inc, New York.
4. Salton, G., 1975. A Theory of Indexing. Regional Conf. Series in Appl. Math., Soc. for Indust. and Appl. Math., Philadelphia, Pennsylvania.
5. Yates, R. and B. Neto, 1999. Modern Information Retrieval. Addison-Wesley, New York.
6. Aljaly, M. and O. Frieder, 2002. On Arabic Search: Improving the Retrieval Effectiveness via light Stemming Approach.
7. Lassi, M., 2002. Automatic Thesaurus Construction. University Collage of Boras, Sweden.
8. Xu, J., A. Fraser and R. Weischedel, 2002. Empirical Studies in Strategies for Arabic Retrieval.