Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents

Ismail Hmeidi
Jordan Institute of Science and Technology, Irbid, Jordan

Ghassan Kanaan and Martha Evens*
Computer Science Department, Illinois Institute of Technology, 10 West 31st St., Chicago, IL 60616.
E-mail: mwe@math.nwu.edu; csevens@minna.acc.iit.edu

We have put together a corpus of 242 abstracts of Arabic documents using the *Proceedings of the Saudi Arabian National Conferences* as a source. All these abstracts involve computer science and information systems. We also designed and built an automatic information retrieval system from scratch to handle Arabic data. The system was implemented in the C language using the GCC compiler and runs on IBM/PCs and compatible microcomputers. We have implemented both automatic and manual indexing techniques for this corpus. A long series of experiments using measures of recall and precision has demonstrated that automatic indexing is at least as effective as manual indexing and more effective in some cases. Since automatic indexing is both cheaper and faster, our results suggest that we can achieve a wider coverage of the literature with less money and produce as good results as with manual indexing. We have also compared the retrieval results using words as index terms versus stems and roots, and confirmed the results obtained by Al-Kharashi and Abu-Salem with smaller corpora that root indexing is more effective than word indexing.

1. Introduction

Nowadays the rapid development in computer technology makes computer applications and software systems available and easy to use in almost every field in the Arabic countries. But users of computers in the Arab world need to know English because Arabic interfaces and Arabic natural language processing systems are not widely available.

We hope that this problem is of concern to a still more general audience, since interest in the application of techniques developed for English to other languages has been stimulated by the D.O.D. Tipster project (Harman, 1993). Arabic provides a very different context from English, since it is a non-Indo-European language with a complex morphological structure.

Investigation of methods of automatic information retrieval for Arabic is essential to the growth of learning in the Arab world. Expansion of information retrieval systems is the simplest and most cost-effective way to make the resources of large reference libraries available to the increasing numbers of students and researchers in the Arab world.

1.1. Automatic Indexing

In the United States the large bibliographic database maintained by the National Library of Medicine is indexed by hand using the MESH vocabulary (Salton & McGill, 1983). Two large legal databases, Westlaw (maintained by West Publishing Company) and Lexis (maintained by Mead Data Central), are indexed by hand and largely maintained by hand also. This hand indexing is very expensive and time-consuming. It sometimes takes several months after the publication of a journal before the articles are indexed at the National Library of Medicine. This means that it is often several months before interested researchers become aware that important new articles are available.

Salton (1975) has long argued that automatic indexing would produce better results for the National Library of Medicine and has carried out a number of experiments that substantiate his claims (Salton & McGill, 1983). While human indexers clearly understand the texts better than our best programs can now interpret them, they also make many mistakes in handling the huge numbers of index terms and articles involved. We are convinced by Salton's arguments that machine indexing in English is more accurate and more cost-effective. He has also sug-

* To whom all correspondence should be addressed.

Received November 10, 1995; revised June 13, 1996; accepted August 1, 1996.

TABLE 1. The token to type ratio is the quotient obtained by dividing the number of running words (token) by the number of distinct words (type) (Adapted from Yahya, 1989, pp. 1–7).

Length of text	Arabic distinct words	Arabic Ratio	English distinct words	English Ratio
100	84	1.190	69	1.449
200	149	1.342	124	1.613
400	281	1.423	165	2.424
800	507	1.578	328	2.439
1,600	902	1.774	621	2.576
3,200	1,537	2.082	871	3.674
6,400	2,715	2.357	1,361	4.702
12,800	4,895	2.615	2,337	5.477
16,000	5,775	2.771	2,699	5.928
20,000	6,956	2.875	3,154	6.341

gested methods for cross-checking the results for consistency. We set out to discover whether these methods that work for English are appropriate for Arabic.

Salton's methods are based on term frequency. The most frequent terms are discarded as too general, the least frequent as too specific, and the midrange terms are used as index terms. We set out to discover whether this methodology can be applied to Arabic documents. The transfer is not a simple process since the complexities of Arabic morphology imply that any individual word will appear much less often, that is, the token to type ratio for Arabic is much lower than for English, and the inverse document frequency, which is central to many approaches to automatic indexing, is typically much higher. The token to type ratio is the quotient obtained by dividing the number of running words by the number of distinct words in a text. Yahya (1989) measured the number of distinct words per constant number of running words for texts of different sizes in Arabic and in English. Table 1, adapted from Yahya (1989), shows the different behavior of the Arabic and English languages. We found 48,538 running words in our Arabic corpus and 9,443 distinct words, so the token to type ratio is 5.14.

Why do individual Arabic words typically appear less often than English ones? First, verbs and many derived nouns and adjectives are formed from roots with person, number, gender, and tense expressed by affixes (Ali, 1988). Second, subject and direct object pronouns are often combined with the verb. Third, the definite article, the word for ''and,'' as well as several common prepositions are combined with the following word with no word spaces (Omar, 1984; Saliba and Al-Dannan, 1989). With such a large proportion of words appearing only once, is it possible to find an appropriate lower threshold for frequency of index terms? This was not at all clear before we began our experiments.

The main goals of this research are to investigate some information retrieval problems in the Arabic language, in particular:

1. To implement automatic and manual indexing techniques and compare the applicability of both methods to information retrieval with Arabic documents.
2. To repeat some of the experiments carried out by Al-Kharashi (1991) and Abu-Salem (1992) with a larger collection of abstracts.

1.2. Outline of This Article

After a brief review of the literature, this article describes the methodology we used in automatic indexing, and then describes the experimental procedure. We compare the results of manual and automatic indexing and also the results of using words, stems, and roots as index terms. The last section presents our conclusion and our plans for future research.

2. Review of Literature

Salton has long argued for automatic indexing (1975, 1989). Salton and McGill (1983) list some of the weaknesses of the manual indexing process. Such weaknesses are: (a) The indexer must be trained and have a knowledge of the collection; (b) the results may be inaccurate because of inexperienced indexers; different experts may produce different index terms; and (c) the cost. They argue that automatic indexing is much more cost effective.

Salton (1975) reviews several techniques capable of measuring the significance of indexing terms. The best known of these techniques are those that make use of term frequency. These techniques are based on the observation of Luhn (1957) that the frequency of word occurrence in an article furnishes a useful measurement of word importance.

Since neither the high nor the low frequency terms are good content identifiers, Luhn (1957) conjectured that the ''resolving power'' of index words extracted from the document abstract would peak in the middle-frequency range. These results were confirmed for English in a long series of experiments by Salton and McGill (1983).

Word stems have long been used in automatic text processing. The purpose is generally to increase the level of exhaustivity of indexing (Lancaster, 1978). Indexing exhaustivity is defined as the number of different topics indexed, while indexing specificity is defined as the level of precision with which a document is indexed.

In order to use this kind of algorithm with Arabic documents, we need a way of finding roots automatically. Al-Fedaghi and Al-Anzi (1989) describe a new algorithm to generate Arabic root-pattern forms of a given Arabic word based on a mathematical method. Hilal (1985, 1989) describes a more comprehensive theoretical approach to Arabic root and pattern extraction. Fortunately, our colleague Riyad Al-Shalabi (1996) has implemented a morphological analysis program for our use.

2.1. Other Experiments with Arabic Information Retrieval

Morfeq (1990) described a text database management system for Arabic engineering documents called Bayan.

A morphological analyzer for Arabic words was designed, which has the capability of handling a database of bilingual documents.

Al-Kharashi and Evens (1994) designed the Micro-AIRS system, a microcomputer system for Arabic Information Retrieval, as an experimental system to investigate indexing and retrieval processes for Arabic bibliographic data. A series of experiments were performed using 29 queries against a base of 355 Arabic bibliographic records, covering computer and information science from the library at King Abdul-Aziz City for Science and Technology. The records contained titles but not abstracts. These experiments revealed that using roots and using stems as index terms gives better retrieval results than using full words. The root performs as well as, or better than, the stem at low recall levels and definitely better at high recall levels. Several different binary similarity coefficients were tried: The Cosine, Dice, and Jaccard coefficients. All three led to exactly the same document rankings for every query. This suggests that it makes sense to use whichever is easiest to compute on a given hardware/software configuration. The experiments were run on an IBM/AT compatible microcomputer. Micro-AIRS is written in Turbo C, version 2.0 (Al-Kharashi & Evens, 1994).

Abu-Salem (1992) described the construction of a microcomputer-based bibliographic information retrieval system for Arabic documents (Arabic IRS) that interprets queries and retrieves relevant abstracts, with a reasonable response time. He repeated the experiments of Al-Kharashi (1991) with consistent results. He found that the system functions better with roots than with stems as index terms, and better with both roots and stems than with words. He put 120 abstracts in Arabic into machine-readable form and he demonstrated that using abstracts, as opposed to titles, gives superior results no matter what kind of index terms are used. Also, he discovered that a relational thesaurus used interactively gives the same good results as using roots as index terms.

3. Experimental Procedure

3.1. Data Description

The first step in this research was to create machine readable abstracts for more Arabic documents with abstracts. We entered 122 more Arabic abstracts using the *Proceedings of the Saudi Arabian National Conferences* as a source. This gave us a total of 242 abstracts when we combined them with the 120 entered by Abu-Salem. This collection contains records with categories, titles, authors, sources, and abstracts. All the documents involve computer science and information systems; most are about Arabic computational linguistics. We used the same organization that Al-Kharashi (1991) and Abu-Salem (1992) used for their data. The text of the records prepared by Abu-Salem had a few typing and spelling mistakes. To reduce the effect of these mistakes on the evaluation process, they were corrected before the final indexing process was carried out.

The first author used the Nafitha software developed by 01 System, Manama, Bahrain, to represent the Arabic character set (01 Systems, 1988).

The second step was to develop queries. We asked our colleagues in the Arabic Language Processing Group, who are also Arabic students in the Computer Science Department at IIT, to develop queries about topics related to their course or research interests. The list of all 60 queries is shown in Appendix A. Our colleagues in the Arabic group generously agreed to help us out and to make the relevance judgments. We divided our abstracts into six sets, each set with 40 abstracts. The resulting relevance judgments for all 60 queries and all 242 abstracts are shown in Appendix B.

We built an automatic information retrieval system from scratch to handle Arabic data. The system was implemented in the C language on DOS using the GCC Compiler, and runs on IBM/PCs and compatible microcomputers.

4. Indexing Process

4.1. The Automatic Word Indexing Process

In this section, we describe the automatic indexing process based on the three retrieval indexing methods: Full word, stem, and root indexing. To run the experiments for these three different retrieval methods, we built an automatic word indexing system to carry out information retrieval for all 60 queries against the collection of 242 Arabic abstracts.

First, we extracted all words for each document, and we calculated the frequency of each word in that document. This is the frequency of term k in document i, or $FREQ_{ik}$. Second, we determined the total collection frequency $TOTFREQ_k$ for each word by summing the frequencies of each word across all n documents. That is,

$$TOTFREQ_k = \sum_{i=1}^{n} FREQ_{ik}.$$

Third, we arranged the words in decreasing order according to their collection frequency, in order to decide on some suitable high-threshold value and remove all words with a collection frequency above this threshold. This eliminates the high frequency function words. Fourth, in the same way, we chose some low threshold to remove all words with a collection frequency below this threshold. Fifth, the remaining medium-frequency words were now used for assignment to the documents as index terms.

Let us describe this process in more detail. In our collection, we have 242 abstracts. The total number of tokens in the abstracts including the stopwords is 46,968. We counted the collection frequency for each word across all documents, and we arranged them in decreasing order according to their collection frequency. In the full word case, we have a high frequency value of 1,105 and a low frequency value of 1. Now, we must choose the high threshold and low threshold value between 1,105 and 1. For example, if we choose the high threshold value = 240 and the low threshold value = 3, then the words with frequency between 240 and 3 are assigned to the abstract as index terms. In fact, we chose a high threshold value = 260 and low threshold value = 2 for all three series of experiments. That is, the thresholds for stems and roots were chosen in the same way.

4.2. Experiments with Stem Indexing

Al-Kharashi's (1991) results and Abu-Salem's (1992) results both suggest that it is useful to use the stem indexing method, thereby reducing the original words to word stem form. The word stem will have a higher frequency of occurrence in the document abstracts than any of the variant forms. The generation of word stems, and subsequent identification of common stems, is relatively easy to do for many languages (including English) and serves as a recall enhancing device. When the stems are used as index terms, a greater number of potentially relevant items can be identified than if the full words from the text are used.

In English, several stemming algorithms have been used in different experiments (Lennon, Peirce, Tarry, & Waillett, 1981; Lovins, 1968; Porter, 1980; Salton, 1971). Stemming is used to improve retrieval effectiveness and to reduce the size of indexing files.

In Arabic, automatic prefix and suffix cutoff algorithms are much more expensive and complicated because of the complexity of Arabic morphology.

In this research, the stemming process started with the keyword list file. We extracted all word stems for each document, and we calculated the frequency of term k in document i, or $FREQ_{ik}$. Second, we determined the total collection frequency $TOTFREQ_k$ for each stem by summing the frequencies of each word across all n documents. Third, we eliminated all word stems with a collection frequency above the high threshold. Fourth, in the same way as before, we chose a low threshold and removed all word stems with a collection frequency below this threshold. Fifth, the remaining medium-frequency word stems were now used for assignment to the documents as index terms.

4.3. Experiments with Root Indexing

The results of Al-Kharashi and Evens (1994) and of Abu-Salem (1992) also suggest that it is useful to use roots as index terms. Roots are also used to improve retrieval effectiveness and to reduce the size of indexing files. The roots have a higher frequency of occurrence in the document abstracts than either of the other forms, full words or stems. The indexing for the roots was carried out just like the processes for words and stems. The remaining medium-frequency word stems were now used for assignment to the documents as index terms.

5. Results of Retrieval Experiments with Different Sets of Index Terms

5.1. Manual Indexing Experiments

The results of processing 60 queries against 242 abstracts using the manual indexing method are shown in Table 2.

5.2. Automatic Indexing Experiments

The results of processing 60 queries against 242 abstracts using the automatic root indexing method with Arabic data are shown in Table 3.

6. Comparison of Experimental Results

This section is concerned with the evaluation of retrieval efficiency and effectiveness using different indexing methods. It also compares and discusses the experimental results from using full words, stems, and roots as index terms. It compares the results of manual and automatic indexing using abstracts with the three indexing methods based on recall and precision measures.

6.1. Recall and Precision Measurements

In this experiment, we have used recall and precision measurements. Recall (R) is the proportion of relevant documents that are retrieved, while the precision (P) is the proportion of retrieved documents that are relevant. To smooth a sawtooth curve, we used the smoothing algorithm used by Al-Kharashi (1991) and adapted from Keen (1972). The smoothing algorithm is shown below.

A. Distribute the recall values into 10 equal sub-intervals of the interval (0..1).
B. Assign the largest precision value beginning in that interval to that interval.
C. Assign the largest precision value found in the table to the first interval (0..1).
D. Beginning from the 10th interval, remove all sawtooth lines by assigning the current interval's precision value to the next interval, if its precision value is lower than the current one.
E. To make sure that the precision value will drop gradually from a certain precision value to zero value, assign any interval with a zero precision to half of the precision value of the previous interval.

6.2. Manual Indexing

Table 4 and Figure 1 summarize the precision averages for 60 queries for manual indexing using words, stems, and roots.

TABLE 2. Manual indexing results of 60 queries against 242 abstracts using full word, stem, and root methods.

		Words		Stems		Roots	
Query no.	Rel. jdg.	Ret.	Rel.	Ret.	Rel.	Ret.	Rel.
Q1	35	3	3	3	3	7	6
Q2	12	0	0	0	0	39	5
Q3	15	0	0	4	3	13	5
Q4	5	9	4	10	5	12	5
Q5	16	8	2	11	2	11	2
Q6	1	7	0	10	0	13	1
Q7	3	3	3	3	3	5	3
Q8	3	0	0	2	1	12	1
Q9	3	4	3	5	3	5	3
Q10	2	1	0	1	0	1	0
Q11	11	7	3	18	8	23	10
Q12	11	4	0	14	3	19	4
Q13	3	2	1	2	1	4	1
Q14	10	1	1	7	2	13	5
Q15	11	1	1	15	4	19	5
Q16	11	1	0	4	0	8	0
Q17	25	4	1	7	1	65	15
Q18	4	0	0	1	0	13	1
Q19	12	7	4	10	4	13	6
Q20	3	0	0	2	0	4	1
Q21	16	1	0	14	1	19	1
Q22	5	2	0	8	1	10	3
Q23	4	0	0	1	1	2	1
Q24	7	1	0	12	2	12	2
Q25	4	0	0	3	1	16	2
Q26	26	7	4	8	4	8	4
Q27	48	7	2	23	8	38	11
Q28	5	2	1	2	1	3	1
Q29	13	6	4	14	9	17	10
Q30	6	2	1	2	1	2	1
Q31	5	1	1	6	4	9	4
Q32	7	1	1	1	1	3	2
Q33	0	0	0	1	0	1	0
Q34	0	0	0	0	0	0	0
Q35	2	0	0	0	0	0	0
Q36	11	8	1	10	1	15	1
Q37	1	0	0	0	0	1	0
Q38	2	0	0	4	1	9	1
Q39	2	0	0	0	0	0	0
Q40	7	2	1	5	2	9	5
Q41	4	0	0	0	0	0	0
Q42	6	3	2	3	2	5	2
Q43	8	1	1	8	3	10	3
Q44	21	9	4	18	8	22	8
Q45	6	0	0	1	0	11	3
Q46	2	0	0	0	0	0	0
Q47	0	0	0	0	0	0	0
Q48	26	6	5	6	5	16	6
Q49	2	0	0	0	0	0	0
Q50	3	0	0	0	0	0	0
Q51	8	0	0	4	1	6	1
Q52	1	0	0	0	0	1	0
Q53	2	0	0	2	0	4	0
Q54	5	2	2	2	2	2	2
Q55	9	1	0	3	1	6	1
Q56	3	1	1	1	1	1	1
Q57	6	2	0	8	0	8	0
Q58	5	3	3	5	3	9	5
Q59	10	0	0	3	0	11	0
Q60	12	0	0	0	0	31	6

TABLE 3. Automatic indexing results of 60 queries against 242 abstracts using full word, stem, and root methods.

		Word		Stem		Root	
Query no.	Rel. jdg.	Ret.	Rel.	Ret.	Rel.	Ret.	Rel.
Q1	35	10	7	16	9	23	13
Q2	12	1	0	2	0	113	12
Q3	15	1	0	9	5	29	7
Q4	5	13	5	17	5	31	5
Q5	16	70	6	137	14	160	15
Q6	1	9	0	13	0	25	1
Q7	3	3	3	3	3	24	3
Q8	3	1	0	7	1	36	2
Q9	3	4	3	6	3	6	3
Q10	2	5	1	17	1	104	1
Q11	11	8	4	22	9	26	10
Q12	11	24	2	57	8	109	9
Q13	3	9	1	27	3	46	3
Q14	10	5	2	11	3	99	7
Q15	11	11	4	37	8	82	9
Q16	11	11	0	68	6	77	6
Q17	25	4	1	12	1	221	25
Q18	4	1	0	7	0	114	4
Q19	12	12	7	18	7	29	9
Q20	3	1	0	14	0	23	1
Q21	16	59	5	121	14	132	14
Q22	5	5	1	23	3	36	4
Q23	4	1	1	6	2	7	3
Q24	7	4	1	20	4	23	5
Q25	4	2	0	20	1	50	2
Q26	26	20	16	21	16	21	16
Q27	48	29	13	154	40	163	43
Q28	5	2	1	14	2	15	2
Q29	13	17	8	33	13	34	13
Q30	6	5	2	9	3	9	3
Q31	5	5	2	15	5	28	5
Q32	7	2	1	2	1	4	2
Q33	0	0	0	3	0	23	0
Q34	0	0	0	0	0	0	0
Q35	2	0	0	0	0	1	1
Q36	11	11	2	25	2	40	2
Q37	1	0	0	1	0	7	0
Q38	2	0	0	7	2	49	2
Q39	2	0	0	1	1	5	1
Q40	7	6	1	12	2	25	7
Q41	4	0	0	1	1	1	1
Q42	6	6	4	6	4	34	5
Q43	8	2	2	17	4	29	5
Q44	21	54	14	84	16	129	16
Q45	6	3	1	5	1	23	5
Q46	2	9	0	9	0	9	0
Q47	0	0	0	2	0	2	0
Q48	26	13	8	14	8	30	13
Q49	2	0	0	1	1	1	1
Q50	3	0	0	2	2	3	2
Q51	8	0	0	13	6	17	8
Q52	1	1	0	3	0	12	0
Q53	2	0	0	3	0	6	0
Q54	5	2	2	4	2	5	2
Q55	9	1	0	8	1	72	9
Q56	3	1	1	5	1	11	1
Q57	6	4	0	17	0	19	0
Q58	5	5	3	11	3	18	5
Q59	10	3	0	14	1	99	3
Q60	12	0	0	2	0	113	12

TABLE 4. Average recall-precision table after zero-smoothing in the manual indexing environment.

Recall	Word precision	Stem precision	Root precision
0.1	0.3881	0.4900	0.4968
0.2	0.3340	0.4499	0.4798
0.3	0.2347	0.3271	0.3945
0.4	0.1880	0.2661	0.3222
0.5	0.1240	0.1802	0.2367
0.6	0.0920	0.1357	0.1758
0.7	0.0672	0.1052	0.1260
0.8	0.0550	0.0833	0.1056
0.9	0.0442	0.0635	0.0823
1.0	0.0388	0.0536	0.0710

TABLE 5. Average recall-precision after zero-smoothing in the automatic indexing environment.

Recall	Word precision	Stem precision	Root precision
0.1	0.3901	0.4591	0.4709
0.2	0.3728	0.4373	0.4498
0.3	0.3085	0.3865	0.4195
0.4	0.2658	0.3516	0.3713
0.5	0.1869	0.2890	0.3051
0.6	0.1466	0.2274	0.2399
0.7	0.1167	0.1912	0.2064
0.8	0.0817	0.1391	0.1593
0.9	0.0601	0.1088	0.1229
1.0	0.0513	0.087	0.0978

6.3. Automatic Indexing

Table 5 and Figure 2 summarize the precision averages for 60 queries for automatic indexing using words, stems, and roots.

6.4. Statistical Analysis

Up to this point, we have not been in a position to draw an accurate conclusion that one experimental result is superior to another. To decide whether the manual and automatic indexing results differ significantly, we used two nonparametric statistical tests, the Signed Pair Test, and the Wilcoxon Signed Ranked Test. The Signed Pair Test uses only the sign (not the magnitude) of the differences in sample values; thus the computed probability values depend on whether the differences in sample values are mostly positive or negative. The Wilcoxon Signed Rank Test postulates that as differences between pairs increase (Salton & McGill, 1983), they become more important.

In these experiments, we used two programs Sign-st.c and Wilcoxon.c to analyze our experimental results. These two programs were written by Al-Kharashi (1991). Also, before the use of these two nonparametric statistical tests, we need to state the null hypothesis H0 and the alternative hypothesis H1.

7. Results in the Manual Indexing Environment

7.1. Retrieval Results in the Manual Indexing Environment

H0: Retrieval results using stems are no better than results using words in the manual indexing environment.
H1: The stem retrieval method is better than the word retrieval method in the manual indexing environment.

Table 4 shows the average retrieval and average recall and precision of the 60 queries in the manual indexing environment. The results show the superiority of the root over the word retrieval method. Also, the results and the graph show that the root retrieval method gives an increase in precision when compared with the full word

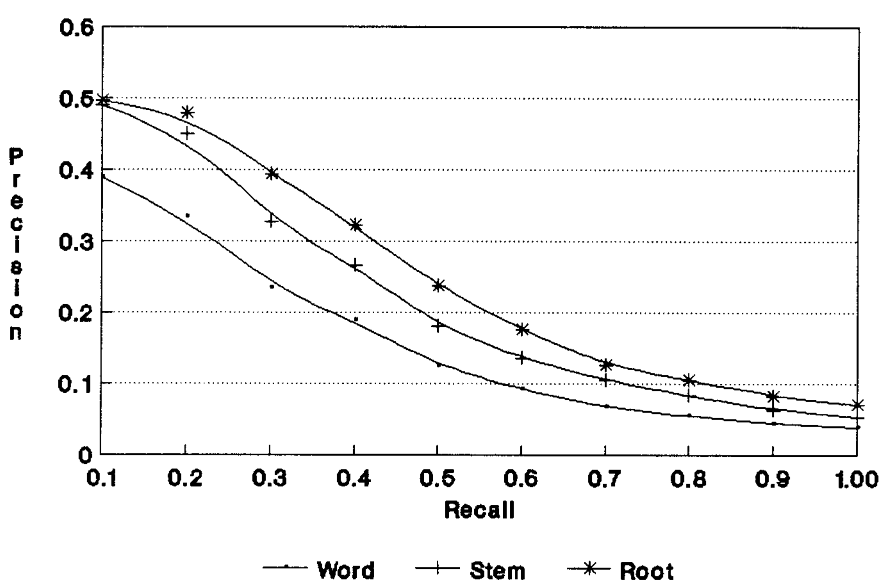

FIG. 1. Average recall-precision graph after zero smoothing in the manual indexing environment.

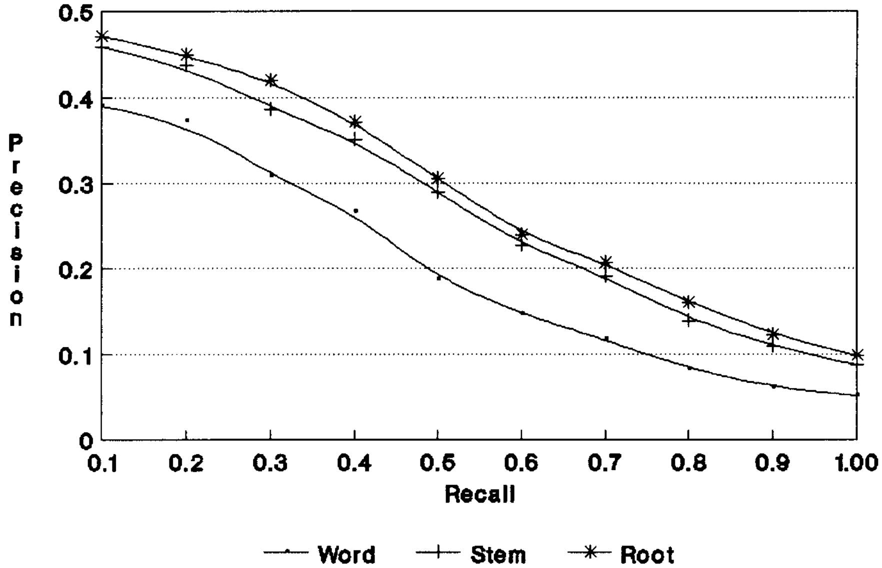

FIG. 2. Average recall-precision graph after zero smoothing in the automatic indexing environment.

stem methods. The statistical results of the Signed Pair Test and the Wilcoxon Signed Rank Test for manual indexing are shown in Tables 6 through 9. We are assuming ALPHA = 0.05.

The results of the Signed Pair Test and Wilcoxon Signed Rank Test (Daniel, 1990) are shown in Tables 6 and 7.

The null hypothesis and the alternative hypothesis for the comparison of word versus root retrieval in the manual indexing environment follow:

H0: Retrieval using roots is not better than retrieval using words in the manual indexing environment.
H1: Retrieval using roots is better than retrieval using words in the manual indexing environment.

The results of the Signed Pair Test and Wilcoxon Signed Rank Test for the comparison of word versus root retrieval in the manual indexing environment are shown in Tables 8 and 9.

7.2 Discussion of Root versus Stem versus Word in the Manual Indexing Environment

Tables 6 and 7 show that the stem retrieval method gives better results than the word retrieval method. Tables 8 and 9 show that the root retrieval method gives better results than the word retrieval method. We also compared root retrieval against stem retrieval, and the root retrieval method seems to perform better than the stem retrieval method except at the low recall levels, but the difference is not significant.

8. Retrieval Results in the Automatic Indexing Environment

Table 5 shows the average retrieval and average recall and precision of the 60 queries in the automatic indexing environment.

TABLE 6. Signed pair test for stem versus word retrieval methods in the manual indexing environment.

Recall	Favoring word	Favoring stem	Tied	One-sided prob.
0.10	6	15	39	0.0250
0.20	6	16	38	0.0166
0.30	5	20	35	0.0013
0.40	5	20	35	0.0013
0.50	5	20	35	0.0013
0.60	5	20	35	0.0013
0.70	5	20	35	0.0013
0.80	5	20	35	0.0013
0.90	5	20	35	0.0013
1.00	5	20	35	0.0013
Combined	52	191	357	0.0000

TABLE 7. Wilcoxon signed rank test for stem versus word retrieval methods in the manual indexing environment.

Recall	Favoring word	Favoring stem	NDF	One-sided prob.
0.1	52.50	178.50	21.00	0.0146
0.2	46.00	207.00	22.00	0.0045
0.3	41.50	283.50	25.00	0.0006
0.4	34.50	290.50	25.00	0.0003
0.5	32.50	292.50	25.00	0.0002
0.6	33.00	292.00	25.00	0.0002
0.7	30.50	294.50	25.00	0.0002
0.8	30.50	294.50	25.00	0.0002
0.9	30.50	294.50	25.00	0.0001
1.0	28.00	297.00	25.00	0.0002

TABLE 8. Signed pair test for root versus word retrieval methods in the manual indexing environment.

Recall	Favoring word	Favoring root	Tied	One-sided prob.
0.10	10	20	30	0.0344
0.20	8	24	28	0.0024
0.30	8	28	24	0.0004
0.40	7	30	23	0.0001
0.50	7	31	22	0.0001
0.60	7	31	22	0.0001
0.70	7	32	21	0.0000
0.80	7	32	21	0.0000
0.90	6	33	21	0.0000
1.00	6	33	21	0.0000
Combined	73	294	233	0.0000

TABLE 10. Signed pair test for stem versus word retrieval method in the automatic indexing environment.

Recall	Favoring word	Favoring stem	Tied	One-sided prob.
0.10	20	16	24	0.2546
0.20	23	16	21	0.1314
0.30	19	21	20	0.3783
0.40	17	23	20	0.1736
0.50	14	28	18	0.0154
0.60	14	28	18	0.0154
0.70	14	28	18	0.0154
0.80	12	30	18	0.0028
0.90	12	30	18	0.0028
1.00	12	30	18	0.0028
Combined	157	250	193	0.0000

8.1. Roots versus Stems versus Words in the Automatic Indexing Environment

The null hypothesis and the alternative hypothesis for comparison of words versus stems as index terms in the automatic indexing environment follow:

- H0: Retrieval using stems as index terms gives no better results than retrieval using words in the automatic indexing environment.
- H1: Retrieval using stems as index terms gives better results than retrieval using words in the automatic indexing environment.

The results of the Signed Pair Test and Wilcoxon Signed Rank Test for the comparison of stem versus word retrieval in the automatic indexing environment are shown in Tables 10 and 11.

The null hypothesis and the alternative hypothesis for the comparison of word versus root retrieval in the automatic indexing environment follow:

- H0: Retrieval using roots is not better than retrieval using words in the automatic indexing environment.
- H1: Retrieval using roots is better than retrieval using words in the automatic indexing environment.

The results of the Signed Pair Test and the Wilcoxon Signed Rank Test for the comparison of word versus root retrieval in the automatic indexing environment are shown in Tables 12 and 13.

8.2 Discussion of Root versus Stem versus Word Results in the Automatic Indexing Environment

The retrieval results for 60 queries against 242 abstracts using full word, stem, and root methods are shown in Table 4. Also, Table 5 shows the average retrieval and average recall and precision of the 60 queries. The results show the superiority of roots over words in the automatic indexing environment.

However, the results and the graph do not demonstrate clearly that the root retrieval method improves results when compared with the stem method or that the stem is necessarily better than the word. The statistical results of the Signed Pair Test and the Wilcoxon Signed Rank Test in the automatic indexing environment are shown in Tables 10 through 13. We are assuming ALPHA = 0.05. Tables 10 and 11 show that, when the recall levels range from 0.1 to 0.4, the p-values are greater than 0.05, but when the recall levels range from 0.5 to 1.0, the p-value is less than 0.05. This means that at high recall levels, the stem performs better than the word. The results at the low recall levels are inconclusive. Tables 12 and 13 show

TABLE 9. Wilcoxon signed rank test for root versus word retrieval methods in the manual indexing environment.

Recall	Favoring word	Favoring root	NDF	One-sided prob.
0.1	138.00	327.00	30.00	0.0262
0.2	111.50	416.50	32.00	0.0022
0.3	089.00	577.00	36.00	0.0001
0.4	071.00	632.00	37.00	0.0000
0.5	086.50	654.50	38.00	0.0000
0.6	100.50	640.50	38.00	0.0001
0.7	098.00	682.00	39.00	0.0000
0.8	103.00	677.00	39.00	0.0000
0.9	074.00	706.00	39.00	0.0000
1.0	073.00	707.00	39.00	0.0000

TABLE 11. Wilcoxon signed rank test for stem versus word retrieval method in the automatic indexing environment.

Recall	Favoring word	Favoring stem	NDF	One-sided prob.
0.1	314.00	352.00	36.00	0.3859
0.2	396.00	384.00	39.00	0.4681
0.3	347.50	472.50	40.00	0.2005
0.4	310.50	509.50	40.00	0.0918
0.5	244.50	658.50	42.00	0.0049
0.6	253.00	650.00	42.00	0.0066
0.7	257.50	645.50	42.00	0.0078
0.8	211.50	691.50	42.00	0.0013
0.9	206.00	697.00	42.00	0.0011
1.0	212.00	691.00	42.00	0.0014

TABLE 12. Signed pair test for root versus word retrieval method in the automatic indexing environment.

Recall	Favoring word	Favoring root	Tied	One-sided prob.
0.10	21	22	17	0.4404
0.20	24	21	15	0.3300
0.30	20	27	13	0.1539
0.40	17	30	13	0.0294
0.50	14	36	10	0.0009
0.60	12	38	10	0.0001
0.70	10	40	10	0.0000
0.80	9	41	10	0.0000
0.90	9	41	10	0.0000
1.00	9	41	10	0.0000
Combined	145	337	118	0.0000

TABLE 14. Signed pair test for automatic versus manual indexing using words as index terms.

Recall	Favoring manual	Favoring automatic	Tied	One-sided prob.
0.10	11	10	39	0.4168
0.20	10	12	38	0.3372
0.30	7	20	33	0.0062
0.40	6	21	33	0.0020
0.50	5	22	33	0.0005
0.60	5	22	33	0.0005
0.70	5	22	33	0.0005
0.80	5	22	33	0.0005
0.90	5	22	33	0.0005
1.00	5	22	33	0.0005
Combined	64	195	341	0.0000

that, when the recall levels range from 0.1 to 0.3, the p-values are greater than 0.05, but when the recall levels range from 0.4 to 1.0, the p-value is less than 0.05. This means that at high recall levels the root performs better than the word. At low recall levels, the results are inconclusive. We also compared root indexing versus stem indexing, but the results were not significant.

9. Comparing Manual and Automatic Indexing Results

The null hypothesis and the alternative hypothesis for manual versus automatic indexing using words as index terms follow:

H0: Automatic indexing is no better than manual indexing using words as index terms.
H1: Automatic indexing works better than manual indexing using words as index terms.

The results of the Signed Pair Test and Wilcoxon Signed Rank Test for the comparison of automatic versus manual indexing using words as index terms are shown in Tables 14 and 15.

The null hypothesis and the alternative hypothesis for manual versus automatic indexing using stems as index terms follow:

H0: Automatic indexing is no better than manual indexing using stems as index terms.
H1: Automatic indexing is better than manual indexing using stems as index terms.

The results of the Signed Pair Test and Wilcoxon Signed Rank Test for the comparison of automatic versus manual indexing using stems as index terms are shown in Tables 16 and 17.

The null hypothesis and the alternative hypothesis for manual versus automatic indexing using roots as index terms follow:

H0: Automatic indexing is no better than manual indexing using roots as index terms.
H1: Automatic indexing is better than manual indexing using roots as index terms.

The results of the Signed Pair Test and Wilcoxon Signed Rank Test for the comparison of automatic versus manual indexing using roots as index terms are shown in Tables 18 and 19.

9.1. Discussion of Manual and Automatic Indexing Results

The results of the Signed Pair Test and the Wilcoxon Signed Rank Test for Automatic versus Manual Indexing

TABLE 13. Wilcoxon signed rank test for root versus word retrieval method in the automatic indexing environment.

Recall	Favoring word	Favoring root	NDF	One-sided prob.
0.10	412.00	534.00	43.00	0.2327
0.20	469.50	565.50	45.00	0.2946
0.30	421.50	706.50	47.00	0.0668
0.40	379.00	749.00	47.00	0.0256
0.50	332.00	943.00	50.00	0.0016
0.60	328.00	947.00	50.00	0.0014
0.70	277.50	997.50	50.00	0.0003
0.80	222.50	1,052.50	50.00	0.0000
0.90	201.50	1,073.50	50.00	0.0000
1.00	200.50	1,074.50	50.00	0.0000

TABLE 15. Wilcoxon signed rank test for automatic versus manual indexing using words as index terms.

Recall	Favoring manual	Favoring automatic	NDF	One-sided prob.
0.1	118.00	113.00	21.00	0.4681
0.2	93.50	159.50	22.00	0.1423
0.3	73.00	305.00	27.00	0.0027
0.4	57.00	321.00	27.00	0.0008
0.5	47.50	330.50	27.00	0.0003
0.6	48.00	330.00	27.00	0.0004
0.7	43.50	334.50	27.00	0.0002
0.8	43.50	334.50	27.00	0.0002
0.9	43.50	334.50	27.00	0.0002
1.0	43.00	335.00	27.00	0.0002

TABLE 16. Signed pair test for automatic versus manual indexing using stems as index terms.

Recall	Favoring manual	Favoring automatic	Tied	One-sided prob.
0.10	20	12	28	0.0793
0.20	20	14	26	0.1539
0.30	20	22	18	0.3821
0.40	16	26	18	0.0618
0.50	15	28	17	0.0239
0.60	14	29	17	0.0113
0.70	14	29	17	0.0113
0.80	13	30	17	0.0048
0.90	12	31	17	0.0019
1.00	12	31	17	0.0019
Combined	156	252	192	0.0000

TABLE 18. Signed pair test for automatic versus manual indexing using roots as index terms.

Recall	Favoring manual	Favoring automatic	Tied	One-sided prob.
0.10	22	14	24	0.0918
0.20	25	14	21	0.0392
0.30	27	21	12	0.1949
0.40	24	25	11	0.4443
0.50	25	25	10	0.5000
0.60	19	32	9	0.0344
0.70	15	36	9	0.0016
0.80	14	37	9	0.0006
0.90	13	38	9	0.0002
1.00	13	38	9	0.0002
Combined	197	280	123	0.0001

using words are listed in Tables 14 and 15. These tables show that at recall levels 0.1 to 0.2, the p-values are greater than 0.05, but the remaining p-values are less than 0.05. We conclude that automatic indexing gives better results than manual indexing using words as index terms.

Results of automatic versus manual indexing using stems as index terms are not conclusive. Table 16 shows that at recall levels 0.1 to 0.4, the p-value for the Signed Pair Test is greater than 0.05, and the remaining p-values are less than 0.05. The Wilcoxon Sign Ranked Test, a more powerful test, shows p-values less than 0.05 for recall levels above 0.3 (Table 17). For higher recall values, automatic indexing performs no worse than manual indexing using stems as index terms.

Results of automatic versus manual indexing using roots as index terms in Tables 18 and 19 show that at recall levels 0.1 to 0.5, the p-value is greater than 0.05, and the remaining p-values are less than 0.05. Differences between automatic and manual indexing using roots as index terms are not significant at low recall levels. They become significant at levels about 0.5.

10. Conclusions

We have built an Arabic information retrieval system and used it to compare the results of manual and automatic indexing. We have carried out three separate series of experiments: One series using roots as index terms, one using stems, and one using words.

The results of these experiments support the following conclusions:

1. Automatic indexing gives better results than manual indexing when words are used as index terms.
2. Automatic indexing gives better results than manual indexing when stems are used as index terms, at recall levels above 0.3.
3. Automatic indexing gives better results than manual indexing when roots are used as index terms, at recall levels above 0.5.
4. In the manual indexing environment, using roots as index terms gives better results than using stems or words.
5. In the automatic indexing environment, using roots as index terms gives better results than using words.

In conclusion, for our database, when words are used as index terms, we have shown that automatic indexing is at least as good as manual indexing. While these experiments need to be repeated on larger databases before we can make this claim in general, our results suggest that the Arab world should consider using automatic indexing on a large scale. It is both cheaper and faster than manual indexing. We have also confirmed the results obtained

TABLE 17. Wilcoxon signed rank test for automatic versus manual indexing using stems as index terms.

Recall	Favoring manual	Favoring automatic	NDF	One-sided prob.
0.1	320.50	207.50	32.00	0.1469
0.2	343.00	252.00	34.00	0.2207
0.3	386.50	516.50	42.00	0.2090
0.4	303.00	600.00	42.00	0.0322
0.5	242.00	704.00	43.00	0.0027
0.6	219.00	727.00	43.00	0.0011
0.7	223.50	722.50	43.00	0.0013
0.8	194.00	752.00	43.00	0.0004
0.9	179.00	767.00	43.00	0.0002
1.0	185.00	761.00	43.00	0.0003

TABLE 19. Wilcoxon signed rank test for automatic versus manual indexing using roots as index terms.

Recall	Favoring manual	Favoring automatic	NDF	One-sided prob.
0.1	400.00	266.00	36.00	0.1469
0.2	490.50	289.50	39.00	0.0808
0.3	623.50	552.50	48.00	0.3594
0.4	584.50	640.50	49.00	0.3936
0.5	562.00	713.00	50.00	0.2358
0.6	479.00	847.00	51.00	0.0427
0.7	355.50	970.50	51.00	0.0020
0.8	329.50	996.50	51.00	0.0009
0.9	298.50	1,027.50	51.00	0.0003
1.0	310.50	1,015.50	51.00	0.0005

by Al-Kharashi (1991) using titles only and Abu-Salem (1992) using 120 abstracts, that roots make better index terms than words or stems, at least when phrases are not involved.

In the future, we hope to run the system with a larger collection of abstracts and queries in different areas to establish the effectiveness of automatic indexing in a more thorough fashion. We also hope to investigate various phrase generation methods and experiment with automatic phrase indexing.

Appendix A: Set of 60 Queries Using Full Words

English Meaning	Arabic Meaning	Query
Num. Intelligence Machine	الذكاء و الآلي	.1
Computer Programming	البرمجه و الحاسوب	.2
Translation and Arabization	التعريب و الترجمه	.3
Quran or Hadeeth	القرآن او السنه	.4
Computer in Saudi Arabian	الحاسب و السعوديه	.5
Documentation or Indexing	التوثيق أو الفهرسه	.6
Pilgrimage or Omra	الحج أو العمره	.7
Systems in Libraries	نظم و المكتبات	.8
Pascal or FORTRAN	باسكال أو الفورتران	.9
Computer and Mathematics	الحاسب أو الرياضيات	.10
Dictionary or Lexicon	القاموس أو المعجم	.11
Information Processing	معالجه و المعلومات	.12
Machine Control	التحكم و الآلي	.13
Computers and Teaching	التدريس و الحاسبات	.14
Computer and Telecommunication	الحاسبات و الاتصالات	.15
Computer and Design	التصميم والحاسبات	.16
Computer and Education	التعليم و الحاسوب	.17
Computer and Management	الاداره و الحاسوب	.18
Database	قواعد و البيانات	.19
Operating System	نظم و التشغيل	.20
Computer and Technology	تقنيه و الحاسبات	.21
Grammar and Arabic Language	النحو و اللغه و العربيه	.22
Computer Algorithms	الحاسب و الخوارزميات	.23
Morphology and Arabic Language	الصرف و اللغه و العربيه	.24
Computer and Industry	الحاسب و الصناعه	.25

Geographic Information	المعلومات و الجغرافيه	.52
Artificial Intelligence	الذكاء و الاصطناعي	.26
Computer and (Arabic or English) Language	(الحاسب و اللغه و (العربيه او الانجليزيه	.27
Computer and Technology	تكنولوجيا و الحاسبات	.28
Computer and Networks	شبكات و الحاسب	.29
Screen and Arabic Language	الشاشه و اللغه و العربيه	.30
Computers and (Control or Security)	(الحاسبات و (الرقابه او الامن	.31
Electronic Lexicon	المعاجم و الالكترونيه	.32
Computer and Flying	الحاسب و الطيران	.33
Computer and Music	الحاسب و الموسيقى	.34
Computer and Geology	الجيولوجيا و الحاسوب	.35
Natural Language	اللغات و الطبيعيه	.36
Systems Architecture	نظم و العماره	.37
Computer and Typing	الحاسب و الطباعه	.38
Computer and Hospital	الحاسب و المستشفى	.39
Microcomputers	الحاسبات و المصغره	.40
Computer and Interpretation	الحاسب و التفسير	.41
Science and Religion	العلوم و الشرعيه	.42
Information Retrieval	استرجاع و المعلومات	.43
Information Technology	تقنيه و المعلومات	.44
Computer and Arabic Lexicon	الحاسوب و المعجم و العربي	.45
Electrical Engineering	الهندسه و الكهربائيه	.46
Computer in Military Field	الحاسب و المجال و العسكري	.47
Arabic Text	النص و العربي	.48
Children and Systems	نظام و الاطفال	.49
Drawing System	نظام و الرسم	.50
Parallel Programming	البرمجه و المتوازيه	.51
Geographic Information	المعلومات و الجغرافيه	.52

Appendix A: Continued

Computer Engineering	هندسه و الحاسوب	.53
Management and Security	الامن و الاداري	.54
Human Languages	اللغات والانسانيه	.55
Morphological Analysis	التحليل و الصرفي	.56
Computer Application	تطبيقات و الحاسوب	.57
Programming in Arabic	البرمجه و بالعربيه	.58
Computer and Children	الحاسب و الاطفال	.59
Computer and Agriculture	الحاسب و الزراعه	.60

Appendix B: Relevance Judgments

Query no.	Record no.	Total records
Q1	4 8 15 17 39 41 42 62 92 99 102 131 134 138 107 112 114 118 119 124 141 142 171 187 189 194 196 198 199 231 232 234 235 240 241	35
Q2	2 73 85 108 109 119 144 178 179 181 182 183	12
Q3	4 9 15 41 84 88 89 90 91 117 133 148 151 156 158	15
Q4	10 30 36 145 147	05
Q5	6 11 12 14 73 139 140 148 152 169 170 173 175 176 184 196	16
Q6	144	01
Q7	27 31 146	03
Q8	90 121 140	03
Q9	23 84 117	03
Q10	5 146	02
Q11	81 142 143 149 154 166 167 198 232 241 242	11
Q12	7 33 90 106 128 144 145 146 147 148 237	11
Q13	78 72 97	03
Q14	16 29 60 67 142 143 164 171 176 236	10
Q15	44 52 53 54 55 58 59 80 177 211 212	11
Q16	3 24 29 44 76 77 89 75 112 161 203	11
Q17	5 14 16 18 28 29 35 59 60 61 62 63 64 65 66 67 93 163 169 173 174 175 176 192 236	25
Q18	35 69 79 184	04
Q19	48 70 90 130 144 145 146 147 157 166 198 241	12
Q20	158 172 187	03
Q21	3 7 21 160 202 204 205 213 214 215 216 217 218 219 220 221	16
Q22	86 132 141 142 149	05
Q23	26 34 226 227	04
Q24	43 86 118 138 141 142 149	07
Q25	21 38 68 75	04
Q26	23 34 92 102 106 107 113 114 118 119 123 125 146 189 190 191 192 193 194 195 196 197 198 199 200 219	26
Q27	1 4 8 15 17 19 39 41 42 43 94 101 107 108 109 111 112 113 114 115 120 122 124 125 128 141 142 143 144 146 149 154 165 190 191 192 193 194 195 196 197 198 199 200 231 232 233 235	48
Q28	53 60 72 170 236	05
Q29	45 46 47 48 56 57 58 152 207 208 209 210 236	13
Q30	91 93 103 107 112 117	06
Q31	51 74 96 97 98	05
Q32	42 51 81 137 198 241 242	07
Q33	0	00
Q34	0	00
Q35	20 22	02
Q36	8 39 87 99 103 112 114 121 122 136 235	11
Q37	40	01

Appendix B: Continued

Query no.	Record no.	Total records
Q38	125 143	02
Q39	148 174	02
Q40	75 76 77 78 89 101 133	07
Q41	79 105 147 163	04
Q42	13 37 144 145 238 240	06
Q43	106 136 144 145 146 151 157 168	08
Q44	11 12 29 33 50 126 128 137 144 145 146 148 149 153 155 157 162 202 206 237 242	21
Q45	81 137 142 143 154 198	06
Q46	14 75	02
Q47	0	00
Q48	1 17 83 87 99 107 110 111 116 120 128 138 141 142 143 144 145 146 147 149 156 159 190 191 196 197	26
Q49	164 189	02
Q50	20 22 25	03
Q51	113 123 150 189 193 194 195 233	08
Q52	190	01
Q53	88 182	02
Q54	22 25 95 139 201	05
Q55	178 179 180 181 182 183 184 185 186	09
Q56	13 96 97	03
Q57	17 43 65 64 117 187	06
Q58	82 92 104 118 132	05
Q59	8 15 43 45 46 47 48 49 50 171	10
Q60	2 4 85 94 105 108 109 119 142 181 203 234	12

References

Abu-Salem, H. (1992). *A microcomputer based Arabic bibliographic information retrieval system with relational thesauri.* Unpublished doctoral dissertation, Computer Science Department, Illinois Institute of Technology, Chicago.

Al-Fedaghi, S., & Al-Anzi, F. (1989). A new algorithm to generate Arabic root-pattern forms. *Proceedings of the 11th National Computer Conference and Exhibition,* March, (pp. 391–400). Dhahran, Saudi Arabia: King Fahd University of Petroleum and Minerals.

Ali, N. (1988). *Computers and the Arabic language.* Cairo, Egypt: Al-Khat Publishing Press, Ta'reep.

Al-Kharashi, I. (1991). *Micro-Airs: Microcomputer based Arabic information retrieval system, comparing words, stems, roots as index terms.* Unpublished doctoral dissertation, Computer Science Department, Illinois Institute of Technology, Chicago.

Al-Kharashi, I., & Evens, M. (1994). Words, stems, and roots in an Arabic information retrieval system. *Journal of the American Society for Information Science, 45,* 548–560.

Al-Shalabi, R. (1996). *Design and implementation of an Arabic morphological system to support natural language processing.* Unpublished doctoral dissertation, Computer Science Department, Illinois Institute of Technology, Chicago.

Daniel, W. W. (1990). *Applied nonparametric statistics* (2nd ed., pp. 38–56). Boston: PWS-Kent.

Harman, D. (1993). Overview of TREC-1. *Proceedings of the ARPA Human Language Technology Workshop* (pp. 61–65). Princeton, NJ, np.

Hilal, Y. (1985). Morphological analysis of Arabic speech. *Computer Processing of the Arabic Language* (Workshop Papers), Vol. I. April, Kuwait City: University of Kuwait.

Hilal, Y. (1989). Automatic processing of the Arabic language and its application. *Proceedings of the First Kuwait Computer Conference* (pp. 145–171). March, Kuwait City: University of Kuwait.

Keen, E. M. (1972). Prospects for classification suggested by evaluation tests carried out 1957–1970. In A. Maltby (Ed.), *Classification in the 1970s* (pp. 193–210). Hamden, CT: Linnet Books.

Lancaster, H. (1978). *Information retrieval systems: Characteristics, testing and evaluation* (2nd ed., pp. 193–210). New York: Wiley.

Lennon, M., Peirce, D., Tarry, B., & Waillett, P. (1981). An evaluation of some conflation algorithms for information retrieval. *Journal of Information Science, 32,* 177–188.

Lovins, J. B. (1968). Development of stemming algorithm. *Mechanical Translation and Computational Linguistics, 11,* 22–31.

Luhn, H. P. (1957). A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of Research and Development, 1,* 309–317.

Morfeq, A. H. (1990). *BAYAN: A text database management system for Arabic engineering documents.* Unpublished doctoral dissertation, Computer Science Department, University of Colorado at Boulder.

Omar, A. M. (1984). *Al-Naho Al-Asasy.* Kuwait: Zat-Elsalasel Bookshop.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14,* 130–137.

Saliba, B., & Al-Dannan, A. (1989, March). Automatic morphological analysis of Arabic: A study of content word analysis. *Proceedings of the First Kuwait Computer Conference* (pp. 3–5), March, Kuwait City, University of Kuwait.

Salton, G. (1971). *The SMART retrieval system experiments in automatic document processing.* Englewood Cliffs, NJ: Prentice Hall.

Salton, G. (1975). *A theory of indexing.* Regional Conference Series in Applied Mathematics, No. 18. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer.* Reading, MA: Addison-Wesley.

Salton, G., & McGill, M. J. (1983). *Introduction to modern information retrieval.* New York: McGraw-Hill.

01 Systems (1988). *Nafitha User's guide and advanced programmers reference* (Release 3.0). Manama, Bahrain: 01 Systems.

Yahya, A. H. (1989). *On the complexity of the initial stages of Arabic text processing* (pp. 1–7). Birzeit University, Birzeit, West Bank.