Bassam Hammo	**Azzam Sleit**	**Mahmoud El-Haj**
Jordan University King Abdullah II School for Information Technology Computer Information Systems P.O.Box 13660 Amman, 11942 Jordan b.hammo@ju.edu.jo +96265355000 Ext 22606	Jordan University King Abdullah II School for Information Technology Computer Science Department Amman, 11942 Jordan Azzam.Sleit@ju.edu.jo	Jordan University King Abdullah II School for Information Technology Computer Information Systems Amman, 11942 Jordan madhajj@hotmail.com

Abstract

Modern Arabic text is written without diacritical marks (short vowels), which causes considerable ambiguity at the word level in the absence of context. Exceptional from this is the Holy Quran, which is endorsed with short vowels and other marks to preserve the pronunciation and hence, the correctness of sensing its words. Searching for a word in vowelized text requires typing and matching all its diacritical marks, which is cumbersome and preventing learners from searching and understanding the text. The other way around, is to ignore these marks and fall in the problem of ambiguity. In this paper, we provide a novel diacritic-less searching approach to retrieve from the Quran relevant verses that match a user's query through automatic query expansion techniques. The proposed approach utilizes a relational database search engine that is scalable, portable across RDBMS platforms, and provides fast and sophisticated retrieval. The results are presented and the applied approach reveals future directions for search engines.

Keywords: Arabic Information Retrieval, Searching the holy Quran, Diacritic Text, Question Answering Systems, Arabic Stemming, Arabic Thesaurus

Effectiveness of Query Expansion in Searching the Holy Quran

Abstract

Modern Arabic text is written without diacritical marks (short vowels), which causes considerable ambiguity at the word level in the absence of context. Exceptional from this is the Holy Quran, which is endorsed with short vowels and other marks to preserve the pronunciation and hence, the correctness of sensing its words. Searching for a word in vowelized text requires typing and matching all its diacritical marks, which is cumbersome and preventing learners from searching and understanding the text. The other way around, is to ignore these marks and fall in the problem of ambiguity. In this paper, we provide a novel diacritic-less searching approach to retrieve from the Quran relevant verses that match a user's query through automatic query expansion techniques. The proposed approach utilizes a relational database search engine that is scalable, portable across RDBMS platforms, and provides fast and sophisticated retrieval. The results are presented and the applied approach reveals future directions for search engines.

1 Introduction

Arabic is a Semitic language. Like Hebrew, the orthography of the Arabic language has two main parts: alphabets that represent the consonant sounds, and diacritical marks that represent the short vowels and cause the variations in pronunciation. The diacritics are written above and below the characters. For instance, and according to (Kirchhoff and Vergyri, 2005), the root كتب (ktb) "to write" with the presence of the diacritics (short vowels) has 21 valid interpretations. For example, it could be interpreted using the pattern (made of the three consonant root and short vowels), فَعَلَ (fa3ala) (pronounced كَتَبَ "he wrote"), or using the pattern فُعُلٌ (fo3oln) (pronounced كُتُبٌ "books"). An analysis of 23,000 Arabic scripts by (Debili et al., 2002) showed that there is an average of 11.6 possible ways to assign diacritics for every non-vowelized word. However, modern Arabic scripts that appear on the Internet and most of the books aiming grownups are written without diacritics, which cause ambiguity in sensing the words even for educated readers. The only way to disambiguate the words is to locate the words within the context. Exceptional from this phenomenon is made for the Quran, religious scripts, books for children, children with reading difficulties, and books for foreign students learning Arabic. The major diacritic-less Arabic scripts available on the Internet will prevent two groups of people from accessing their contents. The first group is the visually impaired people, while the second is people with learning disabilities. Both groups rely on computer-based applications of text synthesis and voice recognition. Unfortunately, the success of the Arabic voice applications is highly dependent on the presence of vowelized text to be able to read the words correctly without any ambiguity. Many researches have been carried out to restore diacritics automatically to help such applications (El-Sadany and Hashish, 1988; Gal, 2002; Emam and Fisher, 2004; Zitouni et al., 2006). In addition, research has been also done to improve Arabic Information Retrieval (IR) through deploying techniques and methodologies as morphology among others to improve the recall and the precision (Hmeidi et al., 19997; Abu-Salem et al., 1999; Alsamara et al. 2003; Zitouni et al., 2005).

2 Motivations

The Quran is the sacred book of the followers of Islam (also known as Muslims). It contains teaches of the Islamic religion. Muslims believe that the Quran is the eternal and literal word of Allah (God) revealed in its original Arabic language to Prophet Muhammad who memorized its verses and taught them as they were revealed to his companions. Muslims also believe that Quran covers all aspects of life for all mankind.

The scripts of the Quran are vowelized to prevent reciters (readers) from misspelling a word, and hence changing its meaning, which tremendously changed by the vowels. However, most researches in the field of Arabic IR did not pay much attention to the problem of searching and retrieving vowelized text. Most accomplished works even suggested removing the diacritics in the preprocessing step and unifying the content of the inverted list (Buckwalter, 2002). This is because the majority of the text available on the web is not-vowelized (or using few diacritical marks) as well as, typing and matching words with diacritics is cumbersome.

We still believe that the Quran haven't been served well and made available for researchers to explore its content. In this paper, we proposed an approach to search the words of the Quran without being worried about typing the diacritics. On the contrary, we made the search process very interesting through automatic query expansion using a stemmer. All what the user has to do, is to type in the words and the query is automatically augmented with all morphological variation of the query's words to expand the search. Also, we investigated the effectiveness of applying a thesaurus of semantic classes to expand the search. The obtained results are promising and open directions for enhancing the capabilities of search engines, and other applications, such as, question answering, and information extraction from the Quran.

3 The Search Engine

An Information Retrieval Engine to experiment with Arabic document retrieval is described in (Hammo et al., 2002). Later, this engine has been modified to support passage retrieval for an Arabic open-domain question answering system (Hammo et al., 2004).

As most of the current search engines, our system is based on the famous vector space model (Salton, 1983; 1989). It takes a query in the Arabic language and attempts to provide a ranked list of verses as an output. The main components of the system include: *Verse Splitter*, *Tokenizer*, *Stemmer* and *Thesaurus*. The data flow of our system is depicted in Figure 1.

First, the Quran chapters (*suras*) are divided into verses (*ayat*) through the *Splitter* module. The *Tokenizer* divides the verses into words (tokens), while a rule-based *Stemmer* uses morphological heuristics to determine the morphological root of a given word. In this system we provide three types of indexes: the *Vowelized-Word* Index, the *Non-Vowelized-Word* Index, and the *Root* Index, which are used to manipulate the data. Finally, a *Thesaurus* of semantic word classes is used to expand the user's query. In the following sections we describe the implementation and the basic processing outline of the search engine.

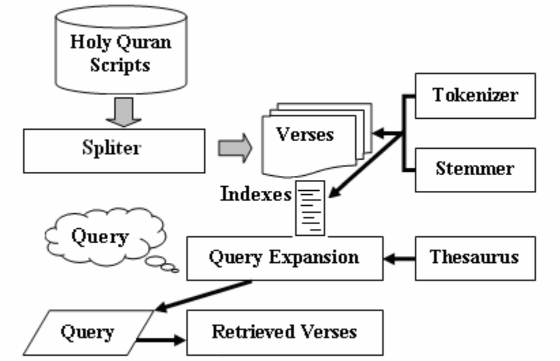

Figure1. Data Flow

3.1 Implementation of the Search Engine

An Information Retrieval (IR) system can be constructed using a relational database management system (RDBMS) (Lundquist et al., 1997). Designing an IR system on the relational model retains the benefits of being scalable, provides fast and sophisticated retrieval, portable across RDBMS platforms as well as being able to use the security and integrity features built in the RDBMS system (Lundquist et al., 1997).

In our work, we adapted the idea of integrating an RDBMS with an IR system to store and manipulate the Quran scripts. The data model of the engine is depicted in Figure 2. It contains the following database relations:

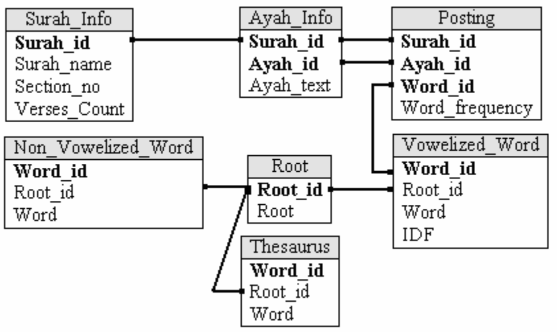

Figure 2. The Relational Database Model

- Surah_Info_Table (**Surah_id**, Surah_name, Section_no, Verses_count): to store chapter information (one row per surah (chapter)).
- Ayah_Info_Table (**Surah_id, Ayah_id**, Ayah_text): to store all the verses extracted from the Quran collection (one row per verse).
- Vowelized_Word_Table (**Word_id**, Root_id, Word, IDF): to store all distinct vowelized words from the Quran collection, the Root_id and the inverse document frequency (IDF) of each word (one row per word).
- Non_Vowelized_Word_Table (**Word_id**, Root_id, Word): to store all distinct non-vowelized words from the Quran collection, the Root_id of each word (one row per word).
- Root_Table (**Root_id**, Root): to store the distinct roots of the words from the Quran collection (one row per root).
- Posting_Table (**Surah_id, Ayah_id**, **Word_id**, Word_frequency): to post all occurrences of the words extracted from the Quran scripts (one row per posted word).
- Thesaurus_Table (**Word_id**, Root_id, Word): to store semantic classes of words from the Quran scripts (one row per word).

4 Processing Outline in the Search Engine

4.1 Verses Preprocessing

The scripts of the Quran, in its current format, are back to the *Rashedi Caliph Othman Bn Affan*, who ordered the companions of the Prophet, Mohammad, to gather the written scripts in one book. This is the book that all Muslims are reading today with its original format as it was revealed.

To make the Quran suitable for searching, verses preprocessing were very essential before indexing. First, we start with obtaining the verses from the verse *Splitter* and getting rid of the verse number and the ۝ (U+06DD) symbol, which marks the end of the verse. Other symbols like (۞ ۩)are used to organize the Quran into parts and sections and thus have to be removed. Finally, a set of diacritics such as (ۖ ۗ ۘ ۙ ۚ ۜ) are used for reciting the Quran also removed before the tokenization process takes place.

4.2 Tokenization

The tokenization process starts with acquiring each surah (chapter) from the Surah_Info_Table. Next a verse *Splitter* module is triggered to split each chapter into verses at the verse boundary symbol "۝". The extracted verses are then tokenized to extract their words. We designed and implemented a tokenizer that extracts words at any white space or punctuation characters. Since the Quran words are bounded only by spaces, the tokens were extracted easily, and correctly without any mistakes with(100%) accuracy.

4.3 Building the Inverted Lists

The indexes of the search engine are built during the tokenization process. The system builds three indexes as tokens start arriving: one index for the vowelized words, a second one for the non-vowelized words and a third one for roots.

4.3.1 The Vowelized-Words Index

This inverted list contains all distinct words, along with their frequencies, obtained from tokenizing all the verses and inserted in this index without preprocessing (i.e. keeping their diacritics intact). This index is not used directly for searching words, as typing and matching the diacritics is cumbersome. Nevertheless, this index is used to fetch all morphological variations of the query's words during the automatically expansion process. The index contains 19273 distinct words (see Table 1).

4.3.2 The Non-Vowelized Index

This index contains all distinct words of the Quran after being processed by stripping their diacritical marks and unifying the alef-hamza character to straight alef (i.e. converting آأإ to ا). The index contains 14918 distinct words. This diacritic-less index is about (22.6%) reduction of the *Vowelized-Words* Index. The *Non-Vowelized* Index is considered the main index of the search engine and is used to fetch the query's bag-of-words. So at least one morphological form of each word in the Quran is indexed. The other words are obtained from the *Vowelized-Words* Index through the query expansion process.

4.3.3 The Root Index

This inverted list contains all distinct roots of the Quran after being automatically stemmed and verified against manually extracted roots from the Quran (Khadir, 2005). The verification is made for completeness and correctness. A total of 1767 distinct root (including proper names appearing in the Quran) were identified and referenced in the *Vowelized-Words* and *Non-Vowelized-Words* indexes as well as the *Thesaurus* Index. The results of the stemmer will be discussed in the experimental section. This index is about (88.2%) reduction of the *Non-Vowelized* Index.

4.3.4 The Thesaurus

It has been argued that Arabic does not have synonyms, especially for the Quran, the way they are used for other languages. Each word in the Quran, although for words of the same meaning, has been chosen to fit a syntactical and semantical role that cannot be obtained by another word. However, in our work we benefited from the work done by (Kubaisi, 2006) to collect words from the Quran and group them into semantical classes for the purpose of query expansion.

4.4 Stemming

Stemming is the task of correlating several words onto one base form. On average, stemming increases similarities between documents and queries because they have additional common terms after stemming which do not exist before. Stemming has a relatively low processing cost. It uses morphological heuristics to remove affixes from words before indexing. It reduces the index size, and it usually slightly improves results (Strzalkowski and Vauthey, 1992). This makes it very attractive for use in IR.

Arabic stemming is more complicated compared to the English language. Major words of the Arabic language are constructed from the three consonant roots by following fixed patterns. Patterns include prefixes, infixes and suffixes to indicate number, gender and tense. Arabic stemming is the process of removing all affixes from a word to extract its root. A stemmer for Arabic, for example, should identify the string, *kateb* كاتب (*writer*), *ketab* كتاب (*book*), *maktabah* مكتبه (*library*), *maktab* مكتب (*office*), as one base form *ktb* كتب (*he wrote*).

In our research, we used a rule-based stemmer developed to experiment with Arabic passage retrieval and question answering (Hammo et al. 2004). The stemmer performed reasonably with more than (96%) accuracy.

5 Experimental Design

5.1 The Test Collection

The Quran consists of 114 chapters (suras) and contains a total of 6236 verses (ayat). Each surah is generally known by a name. Table 1 shows some statistics about the Holy Quran.
For this paper, we used the Quran and a collection of forty non-vowelized, one-word queries obtained from 5 college students. Each one has been asked to provide 10 words that he would like to search in the Quran. Duplicates have been removed and the final list of 40 queries is given in Table 2.

Table 1. Statistics of the Holy Quran

Number of pages	604
Number of chapters	114
Number of parts	30
Number of sections	60
Number of verses	6236
Number of words	77845
Number of distinct vowelized words	19273
Number of distinct non-vowelized words	14918
Number of distinct roots*	1767
* also include proper names of non Arabic origins	

Table 2. Query Collection

Q#	Word	Q#	Word
1	اثاث	21	بزغ
2	اثار	22	بعث
3	اجاج	23	توبة
4	احب	24	جدار
5	الابتر	25	جلباب
6	الابتسام	26	جمل
7	التفكير	27	جميل
8	الجن	28	جوع
9	الرحمة	29	حصاد
10	السجن	30	حظ
11	السحاب	31	خشوع
12	السمع	32	زوج
13	الصافي	33	سابق
14	الصبر	34	سباحة
15	الضعف	35	ستر
16	الفساد	36	طاعة
17	المجرم	37	ظلم
18	المنهج	38	قبر
19	النصر	39	لهب
20	برهان	40	وفاء

5.2 Results form the Tokenizer

We started with one chapter (surah) after being preprocessed as explained earlier. The tokens were manually examined to verify the correctness of the tokenization process. The *Tokenizer* successfully isolated all the words and passed them for the stemmer to extract their roots.

5.3 Results from the Stemmer

In this experiment, we used a stemmer described in (Hammo et al., 2004). We run the stemmer against the *Non-Vowelized* Index and the results were close to (91%) accuracy. We observed that most of the failing cases were due to stemming proper names (such as the names of prophets and names of angels), ancient cities, places and people, numerals, as well as words with doubled characters (represented using the diacritic shada (ّ)). To verify the correctness of the roots, we compared the automatically generated roots with a list of manually extracted and verified roots compiled by (Khadir, 2005). We manually corrected the mistaken ones and added the missing ones so the two lists became identical. Finally, the *Vowelized* and the *Non-Vowelized* Indexes and the *Thesaurus* Index have been connected with this *Root* Index.

5.4 Experimenting with the Search Engine

5.4.1 Searching Diacritic-less Words

An Arabic search engine was designed and implemented to perform query processing using standard SQL over the indexing techniques mentioned earlier. Table 3 shows the results of running 40 one-word diacritic-less queries over the search engine. The system fails in many cases, where an exact match could not be found. Table 4 shows a sample of what is found in the *Non-Vowelized-Word* Index that could match some of these queries. It is clear that the failure in most of the cases is due to the missing diacritics. Figure 3 shows the output of this experiment.

Table 3. Non-Stemmed Arabic words

Q#	Word	Verses retrieved	Q#	Word	Verses retrieved	Q#	Word	Verses retrieved	Q#	Word	Verses retrieved
1	اثاث	0	11	السحاب	2	21	بزغ	0	31	خشوع	0
2	اثار	1	12	السمع	2	22	بعث	1	32	زوج	1
3	اجاج	1	13	الصافي	0	23	توبة	1	33	سابق	2
4	احب	3	14	الصبر	0	24	جدار	0	34	سباحة	0
5	الابتر	1	15	الضعف	1	25	جلباب	0	35	ستر	0
6	الابتسام	0	16	الفساد	3	26	جمل	0	36	طاعة	1
7	التفكير	0	17	المجرم	1	27	جميل	1	37	ظلم	3
8	الجن	3	18	المنهج	0	28	جوع	1	38	قبر	0
9	الرحمة	3	19	النصر	1	29	حصاد	0	39	لهب	1
10	السجن	3	20	برهان	2	30	حظ	2	40	وفاء	0

Table 4. Sample of words found in Index

Q#	Word	Word in Index
1	اثاث	اثاثا
2	اثار	اثارهم، اثارهما
14	الصبر	صبر، صبرا
21	بزغ	بازغا
24	جدار	جدارا
29	حصاد	حصاده
31	خشوع	خاشعا
38	قبر	اقبره

5.4.2 Effectiveness of Query Expansion

Query Expansion (QE) can be defined as the process of reformulating the query's bag-of-words to overcome the problem of mismatching potential documents and improving the performance of a search engine by including in the results documents which are more relevant (of better quality) , or at least equally relevant (Oiu and Frei, 1993; Vectomova and Wang, 2006). Without query expansion, the documents which have the potential to be relevant to the user's query would not be retrieved. Many QE techniques have been investigated in the information retrieval literature. They include:

- Query expansion through synonymy. This is performed through finding words synonyms in the search query, and searching for the synonyms as well.
- Query expansion through stemming. This is performed by adding the various morphological forms of the input query, and searching for all forms as well.
- Query expansion through word sensing. This is performed through sensing the words to resolve ambiguity from a specialized database such as the WordNet.
- Query expansion through fixing spelling errors, and automatically searching for the corrected form in the search query.
- Query expansion through paraphrasing. This is performed by rewriting the terms of the original query. Some query expansion techniques such as synonymy and stemming have been criticized for increasing the total recall on the expense of lowering the precision. Other techniques like word sense disambiguation (wsd) tends to increase the precision. However, despite the increase in the recall, augmenting the user's query with synonyms and morphological variations and ranking the occurrences of the query's words, cause documents with more approximate terms to migrate near the top of the ranked list, and hence, leading to a higher performance.

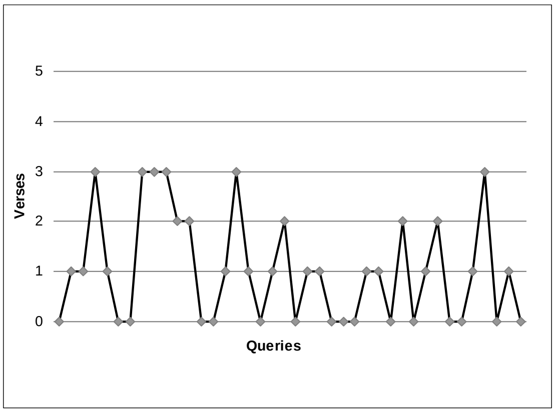

Figure 3. Results of Searching Bag-of-Words

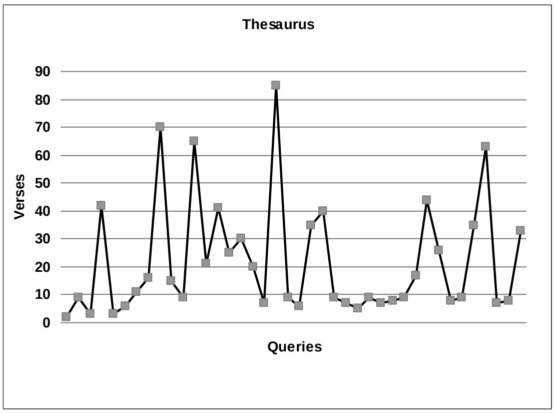

Figure 4. Results of Searching Using Stemmer

5.4.2.1 Query Expansion through Stemming

The goal of query expansion in this regard is to increase recall, and hence, precision can potentially increase. Most of the time using the exact words for searching will not give good results and some times no results at all as was shown in Table 3. This is because of the discrepancy in morphological structure between the words in the corpus and the query's words, which most of the time end up with no-matching. We believe that users interested in a word like كتب (writes, 3PSNG/VBD), also are interested in words such as: كتابة (writing), مكتب (office), مكتبة (library), and مكتوب (some thing written). Nevertheless, all previous words can be correlated to the root (كتب) to be retrieved. Therefore, in our search engine query expansion is done automatically to find all verses with words that correlated to the roots of the query's bag-of-words. The process starts with applying the stemmer on the bag-of-words. For each root we obtain from the query, we search the *Root* Index and fetch from the *Vowelized-Words* Index all the diacritic words that are interrelated with this root. The new query of diacritic words is submitted again to search the *Posting* Table for the occurrences of all verses having these words.

Running the experiment with expanding the search through stemming was very efficient and satisfactory. Table 5 shows the results of applying the stemmer, while Figure 4 shows the improvement over the 40 queries. The obtained results made this technique very practical, especially for a question answering system (Hammo et al., 2004).

5.4.2.2 Query Expansion through Thesaurus

By expanding a search query to search for the synonyms of a user entered term, the recall is also increased, and sometimes at the expense of precision. When used for information retrieval, terms and their synonyms are augmented in the query vector and a new search begins. We compiled a thesaurus of the Quran words and group them semantically. The words related to the queries and the new findings after expanding the search automatically are given in Table 6, while Figure 5 shows the improvement over the 40 queries.

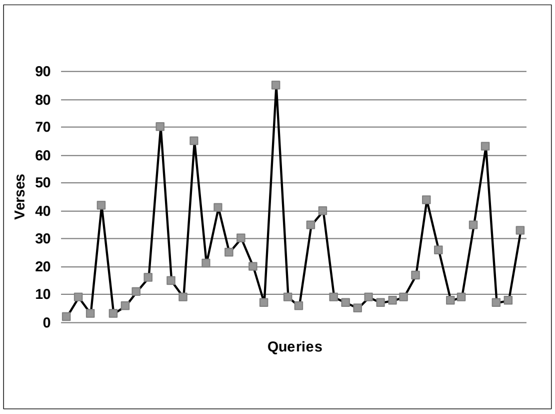

Figure 5. Results of Searching Using Thesaurus

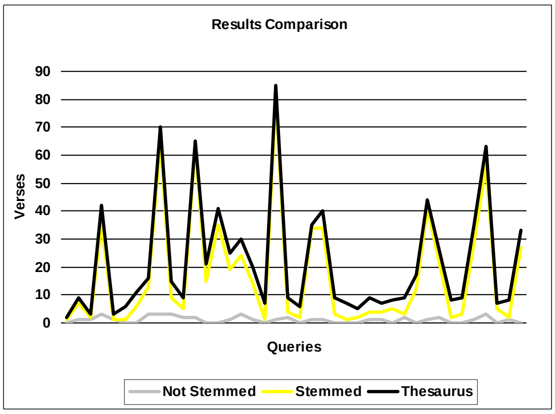

Figure 6. Comparison of Searching Techniques

6 Conclusion

In this paper, we explained the problem of searching diacritized text and provide a solution for the searching problem through indexing. We investigated the use of query expansion on searching the Quran in the absence of diacritics, where queries are automatically augmented with related terms extracted from a vowelized index by applying a stemmer and a thesaurus of semantic classes. We conducted a set of experiments on searching words from the Quran and we concluded that query expansion for searching Arabic text is promising and it is likely that the efficiency can be further improved. Figure 6 shows a comparison of applying the different techniques on the 40 queries. In the future, we plan to use these ideas in improving a question answering system for the Arabic language.

Table 5. Stemmed Arabic words

Q#	Word	Found	Root	#Distinct words of this root	#Relevant verses	Q#	Word	Found	Root	#Distinct words of this root	#Relevant verses
1	اثاث	0	أثث	1	1	21	بزغ	0	بزغ	2	2
2	اثار	1	أثر	15	7	22	بعث	1	بعث	34	34
3	اجاج	1	أجج	2	2	23	توبة	1	توب	34	34
4	احب	3	حبب	37	36	24	جدار	0	جدر	4	3
5	الابتر	1	بتر	1	1	25	جلباب	0	جلب	2	1
6	الابتسام	0	بسم	1	1	26	جمل	0	جمل	7	2
7	التفكير	0	فكر	6	6	27	جميل	1	جمل	7	4
8	الجن	3	جنن	46	13	28	جوع	1	جوع	4	4
9	الرحمة	3	رحم	65	65	29	حصاد	0	حصد	5	5
10	السجن	3	سجن	9	9	30	حظ	2	حظظ	3	3
11	السحاب	2	سحب	6	5	31	خشوع	0	خشع	12	12
12	السمع	2	سمع	63	63	32	زوج	1	زوج	41	41
13	الصافي	0	صفو	16	15	33	سابق	2	سبق	22	22
14	الصبر	0	صبر	35	35	34	سباحة	0	سبح	31	2
15	الضعف	1	ضعف	34	19	35	ستر	0	ستر	3	3
16	الفساد	3	فسد	24	24	36	طاعة	1	طوع	47	29
17	المجرم	1	جرم	16	14	37	ظلم	3	ظلم	64	57
18	المنهج	0	نهج	1	1	38	قبر	0	قبر	5	5
19	النصر	1	نصر	81	81	39	لهب	1	لهب	2	2
20	برهان	2	برهن	4	4	40	وفاء	0	وفي	42	27

Table 6. Semantic Classes of Words from the Holly Quran

Q#	Word	Class	Semantic Groups								Found	#Relevant Verses
1	اثاث	**أثث**	متاع								1	2
2	اثار	**أثر**	علامات	اية							2	9
3	اجاج	**أجج**	ملح								1	3
4	احب	**حبب**	الشهوة	الهوى	الحب	الشغف	الغرام	الهيام	الشهوة	الهوى	6	42
5	الابتر	**بتر**	بتك	قطع							2	3
6	الابتسام	**بسم**	الضحك	السخرية	الاستهزاء	الازدراء	الاستخفاف				5	6
7	التفكير	**فكر**	المميز	الوازع	المتدبر	المفكر	الرشيد				5	11
8	الجن	**جنن**	الابالسة	العفاريت	الشياطين						3	16
9	الرحمة	**رحم**	البر	التفضل	الانعام	الاحسان	المنة				5	70
10	السجن	**سجن**	حبس	امساك	توقيف	اثبات	حجر	رباط			6	15
11	السحاب	**سحب**	الغمام	العارض	الظلة	الصيب					4	9
12	السمع	**سمع**	انصت	استمع							2	65
13	الصافي	**صفو**	الزاكي	الطيب	الطاهر	الممحص	المصنوع	المخلص			6	21
14	الصبر	**صبر**	الحلم	الصوم	العفة	القناعة	كظم	الغيظ			6	41
15	الضعف	**ضعف**	العجز	الوهن	الوهي	الفتور	الكسل	التثاقل			6	25
16	الفساد	**فسد**	الخبث	الرجز	الرجس	النجس	القبح	السوء			6	30
17	المجرم	**جرم**	جبار	قاهر	متكبر	عالي	عتل	عاتي			6	20
18	المنهج	**نهج**	امام	صراط	طريق	سبيل	فج	جدد			6	7
19	النصر	**نصر**	الظفر	الغلبة	الفتح	الفوز					4	85
20	برهان	**برهن**	بينة	اية	الاء	حجة	بصائر				5	9

References

Abu-Salem H., Al-Omari M., and Evens M. (1999). Stemming Methodologies over Individual Query Words for an Arabic Retrieval System. *Journal of the American Society for Information Science (JASIS)*, Vol. 50, No. 6, pp. 524-529.

Alsamara K., Abu-Salem H., Abuleil S., and Hammo B. (2003). Building Automated Indexes to Improve Arabic Information Retrieval. Proceedings of 2003 International Conference on Mnagement Science & Engineering, Georgia, USA, 2573-78.

Buckwalter T. (2002). Buckwalter Arabic morphological analyzer version 1.0. Technical report, Linguistic Data Consortium, LDC2002L49 and ISBN 1-58563-257-0.

Debili F., Achour H., and Souissi E. (2002). De l'etiquetage grammatical a' la voyellation automatique de l'arabe. *Technical report, Correspondances del'Institut de Recherche sur le Maghreb Contemporain 17.*

El-Sadany T., and Hashish M. (1988). Semi-automatic vowelization of Arabic verbs. In *10th NC Conference*, Jeddah, Saudi Arabia.

Emam O., and Fisher V. (2004). A hierarchical approach for the statistical vowelization of Arabic text. *Technical report*, IBM patent filed, DE9-2004-0006, US patent application US2005/0192809 A1.

Gal Y. (2002). An HMM approach to vowel restoration in Arabic and Hebrew. In *ACL-02 Workshop onComputational Approaches to Semitic Languages*.

Hammo, B., Abu-Salem, H., Lytinen, S., and Evens, M. (2002). QARAB: A Question Answering System to Support the Arabic Language. *Workshop on Computational Approaches to Semitic Languages. ACL 2002*, Philadelphia, PA, July. 55-65.

Hammo, B., Abuleil, S., Lytinen, S., and Evens, M. (2004). Experimenting with a Question Answering System for the Arabic Language. *Journal of Computers and the Humanities*. (38) 379-415.

Hmeidi, I., Kanaan, G., and Evens, M. (1997). Design and Implementation of Automatic Indexing for Information Retrieval with Arabic Documents. *Journal of the American Society for Information Science*. Vol. 48, No. 10, pp. 867-881.

Khadir M. (2005). Qural lexicon. Available at http://www.almishkat.com/words/book.htm

Kirchhoff K., and Vergyri D. (2005). Cross-dialectal data sharing for acoustic modeling in Arabic speech recognition. *Speech Communication*, 46(1):37–51.

Kubaisi A. (2006). Quran words. Available at http://www.islamiyyat.com/kalema.htm.

Lundquist, C., Frieder, O., Holmes, D., and Grossman, D. (1997). A Parallel Relational Database Management System Approach to Relevance Feedback in Information Retrieval. *Journal of the American Society of Information Science (JASIS)*, Vol. 50, No. 5. pp. 413-426.

Qiu Y., and Frei H. (1993). Concept Based Query Expansion. In *Proceedings of SIGIR-93, 16th ACM International Conference on Research and Development in Information Retrieval*, Pittsburgh, SIGIR Forum, ACM Press.

Salton, G., and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill Book Company, New York.

Salton, G. (1989). *Automatic Text Processing* -*The Transformation Analysis and Retrieval of Information by Computer*. Addison Wesley, MA.

Strzalkowski, T. and Vauthey, B.(1992), Information Retrieval Using Robust Natural Language Processing, In *Proceedings of ACL-92*, pages 104–111.

Vectomova O. and Wang Y. (2006). A study of the effect of term proximity on query expansion. *Journal of Information Science* 32 (4): 324–333.

Zitouni I., Sorensen J., Luo X., and Florian R. (2005). The impact of morphological stemming on Arabic mention detection and coreference resolution. In *Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages*, pages 63–70, Ann Arbor, June.

Zitouni I., Sorensen J., and Sarikaya R. (2006). Maximum Entropy Based Restoration of Arabic Diacritics. *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 577–584, Sydney.