On Arabic Search: Improving the Retrieval Effectiveness via a Light Stemming Approach

Mohammed Aljlayl
Computer Technology Department
Riyadh College of Technology
Riyadh, Saudi Arabia
aljlayl@rct.edu.sa

Ophir Frieder
Information Retrieval Laboratory
Illinois Institute of Technology
Chicago, IL 60616
ophir@ir.iit.edu

ABSTRACT
The inflectional structure of a word impacts the retrieval accuracy of information retrieval systems of Latin-based languages. We present two stemming algorithms for Arabic information retrieval systems. We empirically investigate the effectiveness of surface-based retrieval. This approach degrades retrieval precision since Arabic is a highly inflected language. Accordingly, we propose root-based retrieval. We notice a statistically significant improvement over the surface-based approach. Many variant word senses are based on an identical root; thus, the root-based algorithm creates invalid conflation classes that result in an ambiguous query which degrades the performance by adding extraneous terms. To resolve ambiguity, we propose a novel light-stemming algorithm for Arabic texts. This automatic rule-based stemming algorithm is not as aggressive as the root extraction algorithm. We show that the light stemming algorithm significantly outperforms the root-based algorithm. We also show that a significant improvement in retrieval precision can be achieved with light inflectional analysis of Arabic words.

Keywords
Arabic, stemmer, information retrieval, algorithms

1. INTRODUCTION
With the rapid growth of the Internet in recent years, the World Wide Web (WWW) has become one of the most popular mediums for the dissemination of electronic Arabic documents. Automatic mediation of access to Arabic Web pages is becoming an increasingly important problem.

CIKM'02, November 4-9, 2002, M clean, Virginia, USA.

In Arabic morphology, most Arabic morphemes are comprised of a basic form of a word that is called the root, to which many affixes can be attached to form Arabic words. The inflectional structure of a word impacts the retrieval precision. We present two stemming methodologies for Arabic texts. We investigate the effectiveness of using the surface form of the words. This approach degrades the retrieval effectiveness due to the fact that Arabic is a highly inflected and derived language. Accordingly, we propose the root-based retrieval. We notice a significant improvement over the surface form of the word, which we name as surface-based approach. Many words with different meaning are based on the same root; thus, the root-based algorithm creates invalid conflation classes and results in ambiguous queries. To resolve ambiguity, we propose a light-stemming algorithm for Arabic texts. We show that the light stemming algorithm significantly outperforms the root-based algorithm. The reason behind this improvement is that the root-based algorithm conflates a lot more terms, which degrades the performance by introducing extra noise.

In Section 2, we briefly review the structure of the Arabic language and overview prior work in Arabic information retrieval. Surface-based retrieval in Arabic is described in Section 3. The proposed root-based retrieval method for Arabic is presented in Section 4. A novel stemming technique is presented in Section 5. In Section 6, we present the experimental environment, and then we discuss the results in Section 7. We conclude our study in Section 8.

2. BACKGROUND
2.1 Arabic Language Structure
Arabic, one of the six official languages of the United Nations, is the mother tongue of 300 million people [6]. Unlike Latin-based alphabets, the orientation of writing in Arabic is from right-to-left. The Arabic alphabet consists of 28 letters. As discussed in [17], the Arabic alphabet can be extended to ninety by additional shapes, marks, and vowels. Most Arabic words are morphologically derived from a list of roots. The root is the bare verb form; it can be triliteral, quadriliteral, or pentaliteral. Most of these roots are made up of three consonants. The Arabic language uses a root-and-pattern morphotactics; pattern can be thought of as template adhering to well-known rules. These patterns generate nouns and verbs. Roots are interdigitated with the patterns to form Arabic surface forms.

2.2 Prior Work

Researchers have studied the impact of stemming on the retrieval effectiveness of Latin-based languages such as English. Several different techniques were proposed for stemming English text. One of the simplest techniques is suffix stripping; it uses lists of suffixes to reduce words to their bare form. The most common stemming algorithms for English are Porter [16] and Lovins [13]. Kraaij and Pohlmann [11] concluded that stemming improves recall. A comparative evaluation performed by Hull [9] to investigate the retrieval precision using stemming found little precision improvement as compared to no stemming. Krovetz [12] proposed a different approach to stemming. The proposed approach takes into account the morphological structure of the word for sense disambiguation. Krovetz reported an increase of 12-34% in retrieval effectiveness in a collection where the documents and the queries are fairly short. Xu and Croft [19] proposed a novel technique to stemming; the technique relies on a corpus-based word co-occurrence statistics before the query time to construct the conflation classes.

Abu-Salem, et al. [1] manually built an Arabic dictionary that includes stems and roots,. The proposed method, which was called mixed stemming, showed an improvement over the word indexing method using both the binary and *tf-idf* weighting schemes. A recent study performed by [14] confirmed that the n-gram based retrieval achieved 0.3064 measured in average precision. They used TREC-10 topics and collection, the topics and collection are described in Section 6.

In Arabic morphology, Beesley [4] described a finite-state morphological analyzer of written standard Arabic. The underlying lexicons include about 4930 roots; the system, however, still needed additional proper names to handle multi-word expressions. Hegazi and Elsharkawi [7] implemented a morphological system for Arabic words. The system recognizes the root of a word, the morphological pattern, and its morphological category. Al-Fedagi and Al-Anzi [2] proposed a mathematical method to generate the root-pattern forms for Arabic words. The basic idea of this method is to locate the position of the root letters in the pattern and to examine the letters in the same position to verify whether the triliteral forms a valid Arabic root. Khoja [10] designed and experimented with a novel algorithm for root detection. Khoja concluded that the proposed algorithm is more effective than prior efforts [2,3].

Our work focuses on the improvement of Arabic information retrieval systems. An extensive resource of Arabic information retrieval applications as well as Arabic-English Cross-Language Information Retrieval (CLIR) can be found in [15]

3. ARABIC SURFACE-BASED RETRIEVAL

Unlike Indo-European languages such as English, the Arabic language is a highly inflected language. From an Arabic root, many surface forms can be derived. The surface forms of a word have great impact on a language like Arabic with a strong morphology since surface forms comprise at least two morphemes: a three consonantal root conveying semantic meaning and a word pattern carrying syntactic information. In addition to the different forms of the Arabic word that result from the derivational and inflectional process, most connectors, conjunctions, prepositions, pronouns, and possession forms are attached to the Arabic surface form. In Table 1, we illustrate some additional forms of the word (معلم) "*teacher*". As shown, many letters are attached to the word (معلم) while in English they appear as separable form; thus, a query that contains the Arabic word (والمعلم) "*and the teacher*" will not match any document that contains the Arabic words listed in Table 1.

Document relevancy to each query is determined primarily by the frequency of terms in both the documents and the query. Therefore, transforming different inflections and derivations of the same word to one common stem or base form creates a conflation class or group. Retrieval based on a conflation class that has all related words in one class leads to improvement in languages like English. Such conflation is more important in the Arabic language for the reason that we have mentioned earlier.

Arabic word	English counterparts
المعلم	the teacher
كالمعلم	like the teacher
للمعلم	for the teacher
بالمعلم	by the teacher
ومعلم	and teacher

Table 1. Some of the variability of the word (معلم)

4. THE ROOT-BASED STEMMER

Arabic word formation is based on an abstraction, namely, the root. These roots join with various vowel patterns to form simple nouns and verbs to which affixes can be attached for more complicated derivations.

The ultimate goal of the root-based stemmer is to extract the root of a given Arabic surface word. Root extraction involves very deep syntactic analysis of an Arabic surface form. Table 2 illustrates different variations of the Arabic root *k-t-b* (كتب) which means "*write*" in English. The derived words have similar translated meanings. The underlined font indicates to the affixes. In fact, from the same root *k-t-b,* many words can be derived. The derivational analysis reduces surface forms to the base form from which they were derived, and include changes in semantics. As presented in Table 3, the Arabic words (كتاب) "*book*", and (مكتبة) **"***library***"** stems to (كتب) "*write*", whereas they are semantically different.

We adapted Khoja's algorithm for root-based retrieval [10]. The reason behind this adoption is that Khoja's algorithm showed superiority over previous works in root detection algorithms [2,3].

Arabic Word	Meaning
يكتب	He writes
تكتب	She writes
يكتبون	They write
نكتب	We write

Table 2. Different variations derived from the root *ktb*

Arabic Word	Meaning
كـاتب	Writer
كتـاب	Book
مـكتبـة	Library
مـكتب	Office

Table 3. Variant word senses derived from same root *ktb*

Khoja's algorithm starts to remove the suffixes, prefixes, and infixes of a given Arabic surface word. After every elimination process, the algorithm checks whether the removed affixes are part of the Arabic root or they are additional letters augmented in the derivational process. The resulted stem is then checked for correctness, if any original letter is stripped out, the entire affix is returned to the word. Finally, it matches the remaining letters of the given Arabic root against list of patterns of the same length to extract the root. The algorithm can process any Arabic surface form, vowelized and nonvowelized. The prepositions, pronouns, conjunctions, interjections, and a list of Arabicized words are ignored while processing.

Patterns play an important role in Arabic lexicography and morphology. Each root can canonically combine with orthographically distinct patterns to form surface words. Patterns are used as canonical measurements for Arabic surface forms. For example, the root (كتب) is analyzed as consisting of a three-consonant root. The root (كتب), which is transliterated as ktb, is measured with pattern (فعل). The pattern (فعل) is transliterated as "fàl". "f" corresponds to the first letter "ف", à corresponds to middle letter "ع", and l corresponds to last letter "ل". The pattern preserves f, à, and l in the same order, whereas vowels and other letters can be added to form a pattern. For example, several patterns are derived from the base pattern "f à l" of the morpheme ktb. The pattern "f à alh" form the word (كتابة) by adding the vowel (ا) and letter (ه) to the morpheme ktb.

The root extraction is performed after removing the suffixes and the prefixes attached to the given word. The root extraction process starts by matching the positions of the surface word letters that correspond to a pattern. In Figure 1, we describe the essential steps in the root extraction process after extracting the letters that corresponds to the pattern "*f à l*". These letters represent the root. An additional step is performed to verify whether the extracted root is valid. This step checks the extracted root against a list of roots. The list consists of about 3800 triliteral and 900 quadliteral roots. If it is found, then the extracted root is preserved [10].

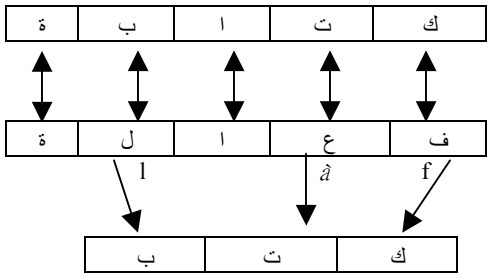

Figure 1. A step to extract the root letters of the word (كتابة) using its pattern

One of the challenges in a root extraction process is that most written Arabic words are free of diacritics, especially in a medium such as the WWW. The ignorance or incompleteness of the surface orthography makes the written text ambiguous. As in [4], the ambiguous written words make an average of five valid morphological analyzes per word.

We modified the root extraction algorithm to make it well suited for information retrieval. In addition, we added one more canonical pattern. For example, the pattern (تفاعيل) interdigitates for many nouns such as (تساهيل), (تماثيل), and others.

Double verbs in Arabic are those words whose tri-literal roots contain two identical consonants; one of the doubled letters is removed in the inflectional paradigm. For example, the tri-literal root (فنن) has two identical consonants "ن", in the second, and in the third letters of the root. Many words can be derived from the root (فنن). In many cases, the derivational process starts by eliminating one of these two identical letters. The root finder algorithm needs to be modified to handle any Arabic tri-literal word that ends with a vowel and that is preceded by two identical letters as signified by "Shaddah". Shaddah is a diacritic (ّ) which is placed above a letter to indicate that the consonant is doubled.

Triliteral roots that contain vowels (ا،ي،و) are classified as irregular roots since some vowels in these roots are altered to other vowels or removed in the derivational process. These vowels can occur in any position; in the beginning, in the middle, or at the end of such triliteral roots. An additional step is augmented to the root finder algorithm to enhance its accuracy and to accommodate handling words that end with a vowel preceded by a doubled letter as signified by a Shaddah. This step is performed for the root that contains a vowel (ا،ي،و) at the end of the tri-literal stem. This step removes the last letter of the triliteral stem, which is a vowel, and checks it against a list of words that are two letters in length. If the stem is found, then the last letter is duplicated to form a root. For example, the word (الفنية) that means "*art*" in English is reduced to (فـنـي) then the letter "ي" is removed. The resulted stem is "فن". The remaining stem "فن" is scanned against a list of words that might be duplicated. If it is found in the list of duplicate words, the algorithm duplicates the last letter, which is the "ن" letter, to form (فنن) as a valid triliteral root of (الفنية، الفني فني).

5. THE LIGHT STEMMER

The key problem of the root detector algorithm in information retrieval is that many word variants do not have similar semantic interpretations. Although these words are different in meaning, they originate from one identical root. Thus, the root-based retrieval increases word ambiguities. Inflected and derived words can have a vigorous impact on the retrieval effectiveness of any information retrieval system. Therefore, it is important to recognize the variants of word morphemes in highly inflected language such as Arabic. Word-sense disambiguation is essential to improve any Arabic information retrieval system. Our main motivation is to develop a new stemmer to minimize the sense ambiguity associated with the root-based retrieval, and to conflate the numerous semantically related words, as described in section 3, into the same conflation class.

Our hypothesis is that developing a stemming algorithm that retains the word meaning intact improves the retrieval performance of an Arabic information retrieval system. To

achieve this goal, we propose a novel technique for stemming, which is called the *light-stemming*. The aim of this technique is not to produce the linguistic root of a given Arabic surface form, rather is to remove the most frequent suffixes and prefixes.

The most common suffixation includes duals and plurals for masculine and feminine, possessive forms, and pronoun forms. For instance, the plural is formed via suffixes or via pattern modification of the nouns. In the first case, the suffix ~uun for the accusative (معلمين) and genitive or ~oon for the nominative (معلمون) is appended to the masculine noun. While ~aat (معلمات) is appended to the plural feminine noun and the letter "h" is attached to the end of the word to form the singular feminine noun (معلمة). The dual is formed by adding "ان" or "ين" at the end of the noun as in (معلمان). In broken plurals, the pattern of the singular noun is dramatically altered, thus, the suffixes and prefixes are not certain. The personal pronoun can appear as an isolated form or as suffixes attached to the nouns, verbs, or prepositions. Certain suffixes are attached at the end of words to signify possessive pronouns. The affix can be one letter, for example (بيتي) when the letter "ي" is attached to the end of the word (بيت) to form "my house" in English. For the plural, two letters are attached to the end of the word, for the masculine, the letters "هم" are attached (بيتهم), and the letters "هن" for the feminine nouns (بيتهن) . These are the most common modifications to the nouns and verbs. In Table 4, we illustrate additional suffixes.

In the Arabic language, nouns can be definite as in (المعلم) or in indefinite as in (معلم). Adding the prefix (ال) "*al*" makes the noun definite. The definite articles and prefixes that can be attached to the head of the definite article are considered the most common prefixes. In addition, the letter (و) is a commonly used letter to start the sentences within the Arabic language. This letter is equivalent to the English conjunction "*and*". In Table 5, we show an example of these prefixes. Our approach is mainly based on suffix and prefix removal and normalization.

Suffixes	Example	English explanations
ات	معلمات	teachers, feminine, plural
ان	باحثان	two Researchers, masculine, dual
ون	مخترعون	inventors, masculine, plural
كم	بحثكم	your research, masculine, plural
ين	مخترعين	inventors, masculine, genitive plural
هم	كتابهم	their book, masculine, plural

Table 4. Some frequent suffixes

Prefix	Example	English explanations
ال	الكتاب	*the* book
وال	والكتاب	and the book
كال	كالطير	*like the* bird
سي	سيقول	*he will* say
ست	ستقول	*she will* say

Table 5. Some frequent prefixes

The basis of the light stemmer consists of several rounds that attempt to locate and strip out the most frequent prefixes and suffixes. The light stemming algorithm mainly processes the affixes of inflectional morphology that are typically associated with the syntax, and have relatively little influence on the word senses. As a matter of fact, the inflectional affixes are the most frequent. The light-stemming algorithm adheres to the following steps:

Let T denote the set of characters of the Arabic surface word
Let L_i denote the position of letter i in term T
Let Stem denote the term after stemming in each step
Let D denote the set of definite articles
Let S denote the set of suffixes
Let P denote the set of prefixes
Let n is the total number of characters in the Arabic surface word

Step 1: Remove any diacritic in T,
Step 2: Normalize أ, إ, آ in L_1 of T to ا (plain alif)
Normalize ى in L_n of T to ي
Replace the sequence of ى in L_{n-1} and ء in L_n to ئ
Replace the sequence of ي in L_{n-1} and ء in L_n to ئ
Normalize ه in L_n of T to ة
Step3: If the length of T is greater than or equal to 3 characters then,
Remove the prefix Waw "و" in position L_1
Step 4: For all variations of D do,
Locate the definite article D_i in T
If D_i matches in T
$D_i = D_i$ + Characters in T ahead of D_i
$Stem = T - D_i$
Normalize أ, إ, آ in L_1 of S to "ا" (plain alif)
Step 5: If the length of *Stem* is greater than or equal to 3 characters then,
For all variations of S, obtain the most frequent suffix,
Match the region of S_i to *longest* suffix in *Stem*
If the length of ($Stem - S_i$) greater than or equal to 3 characters then,
$Stem = Stem - S_i$
Step 6: If the length of *Stem* is greater than 3 characters then,
For all variations of P do
Match the region of P_i in *Stem*
If the length of ($Stem - P_i$) greater than to 3 characters then,
$Stem = Stem - P_i$
Step 7: Return the *Stem*

All Arabic words are based on triliteral or quadriliteral roots. Thus, choosing 3 letters as the minimum root preserves the integrity of the word-sense. Reducing the stem to less than 3 letters results in the loss of at least one of the original letters. Within each step, if an affix is matched to a word, then the condition that the stem be greater than or equal to 3 characters attached to that action are tested on what would be the resulting stem, if that affix was removed. Once an affix is matched in a word and the remaining characters satisfy the condition then that affix is removed and control moves to the next step; if the rule is not accepted, then the next affix is tested until either a rule from that step fires and control passes to the next step or there are no more affixes that satisfy the rules in that step, hence control moves to the next step.

In step 1, the diacritics of the given Arabic word are removed. Diacritics (ً, ٌ, ُ, ٍ, ِ, َ, ْ) , which are marks above or below letters, are used in orthography. The algorithm removes all the diacritics except the diacritic "Shaddah" (ّ) since "Shaddah" is placed above a consonant letter as a sign for the duplication of that consonant; thus, it acts like a letter. In modern Arabic writing, people rely on their knowledge of the language and the context while writing the Arabic text. The Arabic surface form can be fully, partially, or entirely free of diacritics. The incompleteness of the surface orthography in most of the standard written Arabic in the WWW makes the written Arabic words ambiguous. Thus, removing diacritics is of great importance to normalizing the queries and the collection.

In step 2, the algorithm changes the letters (أ) "hamza-above-alif", (إ) "hamza-under-alif" and "alif-madah" (آ) to plain "alif" (ا). The reason behind this conversion is that most people do not formally write the appropriate "alif" at the beginning of the word. Thus, the letter "alif" is a source of ambiguity. For example, the verb (أخذ), which means "take" in English, and the plural noun (أحرف), which means "letters" in English, can be written as (اخذ) and (احرف). The normalization process preserves the word sense intact. Similarly for words that contain "hamza-under-alif" such as (إنسان), which means "human" in English, can be written as (انسان). Similarly, the letter "ta-marbotah" (ة) that occur at the end of the Arabic word which indicates mostly the feminine noun is, in most cases, written as "ha" (ه) which makes the word ambiguous. To resolve the ambiguity, we replace any occurrences of (ه) in the end of the word with (ة). For example, the word (حقيقة) alternately appears as (حقيقة) or (حقيقه) in Arabic text. We further replace the sequence of "alif-maksoura" (ى) in L_{n-1} and "hamza" (ء) in L_n to (ئ) "alif-maksoura-mahmozah". Similarly, we replace the sequence of "ya" (ي) in L_{n-1} and (ء) in L_n to (ئ).

In step 3, the connector "waw", "and" in English, is removed. Unlike the English language, the prefix "waw" in Arabic is attached to the beginning of a word. The prefix "waw" refers to the simultaneous "and". It is frequent in Arabic text, especially, at the beginning of a sentence or phrase. For example, (وقال), (وذكر), and (ومصدره) mean "and he says", "and he mentions", and "and its source" in English, respectively.

In step 4, the algorithm strips out the definite article of any Arabic word. The technique starts to match the most frequent and longest definite article in a list of definite articles to the given Arabic surface form. After locating the definite article in the given surface form, the algorithm strips it out and all letters ahead to the definite article. After removing the definite article, the algorithm checks the retained stem whether it starts with "alif" or not. If it begins with "alif", then the algorithm normalizes it as described in Step 2. Some definite articles are illustrated in Table 5.

In step 5, the technique attempts to locate and remove the suffixes. The most frequent suffixes are solely considered for removal. The longest suffix has greater priority for matching. If the algorithm fails to locate the suffix, then it considers shorter suffixes. When the region of the term to be stemmed matched the suffix, the algorithm removes that suffix. Before removing the stem, it checks the length of the target stem. If there are fewer than 3 characters, then it leaves the term intact; otherwise, it returns the stemmed term. Table 4 illustrates some simple examples of the most frequent suffixes in Arabic text.

The final step is to remove the remaining prefix. For instance, if the retained stem is greater than 3 characters, then the algorithm checks for the preposition (ل). If it is detected, then the prefix is removed, and the stem is checked again. If it is less than 3 letters, then the stem is left intact; otherwise, the stem is returned. In the second round, the algorithm diagnoses the stem to detect the preposition letter (ب). If the first letter is (ب) "baa" and the second letter is (ت) "taa", then the algorithm eliminates the preposition (ب). Another round is to check if the given stem starts with (ي); if it is detected then the algorithm removes the prefix (ي) "yaa" solely if the second letter is (ت). After removing the prefix, a normalization step is performed as in step 2.

Arabicized words are exceptions. The algorithm checks the stem against a list of Arabicized words; if it is found, then the Arabicized word is returned; otherwise, the technique proceed further in the stemming process. For example, the Arabic words (تكنولوجي), (انترنت), and (كمبيوتر) which mean "technology", "Internet", and "computer", respectively, in English are Arabicized.

Arabic words are based on trilateral, quadliteral, or pentliteral roots, as described in section 2.1. Thus, choosing 3 letters as the minimum root preserves the integrity of the word-sense. Reducing the stem to less than 3 letters results in the losing of at least one of the original letters.

6. EXPERIMENTAL APPROACH

We conducted our experiments using the benchmark data provided by the Text Retrieval Conference (TREC). TREC has three distinct parts: the documents, the topics, and the relevance judgments. We used the Arabic collection that consists of 383,872 documents of newswire stories published in the Agency France Press (AFP) between 1994 and 2000. We converted the encoding of the queries and the collection from UTF-8 to ISO8859-6 format. The TREC queries (or topics in the TREC vernacular) consist of three fields: title, description, and narrative. The title is considered short; it consists of one, two or three concept terms. The description field is of medium length; it consists of one or two sentences. We experimented with the 25 topics provided in TREC-10. In Table 6, we show an example of title and description fields. The average length of the titles of Arabic TREC topics is 6.2 words.

Title	Description
النقد و الشعر السياسي في العالم العربي	كيف يعبر النقاد العرب عن مواقفهم تجاه الشعر السياسي سواء كان مع او ضد النظام السياسي في بلدهم؟

Table 6. The title and the description fields of query topic 7

We modified the AIRE information retrieval system [5] to index the queries and collection separately for each stemming algorithm. The retrieval-based approaches are: surface, light stem, and root. The first approach indexes the collection without any stemming, i.e., leaving the Arabic surface word intact. The second approach indexes the collection using the roots as described in section 4. The third approach is to index the collection using light stemming as described in section 5. Obviously, a similar approach is used for the queries. Our parser eliminates any encountered stop-

words, punctuation marks, tags, and other noise terms. Similarity, querying is done after stemming and stop-word elimination. The stop-word list consists of about 750 words. This list includes distinct stop-words and its possible variations.

7. RESULTS

Using the TREC benchmark collections and queries described earlier, we evaluated our methods. We used three performance measures. The first uses the recall-precision scores at 11 standard points. In the Web, a user is certainly likely to be interested in only the top few retrieved Web pages. Thus, we provide measures for the top *n* documents retrieved. We also provide the overall average of precision of each run.

In general, it appears that all stemmers significantly perform better than no stemming at all. Not surprising, this comes from the fact that Arabic is a highly inflected language; thus, the stemming will group the huge variety of word forms into smaller conflation classes. It reflects our hypothesis as stated in Section 2. In Table 7, we provide comparisons measured in average precision of Light Stemming (LS), root, and surface. The baseline of comparisons is not stemming at all, which we call the surface form approach. T+D indicates the title and the description fields of TREC-10's topics, and T+D+RF refers to title and description enhanced by relevance feedback mechanism, which we refer it as RF. In relevance feedback, the top 15 terms from the top 10 documents, which are assumed relevant, are added to original Arabic query to produce new expanded query.

In terms of average precision-recall, as shown in Table 7, the LS algorithm achieved 0.3715 with no relevance feedback, and yielded an 87.7% improvement in the performance over the baseline. With query expansion using relevance feedback, it achieved 0.4312 measured in average precision and yielded a 71.3% improvement in the effectiveness over the baseline. The root detector algorithm achieved 0.3604, and 0.2987, measured in average precision with and without relevance feedback, respectively.

Average Precision	Surface (baseline)	Root-Search based on Khoja	LS
T+D	0.1979	0.2987 (50.9%)	0.3715 (87.7%)
T+D+ RF	0.2516	0.3604 (43.2%)	0.4312 (71.3%)

Table 7. Average Precisions of the 6 runs

The LS approach achieved superior performance over the root-search approach based on Khoja's stemmer; it improved the retrieval effectiveness by 19.6% and 24.3%, using relevance feedback, and without relevance feedback, respectively. The improvement over the root algorithm reflects our hypothesis as the root algorithm introduces extra noise in the conflation classes by conflating words with different meanings.

It is possible, in some cases, that the average measurements are not enough to describe and to confirm the overall performance. Standard statistical significance tests to evaluate information retrieval systems are described in [8]. In Table 8, we summarize the statistical significance test interpretation of our experiments. The test determines the probability that the obtained results could occur by chance. The evaluation is conducted using two statistical significance tests. We used parametric and non-parametric statistical significance tests. The parametric test is the paired t-test, and the non-parametric test is the Wilcoxon sign test [18].

The obtained *p*-values demonstrate that the observed performance differences of the Root over Surface form is significant at a 98% and 97% confidence interval using paired t-tests for T+D and T+D+RF, respectively. Similarity, the observed difference is significant at the 99% and 95% level using the Wilcoxon test. Statistically, the light-stemming algorithm (LS) significantly outperforms the root algorithm. The difference between LS and Root is significant at the 99% and 96% confidence interval using the paired t-test for T+D and T+D+RF, respectively. It is also statistically significant at the 99% and 99% confidence interval using Wilcoxon test for T+D and T+D+RF, respectively. The results suggest that the observed effect reflects an underlying difference in the effectiveness.

	Statistical significant Test	T+D	T+D+RF
Root vs. Word	Paired t-test	P=0.02	P=0.03
	Wilcoxon sign test	P=0.01	P=0.05
LS vs Root	Paired t-test	P=0.003	P=0.04
	Wilcoxon sign test	P=0.0001	P=0.01

Table 8. Statistical significance test

As it is most likely that the users in a medium like the Web do not read many retrieved documents, we demonstrate the effects on the precision-recall measure for the three approaches at fixed document cutoff points; we present them in terms of the 5, 10, 15, 20, and 30 top documents retrieved. Column one corresponds to the light stemming (LS). Column two shows the root-based retrieval approach. Column three shows the surface form approach. As illustrated in Tables 9, and 10, the surface form runs consistently performed the poorest while the LS algorithm was consistently the best. The reason behind the degradation of the performance by using the Arabic surface form is the large number of inflected variations of words in the Arabic language reducing the possibility of matching the query against the documents.

Precision	LS	Root	Surface
at 5 Docs	0.6720	0.5040	0.4480
at 10 Docs	0.6280	0.4960	0.4000
at 15 Docs	0.5627	0.4853	0.3813
at 20 Docs	0.5320	0.4660	0.3900
at 30 Docs	0.5067	0.4333	0.3587

Table 9. The top 30 documents retrieved for description run using light stemming, root, and word

Precision	LS	Root	Surface
at 5 Docs	0.7200	0.6400	0.4560
at 10 Docs	0.6800	0.5720	0.4840
at 15 Docs	0.6560	0.5387	0.4667
at 20 Docs	0.6360	0.5200	0.4560
at 30 Docs	0.5840	0.4840	0.4213

Table 10. The top 30 documents retrieved for description run using light stemming, root, and word with relevance feedback

A comparison of the retrieval performance of the three runs is shown in Figures 2 and 3. In Figures 2 and 3, we show the three curves of average precision at 11 recall points for the three run. As shown, the LS algorithm outperforms all the other methods. At the higher precision-lower recall levels (recall up to 0.4), the difference between the LS algorithm and the other approaches is even more noticeable. The higher precision region is of greater interest. Since users in a Web-like medium are unlikely to read many retrieved documents, the higher precision lower recall results obtained by the LS algorithm are even more significant. We note, however, that the improvements achieved by LS outperform the other approaches at all levels of recall (0.0-1.0).

The root detector algorithm performed much worse than the LS. An explanation for this result is that root-based retrieval is more aggressive than the light stemmer. This means that it conflates a lot more terms, which reduces performance greatly on many queries by introducing extraneous words in the same conflation class. For example, the root algorithm turned (مكتبه) "*library"* into the root (كتب), and (كتاب) "b*ook*" into the same root (كتب). Another invalid class is formed by conflating the word (معلومات) "*information*" and (معلم) "*teacher*" into same class. This yields to query drift and degradation of performance since the original Arabic query is expanded with ambiguous and unrelated terms.

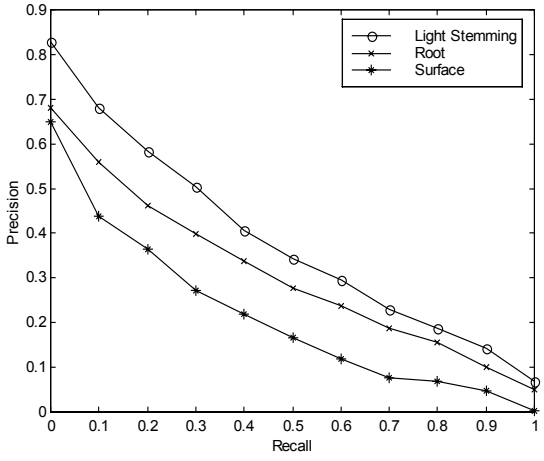

Figure 2. Average precision and recall on the description run using light stemming, root, and surface forms

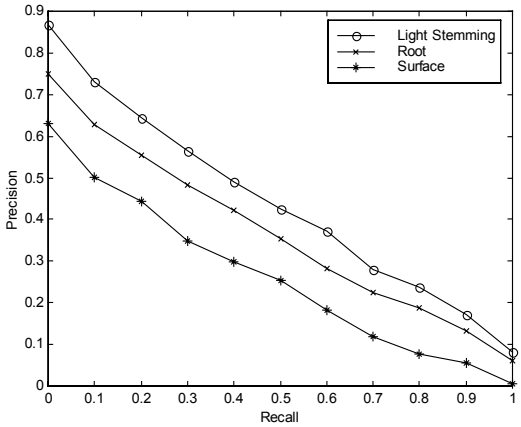

Figure 3. Average precision and recall on the description run using light stemming, root, and surface forms with relevance feedback

8. CONCLUSION

Our work demonstrates the potential of stemming in highly inflected languages such as Arabic. We demonstrated that stemming does result in significant improvements in retrieval effectiveness of Arabic information retrieval systems. We also evaluated the performance of using two different stemming algorithms. The root algorithm based on the work of Khoja, which is considered as an aggressive stemmer, has shown performance superiority over surface-based (no stemming) approach. The difference of the performance between the root algorithm and the Arabic surface word approach is statistically significant.

To resolve the ambiguity associated with the root algorithm, we designed and experimented with a novel stemming algorithm called light stemming (LS). LS is considered a non-aggressive stemmer. This approach is mainly based on suffix and prefix removal and normalization. The LS algorithm significantly outperforms the root algorithm. We found an 87.4% and 24.1% increase in average precision over the Arabic surface form and root algorithm, respectively. The root algorithm stems the surface form to a base form from which the word variants are derived. Many word variants with different semantic interpretation are based on an identical root. Therefore, the over-stemming of the root algorithm resulted in a deterioration of the retrieval performance as compared to the LS algorithm. The LS algorithm outperforms the latest n-gram based retrieval described in [14].

Our experimental findings also confirmed the well-known belief that automatic relevance feedback methods that improve retrieval performance in most languages also improve the retrieval performance of Arabic information retrieval systems.

With respect to stemming, our future work is to enhance the LS algorithm. We are going to develop more stemming rules based on canonical patterns. This approach will increase the number of candidates in each conflation class to include some of omitted related words. Moreover, it determines the effect of adding morphological variants in the query based on the meaning of the query word. In addition, we are going to add more Arabicized words to the exception word list.

9. REFERENCES

1. Abu-Salem, Hani, Al-Omari, Mahmoud, Evens, Martha. Stemming Methodologies over Individual Query Words for an Arabic Information Retrieval System. *JASIS 50(6):* 524-529, 1999.
2. Al-Fedaghi, S., and Al-Anzi, F. A New Algorithm to Generate Arabic Root-Pattern Forms. In *Proceedings of the 11th National Computer Conference and Exhibition*, March, Dhahran, Saudi Arabia, 1989.
3. Al-Shalabi, R, Evens, M. A Computational Morphology System for Arabic. *Workshop on Computational Approaches to Semitic Languages, COLING –ACL*, 1998
4. Beesley, K. Arabic Morphological Analysis on the Internet. In *the proceedings of the 6th International Conference and Exhibition on Multi-lingual Computing*, Cambridge, 17-18 April, 1998.
5. Chowdhury, et. al, "AIRE in TREC-9", *Proceedings of TREC-9*, NIST, 2001.
6. Egyptian Demographic Center, (2000). http://www.frcu.eun.eg/www/homepage/cdc/cdc.htm.
7. Hegazi, N., and Elsharkawi, A. Natural Arabic Language Processing. In *Proceedings of the 9th National computer Conference and Exhibition*, Saudi Arabia, 1986.
8. Hull, D. Using Statistical Testing in the Evaluation of Retrieval Performance. In *Proceedings of the 16th ACM/SIGIR Conference*, pages 329-338, 1993.
9. Hull, D. Stemming Algorithms - A Case Study for Detailed Evaluation. *JASIS*, 47(1):70-84, 1996.
10. Khoja, S. Stemming Arabic Text. Lancaster, U.K., Computing Department, Lancaster University. www.comp.lancs.uk/ computing/users/khoja/stemmer.ps, 1999.
11. Kraaij, W. Viewing stemming as recall enhancement. In *Proceedings of ACM SIGIR*, pp. 40-48, 1996.
12. Krovetz, R. Viewing morphology as an inference process. In *Proceedings of ACM-SIGIR*, pp. 191–202, 1993
13. Lovins, J. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11,22-31, 1968.
14. Mayfield, J., McNamee P., Costello C., Piatko C., Banerjee A. JHU/APL at TREC 2001: Experiments in Filtering and in Arabic, Video, and Web Retrieval. *Proceedings of TREC-10*, NIST, 2001
15. Oard, D. http://raven.umd.edu/dlrg/clir/arabic.html, 2002.
16. Porter, M. F. An algorithm for suffix stripping. *Program*, 14(3):130-137, 1980.
17. Tayli, M., and Al-Salamah, A. Building Bilingual Microcomputer Systems. *In Communications of the ACM*, Vol. 33, No.5, Pages 495-505, 1990.
18. Wonnacott, R., Wonnacott, T. Introductory Statistics, John Wiley & Sons, Fourth Edition,1990.
19. Xu, J. and Croft, W. . Corpus-Based Stemming using Co-occurrence of Word Variants. *ACM Transactions on Information Systems*. 16(1): 61-81, 1998