Improving Stemming for Arabic Information Retrieval: Light Stemming and Co-occurrence Analysis

Leah S. Larkey
Univ. of Massachusetts
Dept. of Computer Science
Amherst, MA 01003
larkey@cs.umass.edu

Lisa Ballesteros
Computer Science Dept.
Mt. Holyoke College
South Hadley, MA 01075
lballest@mtholyoke.edu

Margaret E. Connell
Univ. of Massachusetts
Dept. of Computer Science
Amherst, MA 01003
connell@cs.umass.edu

ABSTRACT
Arabic, a highly inflected language, requires good stemming for effective information retrieval, yet no standard approach to stemming has emerged. We developed several light stemmers based on heuristics and a statistical stemmer based on co-occurrence for Arabic retrieval. We compared the retrieval effectiveness of our stemmers and of a morphological analyzer on the TREC-2001 data. The best light stemmer was more effective for cross-language retrieval than a morphological stemmer which tried to find the root for each word. A repartitioning process consisting of vowel removal followed by clustering using co-occurrence analysis produced stem classes which were better than no stemming or very light stemming, but still inferior to good light stemming or morphological analysis.

Categories and Subject Descriptors
H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing – *Indexing methods, Linguistic processing;* H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *Clustering.*

General Terms
Experimentation, Performance, Algorithms.

Keywords
Cross-language information retrieval, cross-lingual, stemming, Arabic.

1. INTRODUCTION
Stemming is one of many tools used in information retrieval to combat the vocabulary mismatch problem, in which query words do not match document words. Stemmers equate or *conflate* certain variant forms of the same word like (*paper, papers*) and (*fold, folds, folded, folding…*). In English and many other western European languages, stemming is primarily a process of suffix removal [32, 40]. Such stemmers do not conflate irregular forms such as (*goose, geese*) and (*swim, swam).* In this work, we use the term stemming to refer to any process which conflates related forms or groups forms into equivalence classes, including but not restricted to suffix stripping. Stemming has been shown to improve performance in information retrieval tasks, usually by a small amount [25] and is considered to aid recall more than precision [29].

SIGIR'02, August 11-15, 2002, Tampere, Finland.

Stemmers are generally tailored for each specific language. Their design requires some linguistic expertise in the language and an understanding of the needs of information retrieval. Stemmers have been developed for a wide range of languages including Malay [42], Latin [23], Indonesian [7], Swedish [11], Dutch [29], German [35], French [36], Slovene [39], and Turkish [15]. The effectiveness of stemming across languages is varied and influenced by many factors. A reasonable summary is that stemming doesn't hurt retrieval; it either makes little difference or it improves performance by a small amount. Stemming appears to improve effectiveness more for highly inflected languages [38, 39] and when queries and/or documents are short [30].

Statistical methods can provide a more language-independent approach to conflation. Related words can be grouped based on various string-similarity measures. Such approaches often involve n-grams. Equivalence classes can be formed from words that share word-initial letter n-grams or a threshold proportion of n-grams throughout the word, or by refining these classes with clustering techniques. This kind of statistical stemming has been shown to be effective for many languages, including English, Turkish, and Malay [15, 17, 18, 37].

Stem classes can also be built or refined using co-occurrence analysis, which Xu and Croft proposed as a promising language-independent approach to stemming [45]. They demonstrated an improvement in retrieval effectiveness for English and Spanish after clustering conventional and n-gram based stem classes. Initial n-gram based stem classes are probably not the right starting point for languages like Arabic in which suffixing is not the only inflectional process. (See the overview of Arabic language issues below). However, co-occurrence or other clustering techniques can be applied to Arabic without using n-grams.

In the research reported here, we developed several light stemmers for Arabic which remove a small number of prefixes and suffixes and a co-occurrence based statistical stemmer which creates large stem classes by vowel removal and then refines these classes using co-occurrence. We evaluate these and several other approaches to Arabic stemming.

2. THE ARABIC LANGUAGE AND ORTHOGRAPHY
Arabic information retrieval has a particularly acute need for effective normalization and stemming. Both orthography and mor-

phology give rise to a huge amount of lexical variation. *Vocalized* text includes diacritics for short vowels and other details, conveying a nearly phonetic representation of a word, but it is only found in special contexts. The newspaper articles that make up the TREC-2001 corpus are not vocalized. Nonvocalized orthography is more ambiguous, and can cause a mismatch with texts, dictionaries, or queries that are vocalized. Regional variations in spelling add to the vocabulary mismatch problem.

Other variability arises from the derivational and inflectional productiveness of Arabic. A given word can be found in huge number of different forms which should possibly be conflated for information retrieval. Many definite articles, conjunctions, particles and other prefixes can attach to the beginning of a word, and large numbers of suffixes can attach to the end. At a deeper level, most noun, adjective, and verb stems are derived from a few thousand roots by infixing, for example, creating words like *maktab* (office), *kitaab* (book), *kutub* (books), *kataba* (he wrote), and *naktubu* (we write), from the root *ktb* [44].

Thus, some of the most closely related forms such as singular and plural nouns are irregular, and are not related by simple affixing (prefixing and suffixing). This situation is seen in English for a tiny fraction of nouns and a small number of very frequent verbs like the examples in the first paragraph of this paper, but is very common in Arabic.

For information retrieval, this abundance of forms means a greater likelihood of mismatch between the form of a word in a query and the forms found in documents relevant to the query. Distributional analyses of Arabic newspaper text show more words occurring only once and more distinct words than English text samples of comparable size.[1] The token to type ratio (mean number of occurrences over all distinct words in the sample) is smaller for Arabic texts than for comparably sized English texts [22]. Stemming should therefore be very important for Arabic information retrieval.

3. PREVIOUS RESEARCH

3.1 Stemming in Arabic

All the factors described in the previous section make Arabic very difficult to stem. First, there is the choice between roots or stems as the desired level of analysis for information retrieval. Considerable research on stemming and morphological analysis is amassing for the Arabic language, but no standard IR-oriented algorithm has yet emerged.

Four different approaches to Arabic stemming can be identified – manually constructed dictionaries, algorithmic light stemmers which remove prefixes and suffixes, morphological analyses which attempt to find roots, and statistical stemmers, which group word variants using clustering techniques.

Manually constructed dictionaries of words with stemming information are in surprisingly wide use. Al-Kharashi and Evens worked with small text collections, for which they manually built dictionaries of roots and stems for each word to be indexed [4]. Tim Buckwalter [9] developed a set of lexicons of Arabic stems, prefixes, and suffixes, with truth tables indicating legal combinations. The BBN group used this table-based stemmer in TREC-2001 [46].

Light stemming refers to a process of stripping off a small set of prefixes and/or suffixes, without trying to deal with infixes, or recognize patterns and find roots. Light stemming is mentioned by some authors without details [3, 14]. No explicit lists of strippable prefixes and/or suffixes or algorithm had been published at the time we did this research. Although light stemming can correctly conflate many variants of words into large stem classes, it can fail to conflate other forms that should go together. For example, broken (irregular) plurals for nouns and adjectives do not get conflated with their singular forms, and past tense verbs do not get conflated with their present tense forms, because they retain some affixes and internal differences.

Several morphological analyzers have been developed for Arabic [2, 5, 6, 12, 27] but few have received a standard IR evaluation. Such analyzers find the root, or any number of possible roots for each word. In addition, some attempt a more complete grammatical analysis of the word [6]. A morphological analyzer developed by Kareem Darwish was used by some of the TREC participants in 2001 [12, 33]. We obtained a simple morphological analyzer from Khoja and Garside [27] for this research.

Published comparisons of stems vs. roots for information retrieval have claimed that roots are superior to stems, based on small, nonstandard test sets [1, 4]. Recent work at TREC found no consistent differences between roots and stems [12].

3.2 Statistical Approaches to Stemming

Statistical techniques have widely been applied to automatic morphological analysis in the field of computational linguistics [8, 13, 16, 20, 21, 24, 26, 28]. For example, Goldsmith finds the best set of frequently occurring stems and suffixes using an information theoretic measure [20]. Oard et al. consider the most frequently occurring word-final n-grams (1, 2, 3, and 4-grams) to be suffixes [37]. Although such systems can be used on many different languages, they cannot be expected to perform well on languages like Arabic in which suffixing is not the only inflectional process.

Mayfield et al. have developed a system which combines word-based and 6-gram based retrieval, which performs remarkably well for many languages [34] including Arabic [33].

Al-Fares and De Roeck [14] used clustering on Arabic words to find classes sharing the same root. Their clustering was based on morphological similarity, using a string similarity metric tailored to Arabic morphology, which was applied after removing "a small number of obvious affixes." They evaluated the technique by comparing the derived clusters to "correct" classes. They did not assess the performance in an information retrieval context.

Stemmers make two kinds of errors. Weak stemmers fail to conflate related forms that should be grouped together. Strong stemmers tend to form larger stem classes in which unrelated forms are erroneously conflated. Most stemmers fall between these two extremes and make both kinds of errors. Xu and Croft [45] employ a corpus analysis approach which is particularly suited to splitting up stem classes created by strong stemmers. The stem-classes are reclustered based on a co-occurrence measure, which is language independent in that it can be applied to any set of stem classes.

[1] We use the term *word* in simple sense of text segmented at white space or punctuation, without any morphological analysis.

Xu and Croft applied their technique to effectively stem English and Spanish and showed two important points. First, one can refine an already-good stemmer by co-occurrence analysis and improve average precision. Second, one can start with a strong crude stemmer like an n-gram stemmer and use co-occurrence analysis to yield stem classes that work as well as a sophisticated stemmer.

In this work we apply the technique to Arabic, which has a more complex morphology than Spanish or English. Our goal was to determine whether a simple and effective Arabic stemmer could be quickly developed without a considerable amount of linguistic knowledge. Details of this technique are given in Section 4.5.

4. OUR APPROACHES TO ARABIC STEMMING

In the present research we compare several different approaches: no stemming, light stemming, morphological analysis to find roots, and statistical stemming using co-occurrence analysis, on the TREC-2001 Arabic data.

4.1 Normalization

Before stemming, corpus and queries were normalized as follows:

- Convert to Windows Arabic encoding (CP1256)
- Remove punctuation
- Remove diacritics (primarily weak vowels). Some dictionary entries contained weak vowels. Removal made everything consistent.
- Remove non letters
- Replace آ , إ , and أ with ا
- Replace final ى with ي
- Replace final ة with ه

4.2 Light Stemmers

Although several researchers allude to light stemming, we found no publication explicitly listing which affixes should be removed. Our guiding principle was to try to remove strings which would be found as affixes far more often than they would be found as the beginning or end of an Arabic word without affixes. We also benefited from discussions with some colleagues at TREC-2001, particularly M. Aljlayl. We tried several versions of light stemming, all of which followed the same steps:

1. Remove و ("and") for light2, light3, and light8 if the remainder of the word is 3 or more characters long. Although it is important to remove و, it is also problematic, because many common Arabic words begin with this character, hence the stricter length criterion here than for the definite articles.
2. Remove any of the definite articles if this leaves 2 or more characters.
3. Go through the list of suffixes once in the (right to left) order indicated in Table 1, removing any that are found at the end of the word, if this leaves 2 or more characters.

The strings to be removed are listed in Table 1. The "prefixes" are actually definite articles and a conjunction. The light stemmers do not remove any strings that would be considered Arabic prefixes.

Table 1: Strings removed by light stemming

	Remove from front	Remove Suffixes
Light1	ال، وال، بال، كال، فال	none
Light2	ال، وال، بال، كال، فال، و	none
Light3	"	ه، ة
Light8	"	ها، ان، ات، ون، ين، يه، ية، ه، ة، ي

4.3 Morphological Analysis

For morphological analysis we used software developed by Khoja and Garside [27], which first peels away layers of prefixes and suffixes, then checks a list of patterns and roots to determine whether the remainder could be a known root with a known pattern applied. If so, it returns the root. Otherwise, it returns the original word, unmodified. This system also removes terms that are found on a list of 168 Arabic stop words. Unlike the Buckwalter approach, this scheme has no table restricting the patterns and affixes applicable to particular stems and roots. Preliminary work with the Khoja stemmer revealed problems with proper nouns, so our implementation included a list of country and major city names translated into Arabic, considered "unbreakable," and exempted from further stemming. We tested the morphological analyzer both with and without the unbreakables.

4.4 Simple Stemmers

Many of the variant patterns derived from a single Arabic root differ internally only in vowels (see the examples in section 2). Removing vowels collapses the light stem classes into a smaller number of larger classes, grouping many forms that belong together and many forms that do not. Simple stemming consisted of light stemming and removal of vowels ا, و, ي, ء. (Short vowels were already removed during normalization). Our simple stemmers were intended to be strong, that is, to conflate too many forms, so that subsequent statistical analysis could find better classes by splitting.

We used three simple stemmers: *Simple* stems were derived by removing vowels from normalized words. *Simple2* applied light2 stemming, and then removal of vowels. *Simple8* applied light8 stemming, and then removal of vowels.

4.5 Co-occurrence Analysis

Co-occurrence analysis was used to refine the simple stemmers and the khoja stemmer, which were the strongest stemmers in the sense of creating the largest and overly-inclusive stem classes. We refer to the combined process of removing vowels and refinement with co-occurrence analysis as *repartitioning*.

Co-occurrence analysis is based on *em*, a variant of *EMIM* (expected mutual information) [43], which measures the proportion of word co-occurrences that are over and above what would be expected by chance. For two terms, *a* and *b*, *em* is defined as:

$$em(a,b) = \max\left(\frac{n_{ab} - En(a,b)}{n_a + n_b}, 0\right)$$

where n_{ab} is the number of times *a* and *b* co-occur in a text window of fixed size. n_a and n_b are the number of occurrences of *a* and *b* in the corpus. *En(a,b)*, the expected number of co-occur-

rences of a and b, is kn_an_b, where k is a constant based upon the corpus and window size. k is estimated from a sample of 5000 randomly chosen word pairs:

$$k = \sum n_{ab} / \sum n_a n_b$$

In our experiments, $k = 4.85 \times 10^{-6}$. The Arabic documents are short (150 words on average), so we used document length as our window size.

To repartition a stem class, the *em* metric is calculated for all the pairs of words in the class. In a first pass, a connected component algorithm is used to connect term pairs if their *em* score exceeds a threshold, *emthresh*. When the size of a resulting cluster is greater than twelve, a second-pass optimization is performed via *approximate optimal partitioning* [45]. In this second pass all pair-wise *em* scores are used to calculate an overall fitness measure, *cohesion*, for the class. A greedy algorithm refines the class by keeping terms that maximize cohesion and removing terms that lower it. This is done by partitioning the class into singleton sets, then repeatedly forming the union of the two sets for which cohesion is greatest. The algorithm stops when no two classes have a positive cohesion or when all terms belong to one class. Cohesion for a pair of terms, a and b, is calculated as $em(a,b) - \delta$, where δ is a limit on the amount by which conflating the terms could hurt precision. Cohesion for a set is the sum of cohesions for all pairs of words in the set.

In Xu and Croft's work, *emthresh* = 0.01 and $\delta = 0.0075$ were found to work well for the range of collections to which they applied the technique. We varied the values of these parameters for Arabic and did not find better values.

5. MONOLINGUAL ARABIC STEMMING EXPERIMENTS

5.1 Experimental Method

The TREC-2001 Arabic corpus, also called the AFP_ARB corpus, consists of 383,872 newspaper articles in Arabic from *Agence France Presse*. This fills up almost a gigabyte in UTF-8 encoding as distributed by the Linguistic Data Consortium. There are 25 topics with relevance judgments, available in Arabic, French, and English, with *Title*, *Description*, and *Narrative* fields. Although this test collection is small and has some problems, it is the only standard Arabic test set available [19]. We used the Arabic titles and descriptions as queries in monolingual experiments, and the English titles and descriptions in cross-language experiments.

Corpus and queries were converted to CP1256 encoding and indexed using an in-house version of INQUERY [10]. Arabic strings were treated as a simple string of bytes, regardless of how they would be rendered on the screen. Text was broken up into words at any white space or punctuation characters, including Arabic punctuation. Words of one-byte length (in CP1256 encoding) were not indexed. The experiments reported here used INQUERY for retrieval.

For the normalized conditions and some stemming conditions, we stemmed all tokens before indexing, and stemmed the queries with the same stemmer for retrieval. The co-occurrence experiments employed *query-based stemming*, where all retrieval uses a normalized, unstemmed database. The query was expanded, replacing each query term with a #syn (synonym) operator enclosing all members of the query term's stem class. We verified for several baseline stemming conditions that we get identical results whether we stem queries and corpus, or use query-based-stemming.

Arabic queries were expanded using the technique of local context analysis, adding 50 terms from the top 10 documents, as described in detail in [31]. Expansion was performed in order to show the ultimate level of performance attainable using the stemmers in the context of our whole system.

5.2 Results

5.2.1 Comparison of Basic Stemmers

Figure 1 shows precision at 11 recall points for the primary stemmers tested. *Raw* means no normalization or stemming, *khoja* means the Khoja stemmer, and *khoja-u* refers to the Khoja stemmer with the addition of the *unbreakables* list of items exempted from stemming.

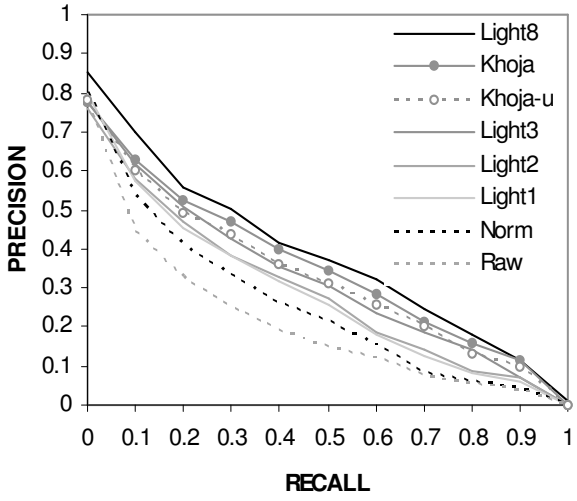

Figure 1: Monolingual 11 point precision for basic stemmers, unexpanded queries

Table 2: Monolingual average precision for basic stemmers, unexpanded

Stemmer	raw	norm	light1	light2	light3
Av. Precision	.194	.238	.273	.284	.317
Pct. Change		23.1	41.1	46.7	63.9

Stemmer	raw	khoja-u	khoja	light8
Av. Precision	.194	.313	.341	.376
Pct. Change		61.7	76.2	94.3

Table 2 shows uninterpolated average precision for the basic stemmers. For raw, normalized, and light stemming conditions performance is better with each successive increment in degree of

stemming. Each of these increments is statistically significant.[2] Surprisingly, on this data set, the Khoja stemmer performed better without the unbreakable list of countries and cities. However, this difference is not statistically significant. Although the light8 stemmer looks better than Khoja these differences are also not significant. Because the Khoja stemmer removes stop words and the other stemmers do not, we consider stop word removal next.

5.2.2 Removing Stop Words

Table 3 shows the effect of removing stop words. For all four degrees of stemming: raw, norm, light2, and light8, removing stop words results in a small increase in average precision, which is statistically significant for light2 and light8, but not for raw and normalized conditions.

Table 3: Monolingual average precision for stemmers with and without removing stop words

Stemmer	raw	norm	light2	light8
Stop words in	.194	.238	.284	.376
Stop words removed	.196	.241	.291	.389
Pct.Change	1.3	1.1	2.4	3.6

The fairer comparison between the khoja stemmers and light8 after removing stop words (light8-s) is summarized in Table 4. The difference between light8-s and khoja-u (with unbreakables) is statistically significant, for both unexpanded and expanded queries. The difference between light8-s and khoja (without unbreakables) is not significant for unexpanded or expanded queries. In short, the best stemmers for monolingual information retrieval were light8-s, a light stemmer, and khoja, a morphological analyzer.

Table 4: Monolingual average precision with and without query expansion

Stemmer	raw	norm-s	khoja-u	khoja	light8-s
Unexp.	.194	.241	.313	.341	.389
Expanded	.271	.330	.360	.378	.427

5.2.3 Simple Stemming and Co-occurrence Analysis

The four sections of Table 5 show retrieval performance and stem class sizes for each of the four baseline stemmers (in boldface) that were subjected to repartitioning (*norm-s, light2-s, light8-s,* and *khoja*). For norm-s, light2-s, and light8-s, the corresponding simple stemmer performance is shown. The khoja stemmer already had very large stem classes, so simple stemming before clustering was unnecessary. Finally, for all four baselines, the clustered stemmer performance is shown. Relative to simple stemming, clustering by co-occurrence significantly improves retrieval effectiveness in all cases.

[2] All significance tests were conducted using the Wilcoxon test [41] with a criterion of p<.05 for significance.

Table 5: Average Precision for monolingual retrieval - baseline, simple, and clustered stemmers

	Average Precision	Query Words Class size		Non-singleton Class Size
		Average	Max	Average
norm-s	**.241**	**1**	**1**	-
simple	.245	34	237	34
simple-c	.268	2.0	11	2.5
light2-s	**.291**	**3.6**	**4**	**3.7**
simple2	.267	58	317	58
simple2-c	.316	3.4	15	4.5
light8-s	**.389**	**22**	**45**	**22**
simple8	.308	141	605	141
simple8-c	.390	5.4	26	7.1
khoja	**.341**	**262**	**929**	**265**
khoja-c	.347	5.6	30	7.0

Relative to the baseline stemmers norm-s and light2-s, repartitioning yields a net improvement. Simple-c (clustered simple stemmer) is significantly better than norm-s, and simple2-c (clustered simple2) is significantly better than light2-s. In other words, starting with no stemming or very light stemming (removing of definite articles and و from the beginnings of the words), then creating overly inclusive stem classes by removing vowels, and repartitioning the classes by co-occurrence analysis, results in a net improvement in stemming without a great deal of linguistic knowledge.

On the other hand, co-occurrence analysis did not improve the more sophisticated stemmers: *khoja-c* is not better than *khoja,* and *simple8-c* is not better than *light8-s*.

It may seem strange that co-occurrence analysis on the stronger baseline stemmers changed class sizes drastically but did not change performance very much. We took a closer look at light8-s stem classes to gain some understanding of this phenomenon. We found that although light8-s is our strongest light stemmer, it is still relatively weak in that it fails to conflate some forms that should be conflated. On the positive side, it rarely groups unrelated forms, and when it does, it groups variants of relatively few words e.g. 2 or 3, except for stop words where we see 5 or 6. The clusters for light8-s are fairly large (avg class size=22), and correctly so. Removing vowels (simple stemming) conflates many unrelated forms (avg class size=141); the co-occurrence clustering generally separates the unrelated forms back out. However, these repartitioned stem classes are much smaller than would be desired (mean class size=5) and contained only the highest frequency forms. This behavior was appropriate for the English and Spanish, for which this algorithm was developed, but not for Arabic. An algorithm that was not so biased against low frequency forms might have yielded a net improvement.

The example of stem classes for the word العربي (*Arab* n. or adj.) in Table 6 is probably typical, given these numbers. The light8-s stem class contains 37 words, of which 26 are variants of the *Arab*. Nine words are variants of *chariot*. Two other words mean *earnest money.* Removing vowels (*simple8*) results in a

stem class with 117 members (not shown) for the target word, which co-occurrence analysis (*simple8-c*) reduces to the 8 words shown. All of the variants of *chariot* and *earnest money* have gone away, but so have many good variants of *Arab.* This small new set includes one correct variant which was not in the original set of 36.

Ideally, the final stem class should include more of the original variants from light8-s and add more new variants if they appeared in simple8 and still keep the *chariot* and *earnest money* variants out. We are continuing to experiment with parameters and co-occurrence measures to see whether we can produce the larger classes appropriate to Arabic without bringing in too many unrelated forms.

Table 6: Example of stem classes under different stemmers

Stem class for العربي under light8-s			
Variants of *Arab*		of *Chariot*	of *earnest money*
العربي	وعربيين	عربات	عربون
العربي	وعربيه	عرباتها	العربون
العربيات	العرب	والعربات	
العربيان	عربي	وعربات	
العربين	عربيات	العربات	
العربيه	عربيان	عربه	
العربيين	عربين	والعربه	
العربييه	عربيه	وعربه	
عرب	وعربي	العربه	
عربيين	والعربي		
عربييه	والعربيه		
والعرب	وعربيان		
وعرب	وعربيه		
Stem class under simple8-c			
All *Arab*			
العربي	عرب	عربيا	والعربي
العربيه	عربي	عربيه	العرب

5.3 Discussion

Although stemming is difficult in a language with complex morphology like Arabic, it is particularly important. For monolingual retrieval, we saw around 100% increase in average precision from raw retrieval to the best stemmer. The best stemmer in our experiments, light8-s was very simple and did not try to find roots or take into account most of Arabic morphology. It is probably not essential for the stemmer to yield the correct forms, whether stems or roots. It is sufficient for it to group most of the forms that belong together.

It was interesting that removing stop words had a significantly positive effect for stemmed Arabic, but not for unstemmed Arabic. This difference is probably due to the fact noted in section 5.2.3 that stem classes for stop words contain larger numbers of unrelated word variants than stem classes for other words.

6. CROSS-LANGUAGE ENGLISH-ARABIC EXPERIMENTS

For generality, the stemmers are compared on the cross-language retrieval task. The cross-language experiments reported here were carried out using the 25 English TREC-2001 queries and the same Arabic AFP_ARB corpus used for the monolingual experiments. Our approach is the common dictionary-based approach, in which each English query word is looked up in a bilingual dictionary. All the Arabic translations for that word are gathered inside an INQUERY #syn (synonym) operator. For an Arabic-English dictionary, we used a lexicon collected from several online English-Arabic and Arabic-English resources on the web, described more completely in [31]. Query expansion was carried out in conjunction with stemming. When English queries were expanded, 5 terms were added from the top 10 documents. When Arabic queries were expanded, 50 terms were added from the top 10 documents, as described [31].

Figure 2 shows precision on unexpanded queries for cross-language retrieval at 11 recall points for raw, norm-s (normalization and stop word removal), light8-s (light8 stemming with stop word removal), khoja-u (with unbreakables), and khoja stemmers. Table 7 shows uninterpolated average precision for unexpanded and expanded queries.

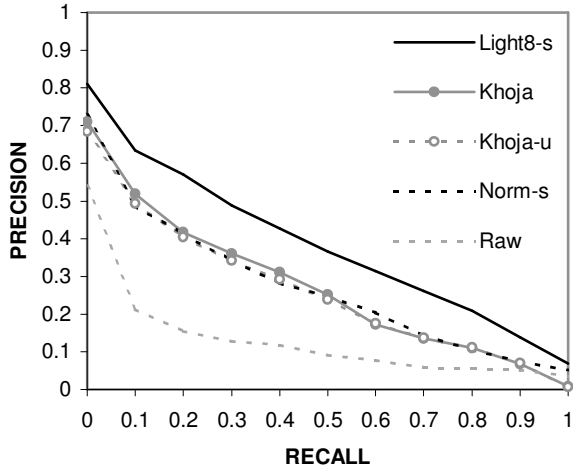

Figure 2: Cross-Language 11 point precision for unexpanded queries.

Table 7: Cross-language average precision different stemmers, unexpanded and expanded queries

Stemmer	**raw**	**norm-s**	**khoja-u**	**khoja**	**light8-s**
Av.Precision	.113	.262	.252	.260	.379
Pct. Change		133	123	130	236
With English Query expansion					
Av.Precision	.139	.306	.293	.308	.422
Pct. Change		120	111	121	204
With English and Arabic Query expansion					
Av.Precision	.163	.336	.316	.321	.436
Pct. Change		106	93	97	167

The results are somewhat different from the monolingual results. Raw retrieval without any normalization or stemming is far worse for cross-language retrieval than for monolingual retrieval. This is probably because many of the Arabic words occurred in vocalized form (with diacritics) in the online dictionary we used for cross-language retrieval. Without normalization these dictionary entries do not match their counterparts in the corpus. Other differences from the monolingual case are that here, the *light8-s* stemmer is significantly better than the root stemmer, *khoja*, which is no better than normalization for cross-language retrieval.

7. CONCLUSIONS

Stemming has a large effect on Arabic information retrieval, at least in part due to the highly inflected nature of the language. For monolingual retrieval we have demonstrated improvements of around 100% in average precision due to stemming and related processes, and an even larger effect for dictionary-based cross-language retrieval. This stemming effect is very large, compared to that found in many other stemming studies, but is consistent with the hypothesis of Popovič and Willett [39] and Pirkola [38] that stemming should be particularly effective for languages with more complex morphology.

It may seem contradictory that while we find a very large stemming effect for both mono- and cross-language Arabic retrieval, Xu et al. found stemming to make a difference only for monolingual Arabic, on the same TREC-2001 data [47]. We believe that the reason is that Xu et al. had a parallel corpus, so their bilingual lexicon contained all the variants of the Arabic words that were likely to occur in documents. Our bilingual lexicon was derived from an online dictionary, so it contained far fewer variants. Without stemming, the dictionary translations of query terms were unlikely to match the forms found in documents. In short, with sufficient parallel data, stemming may be unnecessary.

The best stemmer was a light stemmer that removed stop words, definite articles, and و ("and") from the beginning of words, and a small number of suffixes from the ends of words (*light8-s*). With query expansion, *light8-s* yielded results comparable to that of the top performers at TREC, monolingual and cross-language. We have not ruled out the possibility that a better morphological analyzer could work as well as or better than the light stemmer.

A repartitioning process consisting of vowel removal followed by clustering using co-occurrence analysis performed better than no stemming or very light stemming. However, stemmers produced this way were still inferior to the best light and morphological stemmers. Repartitioning one of these good hand-designed stemmers changes stem classes a great deal, but does not improve (or hurt) overall retrieval performance. We suspect that performance might be improved by modifying the clustering method to have less bias against low frequency variants.

8. ACKNOWLEDGMENTS

We would like to thank Shereen Khoja for providing her stemmer, Nicholas J. DuFresne for writing some of the stemming and dictionary code, Fang-fang Feng for helping with dictionary collection over the web, Mohamed Taha Mohamed, Mohamed Elgadi, and Nasreen Abdul-Jaleel for help with the Arabic language, Victor Lavrenko for the use of his vector and language modeling code. This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsor.

9. REFERENCES

[1] Abu-Salem, H., Al-Omari, M., and Evens, M. Stemming methodologies over individual query words for Arabic information retrieval. *JASIS*, 50 (6), pp. 524-529, 1999.

[2] Al-Fedaghi, S. S. and Al-Anzi, F. S. A new algorithm to generate Arabic root-pattern forms. In *Proceedings of the 11th national computer conference.* King Fahd University of Petroleum & Minerals, Dhahran, Saudi Arabia, pp. 391-400, 1989.

[3] Aljlayl, M., Beitzel, S., Jensen, E., Chowdhury, A., Holmes, D., Lee, M., Grossman, D., and Frieder, O. IIT at TREC-10. In *TREC 2001*. Gaithersburg: NIST, 2001.

[4] Al-Kharashi, I. and Evens, M. W. Comparing words, stems, and roots as index terms in an Arabic information retrieval system. *JASIS*, 45 (8), pp. 548-560, 1994.

[5] Al-Shalabi, R. *Design and implementation of an Arabic morphological system to support natural language processing*. PhD thesis, Computer Science, Illinois Institute of Technology, Chicago, 1996.

[6] Beesley, K. R. Arabic finite-state morphological analysis and generation. In *COLING-96: Proceedings of the 16th international conference on computational linguistics*, vol. 1, pp. 89-94, 1996.

[7] Berlian, V., Vega, S. N., and Bressan, S. Indexing the Indonesian web: Language identification and miscellaneous issues. Presented at Tenth International World Wide Web Conference, Hong Kong, 2001.

[8] Brent, M. R. Speech segmentation and word discovery: A computational perspective. *Trends in Cognitive Science*, 3 (8), pp. 294-301, 1999.

[9] Buckwalter, T. *Qamus: Arabic lexicography*. http://members.aol.com/ArabicLexicons/

[10] Callan, J. P., Croft, W. B., and Broglio, J. TREC and TIPSTER experiments with INQUERY. *Information Processing and Management*, 31 (3), pp. 327-343, 1995.

[11] Carlberger, J., Dalianis, H., Hassel, M., and Knutsson, O. Improving precision in information retrieval for Swedish using stemming. In *Proceedings of NODALIDA '01 - 13th Nordic conference on computational linguistics*. Uppsala, Sweden, 2001.

[12] Darwish, K., Doermann, D., Jones, R., Oard, D., and Rautiainen, M. TREC-10 experiments at Maryland: CLIR and video. In *TREC 2001*. Gaithersburg: NIST, 2001.

[13] de Marcken, C. *Unsupervised language acquisition*. PhD thesis, MIT, Cambridge, 1995.

[14] De Roeck, A. N. and Al-Fares, W. A morphologically sensitive clustering algorithm for identifying Arabic roots. In *Proceedings ACL-2000*. Hong Kong, 2000.

[15] Ekmekcioglu, F. C., Lynch, M. F., and Willett, P. Stemming and n-gram matching for term conflation in Turkish texts. *Information Research News*, 7 (1), pp. 2-6, 1996.

[16] Flenner, G. Ein quantitatives Morphsegmentierungssytem fur Spanische Wortformen. In *Computatio linguae II*, U. Klenk, Ed. Stuttgart: Steiner Verlag, pp. 31-62, 1994.

[17] Frakes, W. B. Stemming algorithms. In *Information retrieval: Data structures and algorithms*, W. B. F. a. R. Baeza-Yates, Ed. Englewood Cliffs, NJ: Prentice Hall, chapter 8, 1992.

[18] Freund, E. and Willett, P. Online identification of word variants and arbitrary truncation searching using a string similarity measure. *Information Technology: Research and Development*, 1, pp. 177-187, 1982.

[19] Gey, F. C. and Oard, D. W. The TREC-2001 cross-language information retrieval track: Searching Arabic using English, French, or Arabic queries. In *TREC 2001*. Gaithersburg: NIST, 2002.

[20] Goldsmith, J. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27 (2), pp. 153-198, 2000.

[21] Goldsmith, J., Higgins, D., and Soglasnova, S. Automatic language-specific stemming in information retrieval. In *Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop*, C. Peters, Ed.: Springer Verlag, pp. 273-283, 2001.

[22] Goweder, A. and De Roeck, A. Assessment of a significant Arabic corpus. Presented at the Arabic NLP Workshop at ACL/EACL 2001, Toulouse, France, 2001.

[23] Greengrass, M., Robertson, A. M., Robyn, S., and Willett, P. Processing morphological variants in searches of Latin text. *Information research news*, 6 (4), pp. 2-5, 1996.

[24] Hafer, M. A. and Weiss, S. F. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10, pp. 371-385, 1974.

[25] Hull, D. A. Stemming algorithms - a case study for detailed evaluation. *JASIS*, 47 (1), pp. 70-84, 1996.

[26] Janssen, A. Segmentierung Franzosischer Wortformen in Morphe ohne Verwendung eines Lexikons. In *Computatio linguae*, U. Klenk, Ed. Stuttgart: Steiner Verlag, pp. 74-95, 1992.

[27] Khoja, S. and Garside, R. *Stemming Arabic text*. Computing Department, Lancaster University, Lancaster, 1999. http://www.comp.lancs.ac.uk/computing/users/khoja/stemmer.ps

[28] Klenk, U. Verfahren morphologischer Segmentierung und die Wortstruktur im Spanischen. In *Computatio linguae*, U. Klenk, Ed. Stuttgart: Steiner Verlag, 1992.

[29] Kraaij, W. and Pohlmann, R. Viewing stemming as recall enhancement. In *Proceedings of ACM SIGIR96*. pp. 40-48, 1996.

[30] Krovetz, R. Viewing morphology as an inference process. In *Proceedings of ACM SIGIR93*, pp. 191-203, 1993.

[31] Larkey, L. S. and Connell, M. E. Arabic information retrieval at UMass in TREC-10. In *TREC 2001*. Gaithersburg: NIST, 2001.

[32] Lovins, J. B. Development of a stemming algorithm. *Mechanical Translation and Computational Linguistics*, 11, pp. 22-31, 1968.

[33] Mayfield, J., McNamee, P., Costello, C., Piatko, C., and Banerjee, A. JHU/APL at TREC 2001: Experiments in filtering and in Arabic, video, and web retrieval. In *TREC 2001*. Gaithersburg: NIST, 2001.

[34] McNamee, P., Mayfield, J., and Piatko, C. A language-independent approach to European text retrieval. In *Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop*, C. Peters, Ed.: Springer Verlag, pp. 129-139, 2000.

[35] Monz, C. and de Rijke, M. Shallow morphological analysis in monolingual information retrieval for German and Italian. In *Cross-language information retrieval and evaluation: Proceedings of the CLEF 2001 workshop*, C. Peters, Ed.: Springer Verlag, 2001.

[36] Moulinier, I., McCulloh, A., and Lund, E. West group at CLEF 2000: Non-English monolingual retrieval. In *Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop*, C. Peters, Ed.: Springer Verlag, pp. 176-187, 2001.

[37] Oard, D. W., Levow, G.-A., and Cabezas, C. I. CLEF experiments at Maryland: Statistical stemming and backoff translation. In *Cross-language information retrieval and evaluation: Proceedings of the CLEF 2000 workshop*, C. Peters, Ed.: Springer Verlag, pp. 176-187, 2001.

[38] Pirkola, A. Morphological typology of languages for IR. *Journal of Documentation*, 57 (3), pp. 330-348, 2001.

[39] Popovic, M. and Willett, P. The effectiveness of stemming for natural-language access to Slovene textual data. *JASIS*, 43 (5), pp. 384-390, 1992.

[40] Porter, M. F. An algorithm for suffix stripping. *Program*, 14 (3), pp. 130-137, 1980.

[41] Siegel, S. *Nonparametric statistics for the behavioral sciences*. New York: McGraw-Hill, 1956.

[42] Tai, S. Y., Ong, C. S., and Abdullah, N. A. On designing an automated Malaysian stemmer for the Malay language. (poster). In *Proceedings of the fifth international workshop on information retrieval with Asian languages,* Hong Kong, pp. 207-208, 2000.

[43] van Rijsbergen, C. J. *Information retrieval*. London: Butterworths, 1979.

[44] Wightwick, J. and Gaafar, M. *Arabic verbs and essentials of grammar*. Chicago: Passport Books, 1998.

[45] Xu, J. and Croft, W. B. Corpus-based stemming using cooccurrence of word variants. *ACM Transactions on Information Systems*, 16 (1), pp. 61-81, 1998.

[46] Xu, J., Fraser, A., and Weischedel, R. TREC 2001 cross-lingual retrieval at BBN. In *TREC 2001*. Gaithersburg: NIST, 2001.

[47] Xu, J., Fraser, A., and Weischedel, R. Empirical studies in strategies for Arabic retrieval. In *Sigir 2002*. Tampere, Finland: ACM, 2002.