T. Rachidi, A.. Chekayri, M. Mhamdi, O. Chhoul, and A.. Fala.
Al Akhawayn University in Ifrane, Ifrane, Morocco

The Effect of Full and Partial Diacritization on Arabic Root Extraction

Abstract

This paper presents a novel approach for extracting roots of vocalized Arabic words. The developed Vocalized Arabic Word Root Extraction (VAWRE) algorithm is a continuation of previous research conducted at the Arabic Computing research laboratory at Al Akhawayn University for the development of an Arabic root extractor [1], which has been integrated onto Barq search engine [2]. The approach takes into account both the non-concatenative morphology and the complex orthography of the Arabic language. The VAWRE algorithm uses a manually constructed dictionary of 8,950 Arabic roots and a maintained list of vocalized morphological templates organized into 45 sets [3]. The constructed root dictionary along with the list of vocalized morphological template sets covers all most frequent words that appear in Arabic modern text. The algorithm extracts the most precise root (or the set of all possible roots in case of ambiguity) rather than stems. The approach makes use of diacritic marks, which are used in the Arabic language mainly as short vowels, for the purpose of reducing the identified root ambiguities and hence, enhancing the root extraction precision. Moreover, it provides enough flexibility to handle fully vocalized, partially vocalized and non-vocalized words, so as to cope with the recognizable lack of a standardized punctuation model in modern Arabic texts.

The implemented approach has been tested on evaluation corpora, which consist of 258 Arabic text documents collected from the Web. The obtained results have shown that the VAWRE algorithm achieved an overall performance of 85% and an average root extraction correctness of 77%. Moreover, the results have proven that the use of vocalization in root extraction achieves an average root ambiguity reduction of 33%.

1. Introduction

Root extraction is defined as the process which conflates related morphological derivational forms into the original root [4]. The root refers to the common denominator, which is composed of consonants referred to as radicals, shared by a number of words connected synchronically by meaning [5]. Researches in many different areas such as document management, machine translation, summarization, and automatic categorization have shown great interest to root extraction in the Arabic language. Still, root extraction is habitually linked to Information Retrieval (IR) systems and precisely to the indexing process. This is for the reason that several studies such as the one described in [6] suggested that indexing Arabic text using roots significantly increases retrieval effectiveness over the use of words or stems. Further, it has been concluded based on evaluation experiments with several suffix-stripping algorithms that suffix removal does not improve retrieval effectiveness [7]. As a result, leading Arabic search engines [8, 9] and many multilingual search engines [10-13,14]] make use of Arabic root extraction algorithms in order to overcome the inefficiencies in the precision and recall. Google search engine [12] is an exception since it does not use any root extraction algorithm for the Arabic language and hence, a decrease in the number of returned Web pages is noticed

when using morphological derivatives as key words for querying instead of basic roots [1]. The use of efficient root extraction algorithms is especially important when dealing with inflected languages that are characterized by huge amount of lexical variation. The Arabic language is a Semitic language that presents many challenges to automatic processing due to both its non-concatenative morphology and orthographic variations [15].

The highly inflectional morphology is the reason that makes stemming not sufficient to derive the basic root of Arabic words. Unlike in Germanic and Indo-European languages where the words are made of concatenating stems and affixes, Arabic words are generated from a closed set of roots by applying morphological templates and then, appending suffixes and prefixes. According to the Arabic dictionary 'Lisan Al-Arab' authored by Ibn Manzur, the number of these roots is around 10,000, but only about 5,000 roots are in common usage [16]. They are made of three or four consonants and rarely five consonants. Each root can be legally combined with only a limited number of morphological templates, which are organized into sets with an average of seventeen morphological templates per set [17].

Although, Arabic infixation rules are deterministic, it is unworkable to construct an infixation-based root extraction algorithm because these rules depend on the number of literals in the root (and obviously on other rules), which is evidently not known to the root extraction process. Furthermore, the application of affix (prefix and suffix) removal rules to Arabic words can lead to extracting wrong roots, and to generating different results depending on the order of their application. In addition, only 70% of Arabic words are considered to be regular since they are generated using derivational rules that can be systematically automated for the purpose of extracting roots. In that case, the root extraction can be based on pre-defined infix rules [3]. However, the remaining 30% of irregular words are derived from weak roots, which contain glides ([*w* : و] and/or [*y* : ي]) in the beginning, middle or end of roots [18]. For the case of these irregular words, no systematic way can be automated to extract roots without using some manually-constructed lookup lists and the available pre-defined infix rules are not sufficient.

Besides the morphological challenge, Arabic language presents another major challenge caused by the non-standardized usage of diacritic marks which result in three vocalization forms of Arabic texts: fully vocalized partially vocalized and non-vocalized texts. This vocalization diversity is caused by the fact that diacritic marks are not considered as essential elements of the orthography of Arabic words and hence, written modern texts are usually made of scripts that exclude most or all diacritic marks. These different vocalization forms usually coexist in modern Arabic documents. Vocalization is mainly used to eliminate semantic ambiguities between Arabic words sharing the same combination of letters and have different phonetics and hence, semantics. There are five possible morphologically correct vocalizations per word on average in the Arabic language [17]. Another challenge related to the Arabic orthography is the changing forms of both the Alif (ا) and the Hamza (ء). The changing form of these two letters causes a

problem since a mechanism is needed to recognize that an Alif that appear in the root as أ takes the form of ؤ in a morphological derivative (e.g. س أ ل : سؤال).

We are proposing an approach to Arabic word root extraction using root+pattern+features representation. Our approach handles the highly inflected nature of Arabic and makes use of any diacritic marks (even partial) attached to the word for the purpose of improving the root extraction precision by reducing root ambiguities. Our approach, like the one of MAGEAD [24-25], uses a templatic morphology, and does on-line analysis, rather than using precompiled stems and only analyzing affixational morphology as is done in [24].

For the purpose of evaluating the developed VAWRE algorithm, corpora made of Arabic text collected from the Web have been used.

2. Previous Works

Conventional stemming algorithms are usually rule-based and only suit the concatenative nature of Germanic and Indo-European languages [19]. They can be broadly classified into four different categories: table look up, N-gram stemming, successor variety and affix removal [20]. Various adjustments of the most successful ones among them (e.g. Porter stemmer [21] and Lovins stemmer [22]) have been proposed for extracting roots of Arabic words. However, these proposed adaptations seriously suffer from at least one of the following problems:

1. Under or over stemming,
2. Extract incorrect roots due to the irregularity of weak verbs and
3. Require huge implementation complexity and/or space overhead.

To combat against these problems, other approaches have been proposed that better suit the complex morphological nature of the Arabic language. The main types of Arabic stemmers can be categorized into the following four classes: manually constructed dictionaries, algorithmic light stemmers, automatic morphological analyzers, and statistical stemmers which group word derivatives using clustering techniques [20]. Manually constructed dictionaries of words with stemming information are in wide use in spite of the facts that the manual construction is a time consuming process and requires a considerable language expertise. Light stemming refers to a process of removing a set of prefixes and suffixes. Light stemmers usually ignore any existing infixes and do not try to recognize the morphological structure of the word. And hence, they produce mainly stems, rather than roots. Morphological analyzers attempt to find a root or any number of possible roots for each word. They are usually equipped with complete language grammatical analyzers. MAGEAD [24-25] and Buckwalter's [23] morphological analyzer are two such analyzers. Finally, statistical stemmers attempt to group word morphological derivatives using clustering/statistical techniques. Most of the proposed Arabic root extraction algorithms make two kinds of errors: weak stemmers fail to conflate related morphological derivatives that should be grouped together and rooted back to the same root, and strong

stemmers tend to form larger stem classes in which unrelated morphological derivatives are erroneously conflated [19]. Most stemmers fall between these two extremes and make both kinds of errors.

3. Proposed Root Extraction Approach

The proposed VAWRE algorithm is based on a manually constructed dictionary of 8,950 Arabic roots and a list of vocalized morphological templates organized into 45 sets. The root dictionary is made of 85% of tri-radical roots (e.g. غ ب ن, س د د and ح م ل) and 15% of quadri-radical roots (e.g. ق ن ط ر, ل م ل م and ب ل و ر) [3]. Every template set is made of fully-vocalized morphological templates made of the letters of the basic morpheme (ف, ع and ل) and a vowel melody. Each morphological template set covers the root transformations with respect to the tenses (perfect and imperfect), the voices (active and passive), person, number, gender and verbal nouns (*masādir*), which can be applied to a group of root verbs sharing some common characteristics. These characteristics classify all maintained roots into classes depending on the applied template set. In addition, two lists of the most common prefixes and suffixes in the Arabic language were manually constructed. The central part of our approach is to build a memory-resident structure (refer to as I-Map) that will hold the automatically generated fully-vocalized morphological derivatives for each Arabic root based on the selected morphological template set. The I-Map will have as many entries as the number of roots and all the morphological derivatives generated for each root will be stored in a separate structure and linked to the root's entry. After the I-Map construction, the process of extracting the root of a word starts by identifying the best candidate root entries with high probability of having that word among the list of automatically generated morphological derivatives. This selective processing of root entries avoids the time consuming, and unpractical, process of going through all I-Map root entries for each word to be processed. Then, Regular Expressions (RE) are used to perform a pattern matching between the processed word and all morphological derivatives linked to each candidate root. REs are used to detect partial matches in order to take into consideration three important facts:

1. The automatically generated morphological derivatives are fully-vocalized, whereas the processed word might be partially or non-vocalized,
2. Both Alif (ا) and Hamza (ء) might have different forms in both the automatically generated morphological derivative and the processed word and
3. The processed word might be appended to prefixes and/or suffixes, whereas the automatically generated morphological derivatives have none.

In case a match occurs between the processed word and one or more morphological derivatives of a given I-Map entry, the algorithm approves that the root corresponding to that entry is a correct root of the processed word after verifying that any detected elements surrounding the match are valid suffixes and/or prefixes.

3.1. Implementation Description

The initialization phase of the VAWRE algorithm consists of constructing the I-Map structure and hence, generating all the morphological derivatives corresponding to each root. Then, as a filtering step, the processed words are looked up in a maintained list of stop word and special word in order to avoid starting the root extraction process when it is not required. The words identified to be stop or special words are returned as they are and the algorithm produces the result "Stop Word" or "Special Word" respectively. In case the root extraction process is initiated, four possible outputs are produced: "Unique Root Found" when one unique root is returned, "Multiple Roots Found" when more than one root are returned, "No Template Found" in case at least one candidate root entry has been identified but no complete match has been obtained and "No Root Found" in case no candidate root entries have been identified.

3.1.1. I-Map Construction

VAWRE algorithm makes use of two distinct files containing the maintained dictionary of roots and a list of morphological template sets to build the I-Map structure containing all recognized morphological derivatives for every root. The constructed I-Map is implemented as a red-black tree in order to guarantees O(log n) complexity for both *lookup* and *put* operations. The process of generating the morphological derivatives of every root pair consists of two main tasks:

1. Identifying the appropriate morphological template set to be applied. This is done by going through a pre-defined decision tree of rules, constructed based on a linguistic expertise where the nodes represent some specific characteristics of the perfect or imperfect tense of the root and the leaves of the tree are the different morphological template sets. These rules include the length of the perfect and/or the imperfect tense of the root and their vocalization.
2. Constructing the morphological derivatives by replacing the strong letters of each morphological template of that template set by the morpheme letters of the root.

3.1.2. Filtering Stop and Special Words

The performed tests have proven that removing stop and special words guaranties a considerable increase in the overall performance of the algorithm. Stop words are the common words (such as ليس, فقد, لكن…) that frequently appear in texts and hold no semantic information. And special words are the ones that are naturally used as they are without morphological transformations. An example of special words include proper names, countries, currencies, numbers written in a letter form and chronological terms such as days and months. Both stop words and special words do not accept any inflectional transformations and hence, they are always used as they are. This characteristic is a sufficient reason to filter them out before initiating the root extraction process. A list of around 440 stop words consisting of Arabic pronouns and prepositions and a set of around 750 special words are maintained. In this phase, VAWRE produces only "Stop Word" or "Special Word" results.

3.1.3. Alif (ا) and Hamza (ء) Normalization

The changing forms of Alif and Hamza presents a real challenge when trying to automatically generate morphological derivatives for roots containing Alif and/or Hamza. The VAWRE algorithm does not include any intelligence to help in deciding which form of the Alif or the Hamza should be used while generating the morphological derivatives in case the root is made of an Alif and/or Hamza. We have developed a normalization model that covers all common forms of Alif and Hamza. This model eliminates the problem of changing forms of Alif and Hamza and effectively allow the correct roots to be extracted by providing a controlled flexibility while matching the processed word with the RE patterns representing the automatically generated derivatives.

Special Letters	Position in Word		
	First	***Middle***	***Last***
ا	ا آ أ	ا أ	ا أ
آ	آ ا أ	NA	NA
أ	أ ا إ ء ؤ ئ	أ ا إ ء ؤ ئ	أ ا إ ء ؤ ئ
إ	إ أ ا	إ أ ا	إ أ ا
آ	آ أ ا	NA	NA
ى	NA	NA	ى ا
ء	NA	ء ؤ ئ أ إ	ء ؤ ئ أ إ
ؤ	NA	ؤ ئ ء أ إ و	ؤ ئ ء أ إ
ئ	NA	ئ ؤ ء أ إ	ئ ؤ ء أ إ

Figure 1: Common forms of Alif (ا) and Hamza (ء) with corresponding possible form derivatives depending on the occurrence position.

This model stands as an intermediate phase between the forms of Alif / Hamza as they appear in the processed word and the automatically generated morphological derivative. To provide a better control over this allowed flexibility, the position of the Alif or Hamza in the word has been taken into consideration so that only the forms specific to that position are considered valid in a given context. Figure 8 shows the various forms of Alif and Hamza and the corresponding allowed forms depending on their occurrence position.

3.1.4. Root Extraction Process

In case the processed word is neither a stop word nor a special word, the root extraction process is initiated. Because the I-Map structure is huge and a lot of pattern matching needs to be done, traversing all the entries of the morphological structure is both unpractical and inefficient. It is unpractical because for every single word to be rooted, the algorithm will need to pattern match it with around 8,950 x 35 words. (8,950 root entries with an average of 35 morphological derivatives per root entry). This might take a long time of processing and probably days if we want to run our algorithm over a text collection of hundreds of megabytes. This non-selective traversal of the structure will be visiting all

the entries even the ones with no chance to contain the word we are trying to match. Our approach tries to approximate the ideal case where only the entries containing the processed word are being visited. Hence, the traversal of the structure is performed selectively in such a way that only the entries with roots having all their letters appearing in the original word are considered to be good candidates for further inspection. Once a row of the I-Map is identified to be a good candidate, the processed word is pattern matched at each time with a generated RE pattern that represent one of the morphological derivatives of that entry. This generated pattern is built with respect to the following three rules:

1. In order to resolve the non-standardized form problem of the Alif (ا) and Hamza (ء), every occurrence of these two letters is replaced by the appropriate replacement group of letters depending on the position in the word.

In case the word to be rooted has some affixes attached to it, the pattern must be made flexible enough to allow a match to happen. For that purpose, the pattern is appended with a header and a to any combination of consonants, vowel letters and diacritic marks.

```
trailer ([[أ-ي]* [ ّ ٔ ٕ ٓ ً ٌ ٍ َ ُ ِ ْ ٰ ]*]*) that corresponds
```

2. When the morphological templates in all the maintained template sets are made fully vocalized, the words we would like to extract their roots might have any of the previously described vocalization forms (full, partial or no vocalization). The pattern matching must provide a level of flexibility with respect to the vocalization marks to avoid the absence of some or all vocalization marks to prevent a matching to happen. For that purpose, the vocalization marks in the generated pattern appears as optional elements that help significantly in reducing the root extraction ambiguity but never prevent a match to happen if the diacritic marks have been omitted in the original word.

To illustrate the use of these three rules in generating the pattern representing the morphological derivatives, Figure 2 shows an example of the generated RE pattern of the morphological derivative ثِمَار .

```
[[أ-ي]* [ ّ ٔ ٕ ٓ ً ٌ ٍ َ ُ ِ ْ ٰ ]*]* ن [ ّ ٔ ٕ ٓ ً ٌ ٍ َ ُ ِ ْ ٰ ]* م [ ّ ٔ ٕ ٓ ً ٌ ٍ َ ُ ِ ْ ٰ ]* [ا أ إ آ] ر [[أ-ي]* [ ّ ٔ ٕ ٓ ً ٌ ٍ َ ُ ِ ْ ٰ ]*]*
```

Figure 2: The generated RE pattern of the morphological derivative ثِمَار .

As shown in Figure 2, the pattern is made of a header and a trailer, the exact letters found in the morphological derivative except of the Alif (ا) that has been replaced with the replacement group of letters with respect to the matrix shown in Figure 1 and the identified diacritic marks that have been made optional. In our algorithm, we always intentionally omit the case ending because it does not help in reducing form and root ambiguity.

In case a match occurs between the processed word and the RE pattern representing one of the morphological derivatives of a given entry, the header and/or trailer taking part of the match are looked up in the maintained list of prefixes and suffixes. If the matched parts of the header and/or the trailer are identified to be valid affixes (suffixes or prefixes), the match is considered to be a complete match. Only then, when at least one complete match occurs in a given entry, the root associated with that entry is considered to be a valid root for the word being rooted. For instance, if the processed word is استثمارات, and the I-Map entry of the root ث م ر is identified as the only candidate entry, the word استثمارات is pattern matched with all RE patterns representing every morphological derivative. When it is pattern matched with the RE pattern representing the morphological derivative ثِمَار (pattern shown in Figure 2), a match will occur and the resulting parts of the header and trailer being part of the match will be است and ات respectively. Then, the algorithm decides that the obtained match is a complete match only when both header and trailer parts are checked up to be valid affixes. This makes the root ث م ر to be reported as a valid root for the original word استثمارات and the algorithm produces the result "Unique Root Found". In case more than one candidate I-Map entries have been identified and a complete match has been obtained in more than one, all root entries are reported as valid roots and the algorithm produces the result "Multiple Roots Found". VAWRE produces the output "No Template Found" when at least one candidate is identified but no corresponding morphological template is found and "No Root Found" when no candidate root is identified.

4. Experimental Results and Analysis

The VAWRE algorithm is based on a set of Arabic support files that have been constructed based on a linguistic expertise and that have been incrementally improved through a sequence of tests. Table 1 provides the number of elements contained in the Arabic support files that have been used in the evaluation phase.

File Name	Number of Elements
Roots	8950
Morphological Template Sets	45
Prefixes	349
Suffixes	450
Stop words	414
Special words	750

Table 1: Arabic support files used by the VAWRE algorithm.

The VAWRE algorithm was evaluated on corpora made of Arabic text documents collected from the web. The collected Arabic documents were retrieved from different Arabic websites specialized in different domains such as economics, arts, science and sports and have been randomly split into three equal-sized sub-corpora in order to

eliminate any bias of the text domains on the results. The evaluation corpora is made of 127,606 word and has an average of around 11% of partially vocalized words and around 5% of fully vocalized words, which approximately reflect the ratios of the different vocalization forms as found in the modern Arabic texts. Table 2 provides the detailed description of the three sub-corpora used for the evaluation.

	Corpus 1		***Corpus 2***		***Corpus 3***		***Total Collection***	
Corpus Size	500 KB		500 KB		500 KB		**1.5 MB**	
Num. of Documents	93		83		82		**258**	
Words Processed	44,958		39,310		43,338		**127,606**	
Distinct Words	14,184		13,648		13,289		**41,121**	
non-vocalized words	37,456	83.31%	29,855	75.95%	35,394	81.67%	**106,705**	**83.62%**
semi-vocalized words	4,968	11.05%	6,640	16.89%	5,156	11.90%	**14,396**	**11.28%**
fully-vocalized words	2,534	5.64%	2,815	7.16%	2,788	6.43%	**6,505**	**5.10%**

Table 2: Description of the Arabic corpora used for root extraction evaluation.

Table 3 shows the obtained results for the three sub-corpora used in the evaluation. The *Root Extraction Performance* is calculated as the ratio of positive root extraction results ("Unique Root Found" and "Multiple Roots Found") over all (positive and negative) root extraction results ("Unique Root Found", "Multiple Roots Found", "No Template Found" and "No Root Found"). The *Overall Performance* is calculated as the ratio of all positive results ("Stop Word", "Special Word", "Unique Root Found" and "Multiple Roots Found") over all (positive and negative) possible VAWRE results.

	Corpus 1		***Corpus 2***		***Corpus 3***		***Total Collection***	
Stop Word	12,503	27.81%	10,031	25.52%	11,609	26.79%	**34,143**	**26.76%**
Special Word	1,413	3.14%	1,102	2.80%	1,421	3.28%	**3,936**	**3.08%**
Unique Root Found	14,718	32.74%	13,019	33.12%	13,750	31.73%	**41,487**	**32.51%**
Multiple Roots Found	10,269	22.84%	9,189	23.38%	9,897	22.84%	**29,355**	**23.00%**
No Template Found	6,037	13.43%	5,942	15.12%	6,622	15.28%	**18,601**	**14.58%**
No Root Found	18	0.04%	27	0.07%	39	0.09%	**84**	**0.07%**
Root Extraction Performance	80.49%		78.82%		78.02%		**79.13%**	
Overall Performance	**86.53%**		**84.82%**		**84.63%**		**85.36%**	

Table 3: Root extraction results of the conducted evaluation.

Eliminating stop words and special words at the filtering stage helped in increasing the overall performance of the VAWRE algorithm since 26.76% of the processed words have been identified to be stop words and 3.08% have been identified to be special words. The

average root extraction performance is about 80%, characterized by a very low percentage of "No Root Found" result, which proves that the used root dictionary is comprehensive and covers all most frequently used roots in modern text. Considering all possible results, the obtained results show that the VAWRE algorithm achieves an overall performance estimated to be around 85%.

Further Human analysis is needed to check the linguistic correctness of the roots generated by the VAWRE algorithm. Obviously, this human analysis covers only the two positive root extraction results: "Unique Root Found" and "Multiple Roots Found", where the VAWRE algorithm returns a single root or a list of possible roots respectively.

	Corpus 1		**Corpus 2**		**Corpus 3**		**Total Collection**	
Unique Root Found	6100		5817		5610		17527	
Correct Unique Root	4368	71.61%	3865	66.44%	3911	69.71%	12144	69.25%
Multiple Roots Found	4713		4401		4323		13437	
Correct Multiple Roots	4255	90.28%	3833	87.09%	3694	85.45%	11782	87.61%
Root Extraction Correctness	**79.75%**		**75.34%**		**76.56%**		**77.22%**	

Table 4: Results of correctness analysis of the roots generated by VAWRE.

The human analysis considers the root extraction result generated by VAWRE as a positive if the linguistically correct root is the one returned (in case of "Unique Root Found" result) or appears in the list of possible roots (in case of "Multiple Roots Found" result). Table 4 tabulates these results. The *Root Extraction Correctness* is the ratio of linguistically correct roots generated over the sum of unique roots and multiple roots obtained by the VAWRE algorithm.

Based on the performed human correctness analysis of the obtained root extraction results, the VAWRE algorithm achieves an average overall root extraction correctness of around 77%.

To verify the impact of considering vocalization in root extraction, we applied the root extraction algorithm on the same three evaluation sub-corpora but with no consideration of the diacritic marks attached to the semi-vocalized and fully-vocalized words. Table 5 shows the obtained results along with the improvement measure that quantify the root ambiguity reduction. The root ambiguity reduction is computed as the reduction of the number of words with "Multiple Roots Found" result or the increase of the number of words with "Unique Root Found" result.

	Corpus 1		Corpus 2		Corpus 3	
	With Voc.	**Without Voc.**	**With Voc.**	**Without Voc.**	**With Voc.**	**Without Voc.**
Unique Root Found	17,837	14,718	16,720	13,019	16,602	13,750
Multiple roots Found	7,150	10,269	5,488	9,189	7,045	9,897
Root Ambiguity Reduction	**30.37%**		**40.28%**		**28.82%**	

Table 5: Root ambiguity reduction results due to the use of vocalization.

In the three evaluation sub-corpora, a root ambiguity reduction is noticed when vocalization is considered in root extraction. The average root ambiguity reduction is around 33.16% for the total text collection.

5. Conclusions and Future Works

We have proposed and implemented an efficient Vocalized Arabic Word Root Extraction (VAWRE) algorithm based on a manually constructed dictionary of roots and a maintained list of morphological template sets. The developed algorithm handles three central challenges in the Arabic language: 1. The non-concatenative nature of the Arabic morphology, 2. The changing forms of both the Alif and Hamza and 3. The non-standardized vocalization style. Moreover, our work on Arabic root extraction is very special in a sense that much linguistic expertise has been invested in maintaining the roots dictionary and the vocalized morphological template sets.

Based on the conducted experimental evaluation, the root extraction performance of the VAWRE algorithm reached 85% with an average correctness of around 77%. Moreover, the obtained comparative results have proven that considering vocalization reduce root extraction ambiguity with around 33%. Now that the basis for a precise vocalized root extraction algorithm has been developed, future work will focus on carefully analyzing the generated root extraction results in order to augment the manually constructed thesauri with the missing basic roots and morphological templates. Another way to extend this work is to use of a morphological analyzer [23] in order to augment root extraction process with context information (verbs / nouns)

References

[14] Al Bahhar, www.albahhar.com

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

[13] Ayna, www.ayna.com

[17] Beesley, K.R.: 1996, "Arabic Finite-State Morphological Analysis and Generation", In Proc. of the 16th Int. Conf. on Computational Linguistics, Volume **1** (pp. 89-94).

[23] Buckwalter, T.: 2004, Buckwalter Morphological Analyzer Version **2**.0, Linguistic Data Consortium, University of Pennsylvania.
[3] Chekayri (ms), A.: 2006, Arabic Grammar, an Introduction. [3]
[19] Darwish, K.: 2002, "Building a Shallow Morphological Analyzer in One Day," ACL Workshop on Computational Approaches to Semitic Languages: 47-54.
[15] De Roeck, A., Al-Fares, W.: "A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots", Proc. of the 38th Annual Meeting of the ACL, Hong Kong.
[1] El Kourdi, M., Rachidi, T., Bensaid, A., Chekayri, A. and Mhamdi, M.: 2006, "A Concatenative Approach to Non-vocalized Arabic Root Extraction", Submitted to 6th Conference on Language Engineering, Egypt.
[20] Frakes W.B., Baeza-Yates R.: 1992, *Information Retrieval: Data Structures & algorithms*. Prentice Hall, Engelwood Cliffs, New Jersey.
[18] Glide: 2006, Encyclopedia of Arabic Language and Linguistics.
[12] Google, www.google.com
[24] Habash, Nizar and Owen Rambow , MAGEAD: 2006 "A Morphological Analyser and Generator for the Arabic Dialects," Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pages 681–688, Sydney, July 2006.
[25] Habash, Nizar, Rambow, Owen and Kiraz, George: 2005, "Morphological Analysis and Generation for Arabic Dialects," Proceedings of the ACL Workshop on Computational Approaches to Semitic Languages, pages 17–24, June 2005.
[11] Hahooa, www.hahooa.com
[7] Harman, D.: 1991, "How effective is suffixing?", Journal of the American Society for Information Science, 42(1),7-15.
[16] Ibn Manzour, *Lisan Al-Arab*. www.muhaddith.org.
[8] Idrisi, www.sakhr.com
[5] Ištiqāq, 2006, Encyclopedia of Arabic Language and Linguistics.
[9] Konouz, www.konouz.com
[22] Lovins, J.B.: 1998, "Development of Stemming Algorithm", Mechanical Translation and Computational Linguistics, Volume **11**.
[10] Maktoob, www.maktoob.com
[21] Porter, M.: 1980, "An Algorithm for suffix stripping", Volume **14**, No 3.
[2] Rachidi, T., Iraqi, O., Bouzoubaa, M., Ben Al Khattab, A., El Kourdi, M., Zahi, A., and Bensaid, A.: 2003, "Barq: distributed multilingual Internet search engine with focus on Arabic language," in proc.of IEEE conf. On Sys., Man and Cyber.
[4] Yates, R.B., Neto, B.R.: 1999, *Modern Information Retrieval*. ACM Press.