A Markovian Approach for Arabic Root Extraction

Abderrahim Boudlal[1], Rachid Belahbib[2], Abdelhak Lakhouaja[3], Azzeddine Mazroui[3], Abdelouafi Meziane[3], and Mohamed Bebah[3]
[1] Faculty of Letters and Human Sciences, University Mohamed I, Morocco
[2] College of Arts and Sciences, Qatar University, Qatar
[3] Department of Mathematics and Computer Sciences, University Mohamed I, Morocco

Abstract: *In this paper, we present an Arabic morphological analysis system that assigns, for each word of an unvoweled Arabic sentence, a unique root depending on the context. The proposed system is composed of two modules. The first one consists of an analysis out of context. In this module, we segment each word of the sentence into its elementary morphological units in order to identify its possible roots. For that, we adopt the segmentation of the word into three parts (prefix, stem, suffix). In the second module we use the context to identify the correct root among all the possible roots of the word. For this purpose, we use a Hidden Markov Models approach, where the observations are the words and the possible roots represent the hidden states. We validate the approach using the NEMLAR Arabic writing corpus consisting of 500,000 words. The system gives the correct root in more than 98% of the training set, and in almost 94% of the words in the testing set.*

Keywords: *Arabic NLP, morphological analysis, root extraction, hidden Markov models, and Viterbi algorithm.*

Received February 21, 2009; accepted August 3, 2009

1. Introduction

The morphological analysis is an important tool in all areas of scientific research and industry that require knowledge of the internal structure of the words.

In Arabic morphological analysis area, root extraction has attracted considerable attention among researchers. Indeed, researches in areas such as automatic document categorization [14], automatic summarization [12] and Text Mining [13] have shown great interest to root extraction. In addition, Arabic root extraction is usually linked to Information Retrieval (IR) systems and precisely to the indexing process. Several studies suggested that indexing Arabic text using roots significantly increases retrieval effectiveness over the use of words or stems [2]. Thus, many Arabic and multilingual search engines [17, 20, 21] make use of Arabic root extraction algorithms in order to overcome the inefficiencies in the precision and recall [24].

It is well known that morphological analysis process often leads to multiple interpretations. This problem of ambiguity is more serious in Arabic language given its morphological richness and complexity. Ambiguity is also increased by the absence of the short vowels in the majority of available documents.

There has been a considerable amount of works on Arabic morphological analysis [3, 5, 15, 24]. Xerox Arabic Finite State Morphology [6] and Buckwalter Arabic Morphological Analyzer (BAMA) [7, 8] are two of the best known morphological analyzers for Arabic language.

Many of works in Arabic morphological analysis have been devoted to the development of techniques for Arabic root extraction. Al Fedaghi *et al*. [1] presented an algorithm to generate the root and the pattern of a given Arabic word. Their algorithm deals only with triliteral word roots. While Kanaan *et al*. [19] proposed an algorithm to extract quadrilateral Arabic roots. Darwish [9] presented a hybrid approach combining linguistic rules with statistical informations in order to identify the possible roots of a given Arabic word. More recently, Rachidi *et al*. [24] have studied the effect of vowelization on Arabic root extraction.

However, in all these works, the analysis of words is done out of context. Thus, the morphological analysis process gives several possible roots because the algorithms mentioned above do not take into account the position of the word in the sentence.

In this work, we propose an Arabic root extractor in which the position of the word in the sentence is taken. The system gives in the first step of analysis a set of possible roots for each word of the sentence. In the second step, a Hidden Markov Models (HMM) approach is used to choose the most likely root of each word depending on the context.

The paper is organized as follows. In section 2, we provide an outline of the Arabic morphology, with emphasis on the segmentation of words used in this system. Section 3 is devoted to a description of our system. We first describe the out of context analysis

module which provides, for each word, the set of its possible roots. Then, a Markovian approach will be developed in order to identify the most likely root given the context. Section 4 presents how we have trained the Markovian model using an annotated corpus. Finally, an evaluation, using this annotated corpus, is given in section 5. Conclusion and future works are presented in section 6.

2. Arabic Morphology

The Arabic lexicon includes three categories of words: nouns, verbs and particles. Arabic language is characterized by its very rich derivational morphology, where almost of the words are derived from roots by applying patterns [9]. In Arabic, there is around 10000 roots, which are linguistic units of meaning composed of three or four (rarely five) letters. Around 85% of the Arabic words are derived from triliteral roots [11].

Identifying roots and patterns of the Arabic words, encounters two principal difficulties: The first is the absence of vowelization in the majority of Arabic texts. This absence causes ambiguity in 74% of Arabic words [10], i.e., in absence of vowelization, the words will accept several readings, and can derive them from several roots. The second difficulty comes from the fact that the majority of Arabic words are composed by agglutination of basic lexical elements. Then, a segmentation of the word into its elementary lexemes is necessary in order to isolate the part containing the letters of the root and consequently to identify root and pattern of the word.

Several types of segmentations are proposed in the literature. We adopted the segmentation widely used [7, 8, 9], which consists in dividing the word into three parts: (prefix + stem + suffix).

The prefix binds before the stem and inform about the tense of the verb: ي, ن, and determination of the word: ال. The stem constitutes the core of the word containing the letters of the root. It informs about the root and the pattern of the word. The suffix binds after the stem like: ين, تا, ة, ات, ون. It informs about the termination of the conjugated verb, the gender and noun number.

Thus the word: فسيدرسونه accepts the following segmentation:

Table 1: The decomposition of the word **فسيدرسونه**

Suffix	Stem	Prefix
ونه	درس	فسي

3. System Description

The root extraction process, in the system, consists of two modules as shown in Figure 1. In The first module, each word in the sentence is analyzed, regardless of its context, in order to identify its possible roots.

The second module deals with the graph resulting from the first module as shown in Figure 2, and using the Viterbi Algorithm, searches the most likely root sequence through the graph. This module is based on a Markovian model that had been trained using an annotated corpus containing about 500.000word.

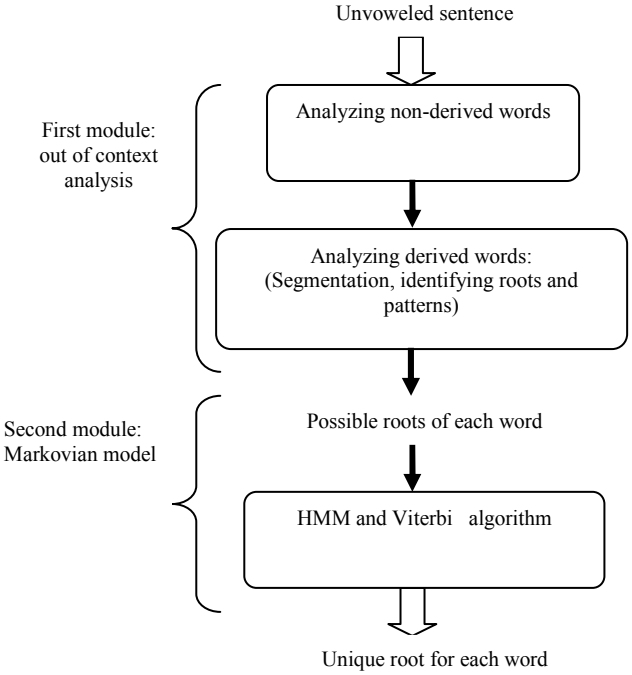


Figure 1. System architecture.

3.1. First Module: Out of Context Analysis

The analysis in this module consists of two steps:

3.1.1. Analyzing Non-Derived Words

It is well known that automatic root extraction is a task that concerns in particular the Arabic derived words. However, we have extended, in this work, the notion of root in order to deal with non-derived Arabic words. We do this in accordance with the annotations of the NEMLAR Arabic writing corpus [4, 18] that we used in both training and testing the system. This allows us, for example, to group, under the root "إلى", many words like: إليهما, وإليكم, and under the root "إلكترون" words like: الإلكتروني, والإلكترونيات.

We have established a database of non-derived words accompanied by their roots. This database serves us at this step, to identify the possible roots of a non-derived word. It is important to note that a non-derived Arabic word can accept more than one possible root. For example, the word من can come from the root "مَنْ" or from the root "مِنْ ".

3.1.2. Analyzing Derived Words

The objective in this step is to generate the possible roots of a given derived Arabic word. The analysis takes place in two stages:

a. Segmenting the word: the system begins by determining all possible segmentations of the word. For this purpose, the system uses the following resources:

- A dictionary of prefixes with 124 prefixes.
- A dictionary of suffixes with 200 suffixes.
- A compatibility table for prefixes-suffixes.

For each combination, the prefix and suffix are checked whether they are contained in the dictionaries or not. If so, the compatibility between them is considered; if they are compatible, this combination is considered as a possible segmentation of the word. The verification of compatibility will reduce considerably the number of possible segmentations of the word.

Table 2 gives an example of possible segmentations of the word وجد:

Table 2. Possible segmentations of the word وجد.

Suffix	Stem	Prefix
Ø	وجد	Ø
Ø	جد	و

where the Ø character represents the empty prefix or suffix.

b. Identifying possible patterns and roots: after recognizing the possible segmentations of the word, the system carries out an analysis of the stems in order to identify, in the order, the possible patterns and roots of the word. For this, the system uses the following resources:

- A database of Arabic patterns containing 160 patterns. Each pattern in this database is accompanied by a field indicating the positions of the root letters. For example, the pattern فاعل is accompanied, in this database, by a field indicating the positions 1, 3 and 4 as positions of the root letters.
- A dictionary of Arabic roots containing more than 9000 roots. Thus, the two stems وجد and جد, resulting from the segmentation of the word وجد in the previous example Table 2, are analyzed as follows: the system compares the stem وجد with the patterns in the database and recognizes the pattern فعل as possible pattern of the word. Then, the system identifies the root و ج د as a possible root of the word.

The system recognizes the pattern فعل as possible pattern of the stem جد. The patterns consisting of two letters may result either from a doubled root (root with the second letter doubled such as ج د د) or from a root containing weak letters (و, ي or ا). This allows the system to recognize many roots as possible roots of the stem جد such as ج د د, و ج د, ج و د, and ج ي د.

3.2. Second Module: Markovian Model

The choice of the most likely root will be taken depending on the position of the word in the sentence. For this purpose, we will use a Markovian modeling of Arabic sentences. We begin by recalling the definition of HMM [23].

Let O = {o_1, o_2, …, o_M} be a finite set of observations and S = {s_1, …, s_N} be a finite set of hidden states (unknown).

Definition: A first-order HMM is a double process $(X_t, Y_t)_{t\geq 1}$ where:

- $(X_t)_t$ is a homogeneous Markov chain with values in the hidden states set S where:

$$\Pr(X_{t+1} = s_j / X_t = s_i, \ldots, X_1 = s_h) = \Pr(X_{t+1} = s_j / X_t = s_i) = a_{ij} \quad (1)$$

a_{ij} is the transition probability from state s_i to state s_j.

- $(Y_t)_t$ is an observable process taking values in the observations set O where:

$$\Pr(Y_t = o_k / X_t = s_i, Y_{t-1} = o_{k_{t-1}}, X_{t-1} = s_{i_{t-1}}, \ldots, Y_1 = o_{k_1}, X_1 = s_{i_1}) = \Pr(Y_t = o_k / X_t = s_i) = b_i(k) \quad (2)$$

$b_i(k)$ is the probability of observing o_k given the state s_i.

In the following, we assume that the observations set W = {w_1, w_2, …, w_M} consists of Arabic words while the hidden states set $\Re = \{r_1, \ldots, r_N\}$ consists of Arabic roots. Let S be an observed sentence consisting of the sequence of words $w_1, \ldots, w_n$. The goal is to find the most likely sequence of roots $(r_1^*, \ldots, r_n^*)$ given the sentence S. This goal can be formulated as follows:

$$(r_1^*, \ldots, r_n^*) = \arg \max_{r_1 \ldots r_n \in \Re} \Pr(r_1 \ldots r_n / w_1 \ldots w_n) \quad (3)$$

Since

$$\Pr(r_1 \ldots r_n / w_1 \ldots w_n) = \frac{\Pr(w_1 \ldots w_n / r_1 \ldots r_n) \Pr(r_1 \ldots r_n)}{\Pr(w_1 \ldots w_n)}$$

Then, the sequence $(r_1^*, \ldots, r_n^*)$ satisfies:

$$(r_1^*, \ldots, r_n^*) = \arg \max_{r_1 \ldots r_n \in \Re} \Pr(w_1 \ldots w_n / r_1 \ldots r_n) \Pr(r_1 \ldots r_n) \quad (4)$$

Given that language can be seen as a Markov source [18], then we will consider the following assumptions:

- The succession of roots in a sentence is a homogeneous Markov chain, therefore, for any $r_1 \ldots r_k \in \Re$ k possible roots of the sequence $w_1 \ldots w_k$ in the sentence S we have:

$$\Pr(r_k / r_1 \ldots r_{k-1}) = \Pr(r_k / r_{k-1}) \quad (5)$$

- Prediction of the word w_k only require knowledge of it root regardless of neighbouring words and their roots. Thus,

$$\Pr(w_k / r_k, w_{k-1}, r_{k-1}, \ldots, w_1, r_1) = \Pr(w_k / r_k) \quad (6)$$

In going from the assumption (A1), we check easily that: for $k \leq n$

$$\Pr(r_1 \ldots r_k) = \prod_{i=1}^{k} \Pr(r_i / r_{i-1}) \quad (7)$$

(with the convention $\Pr(r_1 / r_0) = \Pr(r_1)$). Likewise, the assumption (A2) implies:

$$\Pr(w_1 \ldots w_n / r_1 \ldots r_n) = \prod_{i=1}^{n} \Pr(w_i / r_i) \quad (8)$$

The first module in the system (Analysis out of context) gives, for each word w_i, in the sentence S a list of possible roots $(r_i^1, \ldots, r_i^{n_i})$. Let $\Re_i = \{r_i^1, \ldots, r_i^{n_i}\}$ be the set of these possible roots. Then $r_i^* \in \Re_i$.

Thus, it is sufficient to take in equation 4 the maximum on all sets $\Re_i$ and as a result we have the following equation:

$$(r_1^*, \ldots, r_n^*) = \arg \max_{\substack{r_i^{j_i} \in \Re_i \\ 1 \leq i \leq n}} \Pr(w_1 \ldots w_n / r_1^{j_1} \ldots r_n^{j_n}) \Pr(r_1^{j_1} \ldots r_n^{j_n}) \quad (9)$$

The solution $(r_1^*, \ldots, r_n^*)$ of the equation 9 can be calculated by searching the most likely path through Figure 2.

The viterbi algorithm [16] is well suited for finding the most likely path. In the following, we pose:

$$\phi(t, r_t^k) = \max_{\substack{r_i^{j_i} \in \Re_i \\ 1 \leq i \leq t-1}} \Big[\Pr(w_1, \ldots, w_{t-1}, w_t / r_1^{k_1}, \ldots, r_{t-1}^{k_{t-1}}, r_t^k) \times \Pr(r_1^{k_1}, \ldots, r_{t-1}^{k_{t-1}}, r_t^k)\Big] \quad (10)$$

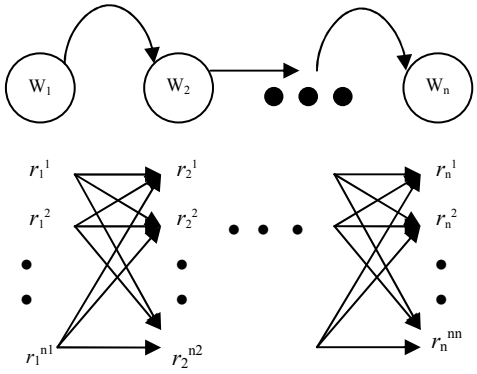


Figure 2. Graph resulting from the analysis out of context for the sentence W1…,Wn.

$\phi(t, r_t^k)$ is the probability of the most likely partial path until time t, and ending at the root r_t^k (r_t^k belongs) to the possible roots set of the word w_t). By using the equations 7 and 8 we get:

$$\begin{aligned}\phi(t, r_t^k) &= \max_{\substack{r_i^{j_i} \in \Re_i \\ 1 \leq i \leq t-1}} \prod_{i=1}^{t-1} \Big[\Pr(w_i / r_i^{j_i}) \times \Pr(r_i^{j_i} / r_{i-1}^{j_{i-1}}) \\ &\qquad \times \Pr(w_t / r_t^k) \times \Pr(r_t^k / r_{t-1}^{j_{t-1}})\Big] \\ &= \Big(\max_{r_{t-1}^{j_{t-1}} \in \Re_{t-1}} \phi(t-1, r_{t-1}^{j_{t-1}}) \times \Pr(r_t^k / r_{t-1}^{j_{t-1}})\Big) \Pr(w_t / r_t^k) \\ &= \Big(\max_{r_{t-1}^{j} \in \Re_{t-1}} \phi(t-1, r_{t-1}^{j}) \times \Pr(r_t^k / r_{t-1}^{j})\Big) \Pr(w_t / r_t^k) \quad (11)\end{aligned}$$

This equation will allow us to find, recursively, the values of ϕ. In order to get the best path, we use a variable ψ that memorizes, at each time t, the root giving the maximum in equation 12 ψ is defined as follows:

$$\psi(t, r_t^k) = \arg \max_{r_{t-1}^{j} \in \Re_{t-1}} \phi(t-1, r_{t-1}^{j}) \Pr(r_t^k / r_{t-1}^{j}) \quad (12)$$

notice that $\psi(t, r_t^k) \in R_{t-1}$.

The equations 11 and 12 allow us to get the best path using the following recursive algorithm.

- Step 1: Initialization
  for $1 \leq k \leq n_1$
  $\phi(1, r_1^k)$ = Pr(having the word w_1 at the beginning of the sentence S with the root r_1^k).
- Step 2: Recursion
  for $2 \leq t \leq n$ and $1 \leq k \leq n_t$:

$$\phi(t, r_t^k) = \Big(\max_{r_{t-1}^{j} \in \Re_{t-1}} \phi(t-1, r_{t-1}^{j}) \times \Pr(r_t^k / r_{t-1}^{j})\Big) \Pr(w_t / r_t^k)$$

$$\psi(t, r_t^k) = \arg \max_{r_{t-1}^{j} \in \Re_{t-1}} \phi(t-1, r_{t-1}^{j}) \Pr(r_t^k / r_{t-1}^{j})$$

- Step 3: Final state

$$\psi(n+1) = \arg \max_{r_n^j \in \Re_n} \phi(n, r_n^j).$$

- Step 4: Deducing the best path

- $r_n^* = \psi(n+1)$,
- For t = n-1:1 $r_t^* = \psi(t, r_{t+1}^*)$.

4. Training

To be able to run the Viterbi algorithm and identify the correct root of the word given the context, it must first go through a training phase, which consists in estimating the different probabilities that appear in equations 11 and 12, in order to calculate the values of the functions ϕ and ψ.

For equations 11 and 12 we need the probabilities of each word for each possible root: $\Pr(w_t/r_t)$ and the transition probabilities between consecutive roots: $\Pr(r_t/r_{t-1})$. A very useful way to get these is from a corpus which has been annotated by hand.

4.1. Training Corpus

The corpus, which we used to train the Markovian model and evaluate the performance of the system, is the NEMLAR Arabic Writing Corpus. The corpus consists of about 500,000 words of journalistic Arabic texts from different categories. The Corpus was produced and annotated by RDI, Egypt for the Nemlar Consortium [4,18]. Each Arabic word in this corpus is replaced by its lexical analysis in the following notation:

{Wv;(Tn) T: (Pn) P, (Rn) R, (Fn) F, (Sn) S}

where *Wv* is the vowelized mnemonic of the vowelized full form word, *Tn* is the mnemonic of word type, *T* is the ID of the word type, *Pn* is the mnemonic of word prefix, *P* is the ID of the word prefix, *Rn* is the mnemonic of word root, *R* is the ID of the root prefix, *Fn* is the mnemonic of word pattern, *F* is the ID of the word pattern, *Sn* is the mnemonic of word suffix, and *S* is the ID of the word suffix.

For example, the word الموقع is lexically analyzed as follow:

{ (),800 (مَفْعِل),4380 (وقع),9 (الـ):1 (مصرَّفةٌ منتظِمة);ٱلمَوْقِعُ}

We used this corpus initially to enrich the dictionary of Arabic roots, which is used in the first module of the system. Also, we have used 93% of word-root pairs extracted from the corpus in order to train the Markovian model used in the second module.

4.2. Estimation of Probabilities

The estimation of probabilities was carried out, by counting the frequencies of words and roots in the training corpus.

For that, we define the following quantities:

- $(Occ(r_{t-1}, r_t))$= The number of times the root r_{t-1} appears in the training corpus followed by the root r_t.
- Occ(r_{t-1}, r_1)= The number of times the root r_t appears in the training corpus.
- Occ(w_t, r_1)= The number of times the word w_t appears in the training corpus with the root r_t.

The probabilities in equations 11 and 12 will be estimated using the following expressions:

$$\Pr(r_t/r_{t-1}) = \frac{Occ(r_{t-1}, r_t)}{Occ(r_{t-1})} \quad (13)$$

$$\Pr(w_t/r_t) = \frac{Occ(w_t, r_t)}{Occ(r_t)} \quad (14)$$

Due to the limitations of the training corpus, the numbers $Occ(r_{t-1}, r_t)$, $Occ(r_t)$ and $Occ(w_t, r_t)$ can be nulls. That poses problems to calculate the values of the functions ϕ and ψ which are necessary to run the Viterbi algorithm. In order to cure this problem, we estimated $\Pr(r_t/r_{t-1})$ by ε when $Occ(r_{t-1}, r_t)$ is zero. In the same way, we estimated $Occ(w_t, r_t)$ by λ when $Occ(w_t, r_t)$ is zero. ε and λ are given by the following equations:

$$\varepsilon = \frac{1}{2} \min_{\substack{r,\bar{r}\in\Re \\ Occ(r,\bar{r})\neq 0}} \frac{Occ(r,\bar{r})}{Occ(r)} \quad (15)$$

$$\lambda = \frac{1}{2} \min_{\substack{(w,r)\in Wt\times\Re \\ Occ(w,r)\neq 0}} \frac{Occ(w,r)}{Occ(r)}. \quad (16)$$

where Wt is the set of words in the training corpus.

5. Experimental Results

Tests were carried out, for the two modules, on two subsets of the NEMLAR Arabic Writing Corpus:

- The first set, called Tr, contains 92965 words from the training corpus. Tr constitutes approximately 19% of the training corpus. The training corpus constitutes 93% of the total NEMLAR Arabic Writing Corpus.
- The second set, called Te, constitutes the remaining 7% that was not used in the training phase. Te consists of 38022 words.

In the first module, the evaluation method consists in comparing, for each word, the root assigned by the annotators to the list of roots generated by the system. If the root is on the list, we consider that the system analyzed successfully the word. We have also performed a comparison between the first module of this system and Darwish's root extractor [9] which was the only Arabic root extractor that we could download and evaluate.

In the second module, the system keeps only one root for each word. We compare this root with that assigned by the corpus annotators.

5.1. Results from the First Module

We started by testing the effectiveness of the first module to find the root assigned by the annotators possibly accompanied by other roots. We noted that the number of roots generated by the system varies from 1 to 12 roots. Table 3 gives the distribution of the words in four classes, according to the number of possible roots resulting from the first module (analysis out of context). We note that multiple roots are generated in more than 52% of the cases.

Table 3. Distribution of words according to the number of roots found in the first module (analysis out of context).

	Nbr of words		Percentage	
	The set *Tr*	The set *Te*	The set *Tr*	The set *Te*
Unique root	44169	18110	47.51%	47.63%
Two roots	14557	5920	15.65%	15.56%
Three roots	11005	4485	11.83%	11.79%
Four roots or more	23234	9507	25%	25%

Moreover, the system provides, in more than 99% of the cases, the correct root among the list of possible roots as shown as Table. 4.

Table 4. Root extraction results for the first module.

Analysis out of context	Nbre of words	Nbr of words of wich the good root appears among the list of possible root	Percentage
The set *Tr*	92965	92271	99.25%
The set *Te*	38022	37715	99.19%

In addition, we tested, in the same way, Darwish's root extractor on a set of 60,000 derived words. Darwish's system succeeded in generating the correct root, among the list of possible roots, only in 86.34% of the cases.

5.2. Results from the Second Module

Then, we tested the effectiveness of the second module to chose, the correct root of each word taking into account the context. The system has chosen the correct root in more than 98% of the training set *Tr*, and in almost 94% of the words in the testing set *Te* as shown as Table 5.

Table 5. Root extraction results for the second module.

Second module	Nb of words	Nbr of recognized correct roots	Percentage
The set *Tr*	92965	91403	98,31%
The set *Te*	38022	35672	93,81%

System effectiveness have ranged from 98% in the training set *Tr* to 93.81% in the testing set *Te* due to the large number of couples (r_{t-1}, r_t) of the testing set that doesn't appear in the training set.

6. Conclusions

We presented in this paper a morphological analysis system for unvoweled Arabic sentences. The morphological analysis process often gives multiple solutions. We showed that by introducing the context, an approach based on hidden Markov models can have very good results in choosing the correct root of the word. The results of tests carried out on both parts of the system are very encouraging. They can be improved by further analysis of hamzated words in the analysis out of context, and by using a larger corpus in the markovian approach.

Currently, we extend the work of the first module in order to generate other tags of the words (noun, verb, particle, adjective, adverb, possible vowelizations, …). Then, we use an adaptation of the Markovian approach to identify the best vowelization of the word in context.

References

[1] Al-Fedaghi S. and Al-Anzi F., "A New Algorithm to Generate Arabic Root-Pattern Forms," *in Proceedings of the 11th National Computer Conference*, King Fahd University of Petroleum & Minerals, Dhahran, pp. 04-07, Saudi Arabia, 1989.

[2] Al-Kharashi I. and Evens W., "Comparing Words, Stems, and Roots as Index Terms in an Information Retrieval," *JASIS*, pp. 548-560, 1994.

[3] Al-Sughaiyer I. and Al-Kharashi I., "Arabic Morphological Analysis Techniques: A Comprehensive Survey," *Journal of the American Society for Information Science and Technology*, vol. 55, no. 3, pp. 189-213, 2004.

[4] Attia M., Yaseen M., and Choukri K. Specifications of the Arabic Written Corpus produced within the NEMLAR project; www.NEMLAR.org, 2005.

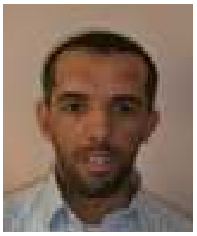
[5] Attia M., "An Ambiguity-Controlled Morphological Analyzer for Modern Standard Arabic Modelling Finite State Networks," *The Challenge of Arabic for NLP/MT Conference*, The British Computer Society, London, 2006.

[6] Beesley K., "Finite-State Morphological Analysis and Generation of Arabic at Xerox Research: Status and Plans in 2001," *in the Proceedings of the Arabic Language Processing: Status and Prospect-- 39th Annual Meeting of the Association for Computational Linguistics*, Toulouse, France, 2001.

[7] Buckwalter T., "Buckwalter Arabic Morphological Analyzer Version 1.0," Linguistic Data Consortium, University of Pennsylvania, LDC Catalog no. LDC2002L49, 2002.
[8] Buckwalter T., "Buckwalter Arabic Morphological Analyzer Version 2.0," Linguistic Data Consortium, University of Pennsylvania, 2004, Catalog Number LDC2004L02.
[9] Darwish K., "Building a Shallow Arabic Morphological Analyzer in One Day," *in the Proceedings of the workshop on Computational Approaches to Semitic Languages in the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 47-54 Philadelphia, USA, 2002.
[10] Debili F. and Achour H., "Voyellation Automatique de l'arabe," *in the Proceeding of the workshop on Computation approaches to Semitic languages*, COLING-ACL, Montréal, 1998.
[11] De Roeck A. and Al-Fares W., "A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots," *in the 38th Annual Meeting of the ACL.*, Hong Kong, 2000.
[12] Douzidia F. and Lapalme G., "Lakhas, an Arabic Summarization System," *in Proceedings of,* Boston, pp. 128-135, 2004.
[13] Eldos M., "Arabic Text Data Mining: A Root Extractor for Dimensionality Reduction," *IASTED International Conference on Applied Informatics*, Innsbruck, Austria, 2002.
[14] El Kourdi M., bic Document Categorization Based on thBensaid A., and Rachidi T., "Automatic arae Naïve Bayes Algorithm," *in Proceedings of COLING 20th Workshop on Computational Approaches to Arabic Script-based Languages,* Geneva, 2004.
[15] Habash N. and Rambow O., "MAGEAD: a morphological analyzer and generator for the Arabic dialects," *in the Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pp. 681-688, Australia, 2006.
[16] Huang X., Acero A., and Hon H., Spoken Language Processing, *Prentice Hall PTR*, New Jersey, 2001.
[17] Idrisi, www.sakhr.com
[18] Jgejan B. and Fersøe H., "Validation report Nemlar Arabic Written Corpus," Center for Sporgteknologi, 2006.
[19] Kanaan G., Al-Shalabi R., Jaam J., Al-Kabi M., and Hasnah A., " A New Stemming Algorithm to Extract Quadri-Literal Arabic Roots," *in Proceedings of International Conference on Information and Communication Technologies: From Theory to Applications,* Syria, 2004.
[20] Konouz, www.konouz.com, Last Visited 2008.
[21] Maktoob, www.maktob.com, Last Visited 2008.
[22] Neuhoff D., "The Viterbi Algorithm as an Aid in Text Recognition," *IEEE Transactions on Information Theory*, pp. 222-226, 1975.
[23] Rabiner L., "A tutorial on Hidden Markov Models and selected applications in speech recognition," *in the proceeding of Readings in speech recognition*, pp. 267-296, 1990.
[24] Rachidi T., Chekayeri A., Mhamdi M., Chhoul O., and Fala A., "The Effect of Full and Partial Diacritization on Arabic Root Extraction," *in Proceedings of CITALA*, pp. 189-200, Morocco, 2007.
[25] Yagi S. and Yaghi J., "Tracking Morphophonemic Transformation in Arabic Word Generation and Root Extraction," *International Arab Journal of Information Technology*, vol. 4, no. 3, pp. 229-236, 2007.

Abderrahim Boudlal Doctorat d'Etat in Arabic linguistic, University Mohammed I Morocco, 2001. He is professor of Arabic language in University Mohammed I, member of Laboratory of Researches in Computer Sciences (LaRI), cofounder of the ANLP unit in the LaRI laboratory member of the Moroccan Linguistics Association and member of CERHSO (Oujda Center of Humanity and Social Studies and Researches).

Rachid Belahbib Doctorat d'Etat in Arabic Linguistic, University Mohammed I Morocco, 1993. He is professor of Arabic language in Qatar University, member of (LaRI) Laboratory Cofounder of the ANLP unit in the LaRI laboratory member of the Moroccan Linguistics Association, and director of CERHSO (Oujda Center of Humanity and Social Studies and Researches).



Abdelhak Lakhouaja Doctorat d'Etat in computer sciences, University Mohammed I Morocco, 2000. He is professor of Computer Sciences in University Mohammed I, member of Laboratory of Researches in Computer Sciences (LaRI), and cofounder of the ANLP unit in the LaRI laboratory.

Azzeddine Mazroui Doctorat d'Etat in numerical analysis, University Mohammed I Morocco, 2000. and PhD in probability and statistics, Pierre & Marie Curie University France, 1993. He is professor of mathematics and computer sciences in University Mohammed I, member of Laboratory of

Researches in Computer Sciences (LaRI), cofounder of the ANLP unit in the LaRI laboratory, and cofounder of the UFR Doctorate in "Approximation, Computer Sciences and Signal Analysis".

Abdeluafi Meziane Doctorat d'Etat" in computer sciences, University Mohammed I Morocco, 1997, DES in probability and statistics, France, 1987. He is professor of mathematics and computer sciences in University Mohammed I, member of Laboratory of Researches in Computer Sciences (LaRI), cofounder of the ANLP unit in the LaRI laboratory, and cofounder of the UFR Doctorate in "Approximation, Computer Sciences and Signal Analysis".

Mohammed Bebah "DESA" in numerical analysis, computer science and signal processing" from Mohamed I University, Oujda, Morocco in 2005. Since January 2006, he prepares his PhD thesis in Arabic natural language Processing within the laboratory LaRI. His research interests are especially in Arabic morphological analysis and automatic diacritization.