International Journal of Pattern Recognition
and Artificial Intelligence
Vol. 28, No. 1 (2014) 1453001 (39 pages)

DOI: 10.1142/S0218001414530012

COLLABORATIVE COMBINATION OF NEURON-LINGUISTIC CLASSIFIERS FOR LARGE ARABIC WORD VOCABULARY RECOGNITION

AFEF KACEM ECHI* and IMEN BEN CHEIKH†

LaTICE-ESSTT, University of Tunisia
5 Av. Taha HUSSEIN Montfleury, Tunis
1008/Bab Menara, Tunisia
**afef.kacem@esstt.rnu.tn*
†imen.becheikh@gmail.com

ABDEL BELAÏD

LORIA, Unversity of Lorraine
Campus Scientifique. BP 239 54506
Vandoeuvre-lès-Nancy Cedex, France
abelaid@loria.fr

Received 28 January 2013
Accepted 18 November 2013
Published 10 January 2014

Most of the actual research in writing recognition focuses on specific applications where the vocabulary is relatively small. Many applications can be opened up when handling with large vocabulary. In this paper, we studied the classifier collaboration interest for the recognition of a large vocabulary of arabic words. The proposed approach is based on three classifiers, named Transparent Neuronal Networks (TNN), which exploit the morphological aspect of the Arabic word and collaborate for a better word recognition. We focused on decomposable words which are derived from healthy tri-consonantal roots and easy to proof the decomposition. To perform word recognition, the system extracts a set of global structural features. Then it learns and recognizes roots, schemes and conjugation elements that compose the word. To help the recognition, some local perceptual information is used in case of ambiguities. This interaction between global recognition and local checking makes easier the recognition of complex scripts as Arabic. Several experiments have been performed using a vocabulary of 5757 words, organized in a corpus of more than 17 200 samples. In order to validate our approach and to compare the proposed system with systems reported in ICDAR 2011 competition, extensive experiments were conducted using the Arabic Printed Text Image (APTI) database. The best recognition performances achieved by our system have shown very promising results.

Keywords: Large arabic word vocabulary; off-line writing recognition; classifier collaboration; transparent neuronal networks.

*Corresponding author.

1. Introduction

The automatic recognition of writing with a large vocabulary remains a challenging task despite the promising improvements in the latest recognition methods and systems. Most of the actual research in handwriting recognition focuses on specific applications such as bank processing,[20,27,30] mail storing,[29,51] commercial form-reading,[40] etc., where many constraints are imposed (size of vocabulary, writer-dependence, writing style, experimental conditions, etc.). One of the most common constraints of current recognition systems is that are only capable of recognizing words that are presented in restricted vocabulary, typically comprised of 10–1000 words.[45,50] By vocabulary, usually called a lexicon, we mean, as defined by Ref. 38, a list of valid words that are expected to be recognized by the system.

One[17] can classify the field of word recognition into four categories according to the size and nature of the lexicon involved: very large, large, limited but dynamic, and small and specific. Small lexicons do not include more than 100 words, while limited lexicons may go up to 1000. Large lexicons may contain thousands of words, and very large lexicons refer to any lexicon beyond that. When a dynamic lexicon (in contrast with specific or constant) is used, it means that the words that will be relevant during a recognition task are not available during training because they belong to unknown subset of a much larger lexicon.

The ability to handle large vocabularies however opens up many applications but recognizing a large vocabulary is a challenging task. As explained by Ref. 38, as the number of words in the lexicon grows, the more difficult the recognition task becomes, because more similar words are more likely to be present in the lexicon. The computational complexity is also related to the lexicon, and it increases relatively to its size.

It is clear that large vocabulary recognition is required in many applications and for any languages. In fact, one cannot have a lexicon representing all exhaustive words. This would make the work of the Optical Character Recognition (OCR) tedious. This is particularly true for Arabic language where there are billions of variations of words. In Ref. 46, a system for recognition of printed Arabic text based on Hidden Markov Models (HMM) is proposed. While HMMs have been successfully used in the past for such a task, authors reported on significant improvements of the recognition performance with the introduction of minimum and maximum duration models. They made improve-ments allowed to build a system working in open vocabulary mode, i.e. without any limitations on the size of the vocabulary. Later on, the system, proposed in Ref. 3 starts by segmenting the printed Arabic text into lines, then each line is also segmented into separate words, after that each word is further segmented to sub-words. Each word or sub-word is segmented into separate characters, and then a features extraction process is applied on each character to calculate its features vector. The minimum distance classifier is used in the classification stage. A lookup dictionary is employed to correct some of the misclassified characters. This resulted in improving the accuracy to 96.1%. It should be pointed that the majority of the errors are cases of either (1) mis-segmentation, due to

connectivity between letters which makes finding the letter borders hard, different forms of many letters according to their position in the word (in the beginning, in the middle or at the end) and variable length of letters based on their neighbors from left and right, or (2) mis-classification due to the similarity between some of the letters. The article, in Ref. 44 provides a comprehensive survey of recent developments in Arabic handwriting recognition.

The idea of research in this area is to assist the work of the recognizer system by the use of morphological decomposition mechanisms suitable for multiple words and for their recognition with the same rules. In fact, the Arabic language has an unlimited vocabulary size because of its derivative property. There is a reliance on verbs made up of three consonants (the "tri-consonantal root", as it is sometimes called) as the basic building blocks from which other elements of the language are derived, following a surprisingly regular set of word patterns (also called schemes).

Most of the works proposed for Arabic word recognition were mainly focused on feature extraction, training and classification methods. To handle with a large vocabulary of Arabic words and for meaningful improvements in recognition, we believe that is necessary to incorporate, into the recognition process, knowledge about language model. As mentioned by Ref. 19, current trends are for incorporating linguistic knowledge, either keeping them as an independent stage, or integrating them, partly or totally, in the recognition chain. As underlined by Ref. 35, most of the existing works integrate the knowledge of the Arabic language in the recognition processus in two ways using either a language dictionary to validate the word hypotheses suggested by the OCR, as done by Refs. 1, 26 and 28 or a statistical model of the language (HMM, N-gram)[8,23] as post processing. Within this framework, the linguistic properties of the Arabic vocabulary were never exploited to set up robust mechanisms for assumption filtering of the words within the recognition process.

In those respects, this paper presents a new approach for large vocabulary Arabic word recognition based on exploiting the morphological structures of the Arabic language. It presents a simple idea to represent Arabic word via its root, scheme and conjugation elements, described by global structural features and recognition is made with the help of neural-linguistic classifiers. Thanks to the collaborative combination of these classifiers, it is shown that the proposed system is able to consider a great number of characteristics, used by different classifiers, exploiting marginal performance and behavior from each one. A vocabulary of decomposable Arabic words which are derived from healthy tri-consonantal roots has been considered in this study.

The rest of this paper is organized as follows. Section 2 summarizes the main characteristics of Arabic morphology. Section 3 briefly introduces the Arabic vocabularies and the handled one. Section 4 sketches some related studies in Arabic writing recognition using linguistic knowledge. Section 5 focuses on Arabic linguistic knowledge integration in the writing recognition process. Section 6 overviews the proposed system based on the collaboration of three classifiers: transparent neuronal

networks (TNN). Section 7 provides and discusses experimental results. A conclusion and some directions for future work are drawn in Sec. 8.

2. Characteristics of Arabic Morphology

We propose, in this section, to analyze the Arabic language considering two aspects: its linguistic properties and writing characteristics. This would help to find some heuristics for developing large vocabulary methods.

2.1. *Linguistic proprieties*

- *Semitic language*: Arabic is a Semitic language that is based on the arabic alphabet containing 28 letters. Its basic feature is that most of its words are built up from, and can be analyzed down to common roots. The exceptions to this rule are common nouns and particles.
- *Inflection*: Arabic is a highly inflectional language with 85% of words derived from trilateral roots. Nouns and verbs are derived from a closed set of around 10 000 roots.[2] For example, the three consonants ش, ر and ب convey a basic idea equivalent to the English word "drink". From this root has been derived a simple verb شَرِبَ meaning "he drank". The simple verb (generally called "Form I") can then be altered into various ways to extend and refine the verbal idea in different directions. For instance, lengthening the vowel following the first consonant شَارَبَ often conveys the idea of doing the Form I action with someone or something else, or doing the action over a period of time. Thus شَارَبَ means "to have a drink with (someone)", "to drink in (someone's) company". There are similar patterns used to form nouns, adjectives and other parts of speech. For example to prefix the syllable م to a root, makes the first consonant vowel, and then follows the second consonant with the vowel ا results in a noun that refers to the place. In the case of the root شَرِبَ, for example, مِشرَاب means most generally "place for drinking" which can be used to refer more specifically to such varied objects as a watering hole, a drinking trough, a fountain or a restaurant bar.
- *Stability of linguistic concepts*: Arabic is known for its rich morphological structure in terms of stable linguistic concepts specific to that language. For example, in Arabic, the "noun pattern" is the active participle pattern where the first vowel of the root is a lengthened ا placed immediately after the initial consonant and a second vowel is inserted between the second and third root letters. One of the most common uses of these nouns is to designate the person or thing that performs the action of the Form I verb. From ش, ر, and ب, then, one can derive the noun شَارِب. One of the meanings of شَارِب is "drinker", English, somewhat similarly, uses the suffix "er" to indicate the doer of a verbal action. But English is much less regular in the way it uses such suffixes and prefixes than Arabic. "Drink-er" may illustrate our general rule perfectly, but "actor" and "flier" show that sometimes irregular vowel changes

occur when these words are derived. Similarly, the nouns "pilot" and "workman" show that we can use entirely different forms to designate the doer of the action.

- *Synthetic language*: Arabic is a synthetic rather than an analytic language. In other words, it uses special endings placed on nouns, adjectives and pronouns, called "cases", to indicate the function of one of these words in a sentence. English, being an analytic language, uses word order to perform this function: if a noun precedes the verb, it is assigned the function of subject ("doer of the action"). If it follows the verb, it will generally be considered the object ("recipient of the action", "thing acted upon"). Arabic can use word order to convey this information, and it often does. But it also (and more characteristically) uses special case endings to ensure the message is understood. In Arabic, the subject of a sentence would be identified by the vowel ـُ placed at the end of the word, and it would remain the subject regardless of where it was positioned in the sentence. For example, in both sentences: رَكِبَ الفَارِسُ الحِصَانَ and رَكِبَ الحِصَانَ الفَارِسُ (The jockey rode the horse), as shown in Figs. 1(a) and 1(b), the subject is الفَارِسُ (The jockey). The object would have the vowel ـَ suffixed to it (see Figs. 1(a) and 1(b)), and the objects of any prepositions would receive the suffix ـِ (e.g. ذَهَبَ الوَلَدُ إلي المَدرَسةِ, the boy goes to the school) as illustrated in Fig. 1(c).
- *Usage of nouns and adjectives*: It also differs in Arabic in some significant ways from English. First, there are three ways to designate number, not just the "singular" and "plural" forms. There is also a "dual" form, used for situations where the noun refers to two, and exactly two, things. Such instances occur more often than one might think: all of the parts of the body that come in pairs (such as eyes, hands, ears, feet, legs) are normally dual. Further, unlike English, where most plurals are formed regularly by adding to the word the suffix "(e)s" and only a few plurals are irregular, like "children", "men", "feet" and so on, in Arabic the situation is reversed. Most plurals are formed irregularly (called in Arabic "broken" plurals, because they "break up" the consonant structure of the singular word) and only a few are formed by adding the regular suffixes ات (for inanimate objects and female human beings: عاملات, مكتبات, etc.) or ون (for male human beings: فلّاحون, etc.).
- *Verbs*: Arabic verbs differ from those of English as well, particularly in how their tenses (whether they refer to past, present or future actions) are perceived. In

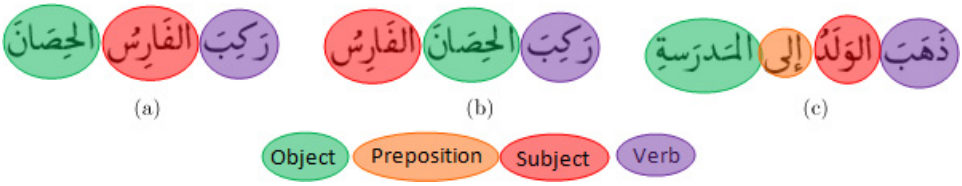


Fig. 1. Examples of special endings to indicate the function of words in sentence.

Arabic, the basic distinction of verb tense is between "completed" (e.g. كَتَبْتُ, I wrote) and "not completed" actions (e.g. أَكْتُبُ, I write). Dependent (the equivalent of the English infinitive (to + <verb>) in expressions like "I want to do that", e.g. "I want to write") and negated verbs (e.g. "I didn't write") for example, are classified as "not completed" actions, even if they describe past events. And, although it is possible to make distinctions in Arabic between present and future, or simple past and past perfect, on the basis of using special words preceding the verb (e.g. أَكْتُبُ: I write (present), سَوْفَ أَكْتُبُ: I will write (futur)), the verb itself will either be conjugated as in the "completed" form (past) or "not completed" form (present or imperfect indicative), and the use of the special "tense markers" are often considered optional, if the writer feels the time reference is clear enough from the context. This lack of congruity between the English and Arabic verb tenses is one of the most difficult obstacles to overcome in producing readable translations of Arabic literary works for English readers.

- *Word decomposition*: Arabic word is either decomposable or not.[5,31] فرنسا (France), دكتور (Doctor) and ثمانية (Eight) are examples of nondecomposable words. Many of nondecomposable words consist of new non-Arabic words that have found their way into the language, particularly terms relating to modern technology. For decomposable words which are derived from a small entity, called root, a set of well-defined rules governs the word decomposition into morphemes: root, prefix, infix and suffix. A root is either tri-consonant (three letters) or quadric consonant (four letters). It can also be healthy or not healthy (contains at least a vowel letter: ي, ى, و, ا). The prefix and the suffix refers to:
  - Conjugation: Time (completed or not completed), Aspect (indicative or subjunctive) and Voice (passive or active),
  - Subject (1st person, 2nd person and 3rd person), Gender (masculine, feminine and neuter), Number (singular, dual and plural),
  - Definition: the noun is definite or indefinite, for example: مدرسة (a school), المدرسة (the school).

The number of schemes can go up to 70 (e.g. مفعول, استفعال, مفاعل, مفعال, فاعل, منفعل, افتعال, etc.). A total of 808 healthy tri-consonant roots can generate a lexicon of 98 000 words.[32] On average, 80 frequently used words can derive from a given root in various schemes.[29] A decomposable word follows a given scheme depending on whether it is a verb (e.g. فعل: كتب, to write), a noun of agent or actor (e.g. كاتب: إسم فاعل, an author), a noun of patient (e.g. مفعول به: مكتوب, written), a noun of machine (e.g. إسم آلة: كتاب, a book), a verbal noun (e.g. مصدر: كتابة, writing), a noun of place (e.g. إسم مكان, مكتبة, a library), a broken plural (e.g. جمع غير سالم: كتب, books). It is worth to note that a root does not fit any scheme. Some coherence rules must be checked before using a scheme for one root. For instance, the similar adjective (صفة مشبّهة) and the superlative (إسم تفضيل) cannot be used with the verb كتب.

3. Arabic Vocabularies

In this section, the focus is on the types of vocabulary. We[9] can classify the Arabic vocabularies into four categories considering (a) decomposable or nondecomposable words, (b) words derived from healthy and unhealthy roots or briefly derivative words, (c) words with and without flexion and (d) words with and without agglutination extensions. This section describes the content of each vocabulary and introduces the handled one.

3.1. *Vocabulary of nondecomposable words*

Nondecomposable words are not generated from roots and cannot be derived according to a given scheme. Some of them can be flexed or agglutinated; others do not have any morphological characteristic (neither derivation, nor flexion nor agglutination). For examples:

- Primitive names: دكتور (a Doctor), دكتورتهما, دكتورة, etc.
- Proper names: محمّد, علي, etc.
- Names of adverbial value: أَبدًا (never), كيف (how), قليل (little), كثير (lot), etc.
- Personal nominative pronouns: نحن (we), etc.
- Personal accusative pronouns: إيّاها, إيّاك, etc.
- Relative pronouns: الّذي, الّتي, (which), etc.
- Demonstrative pronouns: هذا, تلك, etc.

3.2. *Vocabulary of decomposable words*

Most of the Arabic words are morphologically derived from a list of roots. Recall that the root is the simple form of the verb and can be tri- or quadric-consonantal. Most of the roots are built up from three consonants. Roots are interfered with schemes to form the Arabic words. The model (or scheme) can be considered as a pattern which is well governed by well-determined rules. These models produce nouns and verbs.[4]

3.2.1. *Vocabulary of brief decomposable words*

Brief decomposable words are flectional but not agglutinated words. We distinguish between canonical and noncanonical vocabularies for brief decomposable words. The canonical vocabulary of brief decomposable words involves words that are derived from decomposable healthy roots. That is to say that the three letters of scheme to be replaced will be effectively and automatically substituted by those of the root. Figure 2 shows an example of word derivation from a healthy root where scheme letters are colored in green.

The noncanonical vocabulary of brief decomposable words involves words derived from unhealthy roots. In such case, not all root letters will be reproduced as they are. Letter vowels of the root can be converted to other letters according to possible

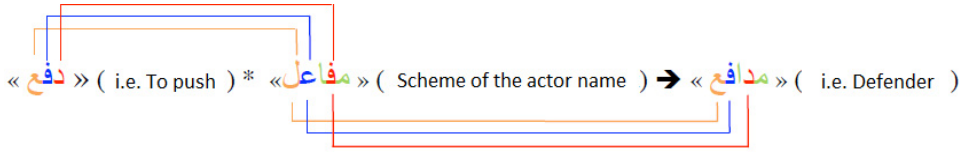


Fig. 2. Example of word derivation from the healthy root دفع (color online).

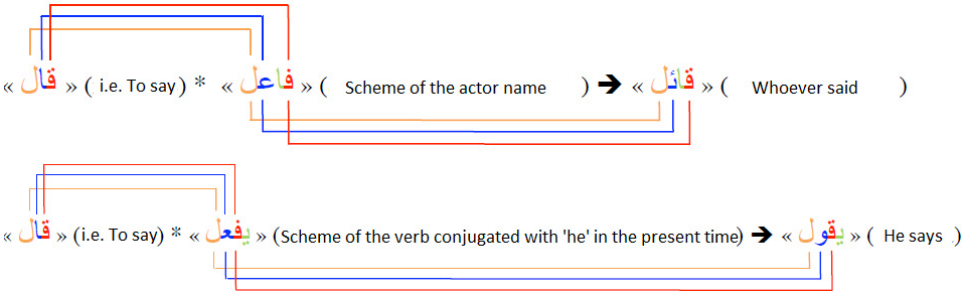


Fig. 3. Examples of word derivation from the unhealthy root قال (color online).

derivations. An example of vowel letter is ا, the second blue letter in Figs. 3(a) and 3(b). This vowel letter of the root قال changes in the words قائل and يقول as illustrated in Fig. 3.

3.2.2. *Vocabulary of extended words (surface forms)*

This vocabulary includes brief words, decomposable or not, that are agglutinated. The word مدافعها is an example of extended decomposable word. دكتورهما is an example of an extended nondecomposable word.

3.3. *Handled vocabulary*

Considering the large number of words and taking into account the diversity of vocabularies, we decide to only consider the vocabulary of decomposable words derived from healthy tri-consonantal roots without agglutination. As underlined by Ref. 16, the majority of Arabic words are decomposable words derived from roots and most Arab roots are trilateral. In fact, the number of decomposable words greatly exceeds hundreds or thousands of words which is considered as a large vocabulary in taxonomy given by Ref. 17. Hence, even by restricting to tri-consonantal roots, we target a large vocabulary since with only 808 roots, a vocabulary of 98 000 words can be derived as mentioned by Ref. 34. Also note that, the target vocabulary is dynamic in the sense that our system is able to recognize words that samples were not learned before i.e. words that do not belong to the vocabulary and we consider intruders as it will be explained in subsection: resolution of corpus explosion.

4. Related Works

To recognize nondecomposable words, global or holistic approaches were proposed. They treat words as single, indivisible entities, and attempt to recognize them as whole, bypassing the segmentation stage. They permit the word segmentation stage to be avoided by extracting global features from the words and exploring the information from word context, therefore not needing their explicit segmentation. These approaches are restricted to applications with small vocabulary recognition, such as reading the courtesy amount on bank checks,[6,22,52] because its complexity grows linearly with the number of word models. Hence, the performance of system based on global approach would surely deteriorate for large vocabulary systems, and also for morphologically similar written words.

For large vocabulary systems, other approaches for word (decomposable or not) recognition have been proposed that can be classified into dependent or free segmentation approaches. In dependent segmentation approaches, also called analytic approaches, the objective is to focus on letters or smaller entities for their interpretation. Systems are then faced with the necessity of word segmentation into characters/letters or pseudo-letters,[21,43] etc. These approaches are known for the difficulty they show when defining a boundary between the characters. Therefore, the recognition system will also depend on the success of the segmentation process. In this manner, the system will be made up of distinct stages, one for segmentation and one for recognition, or even associating segmentation and recognition in a unique stage. Free segmentation approaches are based on modeling words in the form of feature vectors, extracted from overlapping windows[24,25,36,37,48] or in the form of spatial distribution of symbol models.[1] To recognize a word, the system proposed by Ref. 1 does not commit itself to a segmentation of the word, rather it simulates trying different segmentation points and then chooses the best set of segmentation points that provides the best recognition. This system optimizes the segmentation with respect to the whole word. But the price paid for the optimization is having to use a time-consuming search. In Ref. 18, authors presented a segmentation-free system for online Arabic handwriting recognition. They proposed a holistic approach that performs online Arabic handwritten word recognition on a continuous word-part level, while performing training on the letter level. Such a scheme avoids the segmentation of words into individual letters during the recognition process, which is often prone to errors, and substitutes the training for large set (the word-parts) by a small set (the letters). The recognition framework uses discrete HMM to represent each letter shape. These letter shape models are embedded in a network that represents a word-part dictionary. The recognition of word-parts is performed without explicit segmentation into letter shapes and instead, the recognition is performed along paths that represent valid word. Their approach greatly utilizes the fact that Arabic words are composed of word-parts to improve the efficiency of the recognition framework.

As Arabic words are built by concatenation of several independent written parts, called Parts of Arabic Words (PAW), which give another natural segmentation level,

pseudo-global approaches are also proposed particularly for the recognition of Arabic words. As mentioned in Ref. 11, this natural segmentation refine the analysis by reducing the basic vocabulary. It is why some approaches have based their work on this level,[10,12] etc.

For decomposable Arabic word vocabulary recognition, approaches based on exploiting the morphological structures of the Arabic language were recently proposed. In Ref. 34, an "affixal" approach tries to recognize word basic morphemes: prefix, infix, suffix and root contrary to existing approaches which are usually based on recognition of word entity by global approach, PAW entity by pseudo-analytical approach or letter entity by analytical approach.

5. Arabic Linguistic Knowledge Integration

Works, reported in Refs. 16, 19, 32 and 33 highlight the richness and the stability of Arabic in terms of morphological concepts peculiar to this language. As previously underlined, an Arabic decomposable word derives from a root and can be decomposable in morphemes (root, prefix, infix and suffix). A word is, then, composed of root letters and access (nonroot) letters. In Ref. 19, the author proposed to exploit this word vision using any recognition approach. He affirmed that one must focus on what kind of linguistic knowledge is important and how and where it is more appropriate to incorporate.

Kanoun[33] proposed a system for the recognition of printed texts based on an affixed approach. He proposed an "affixal" approach which consists in segmenting words into letters and recognizing their morphological entities. These morphemes are validated using dictionaries of roots, infixes, prefixes and suffixes. They used many linguistic concepts (affixed and semantic restrictions) to guide the recognition. This approach has been then reconsidered by Ref. 32 for the recognition of Arabic typed texts, consists in segmenting words in letters and recognizing their morphological entities.

Our vision of the word is slightly different. We consider a decomposable word as being the derivation of its root according to a conjugated scheme. This latter is the association of prefix and suffix (letters from the conjugation: time, kind, number, etc.) to a brief scheme. For example, the Arabic word متاجرون (traders) is built up from the root تجر. The derivation of the root تجر with the brief scheme مفاعل (i.e. ?? ا?م where ? will be replaced by a root letter) gives rise to the radical متاجر. Prefix and suffix are composed of access letters corresponding to the conjugation, while other access letters belong to the radical and depend on the scheme (see Fig. 4). Thus, to be able to recognize a word, we just need to identify its root and conjugated scheme, without segmenting it in letters and recognizing them. The main idea is to factorize words and so to handle with a large vocabulary while using a global approach to avoid effective segmentations.

Although analytic approach is the preferred choice for applications where large vocabularies are required, we are interested by a global approach for three main reasons:

1. It avoids segmentation especially as cursive word segmentation has been acknowledged as the most difficult of all handwriting segmentation problems,[39]

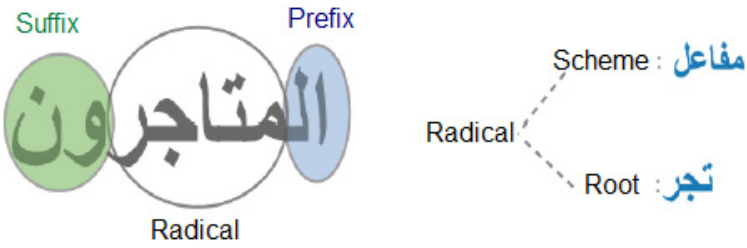


Fig. 4. Our word vision for the Arabic word (e.g. المتاجرون).

2. It is the one that corresponds to the conventional reading task performed by humans. As explained by Ref. 42, the concept of "word superiority effect", concept implies that human proceeds by a general study of the problem. When global vision is not clear enough, the need of local vision, to observe details, seems to be obvious,
3. Considering the reading process itself and the perception of writing, Arabic reading seems to be more global than syllabic. It is facilitated by chopping the word in PAWs which makes it finally semi-global.[11]

6. Proposed System

Our proposed approach tries to benefit from Arabic morphology stability by integrating linguistic knowledge to a perceptive model. Figure 5 summarizes the developed system architecture. The system begins with an unknown word (test pattern) presented at the input of the recognition system as an image. To recognize this word, solutions to a number of challenging problems are needed. The first stage is to extract discriminant features (here global structural primitives) from the input pattern (either test or reference pattern) to generate a sequence of codes used to describe the word. The pattern training stage consists of root, scheme and conjugation element training. In spite of the goal of most recognition systems being to recognize words, it is difficult, when handling with large vocabulary, to associate one class to each word, so then subword models (here root, scheme and conjugation models) are trained instead, and a word restitution step is used to build up word models during the recognition stage.

As shown in Fig. 5, we used a hybrid approach (structural global primitives to describe words and local refinement by perceptive cycles in case of ambiguities at the recognition stage) and transparent neuronal network (TNN_R, TNN_S and TNN_C to respectively train and recognize roots, schemes and conjugation elements) as classifiers.

HMM is an analytical method based on local perception but as our objective is to simulate the whole process of human reading and avoid the problem of segmentation, we follow an AI vision of the recognition and propose an adaptation of the McClelland and Rumelhart model. While presenting the same advantages than traditional ANN, TNNs has the advantage to be transparent that is neurons are no longer black boxes since a concept is associated to each neuron. Such network provides fast collision analysis and result correction.

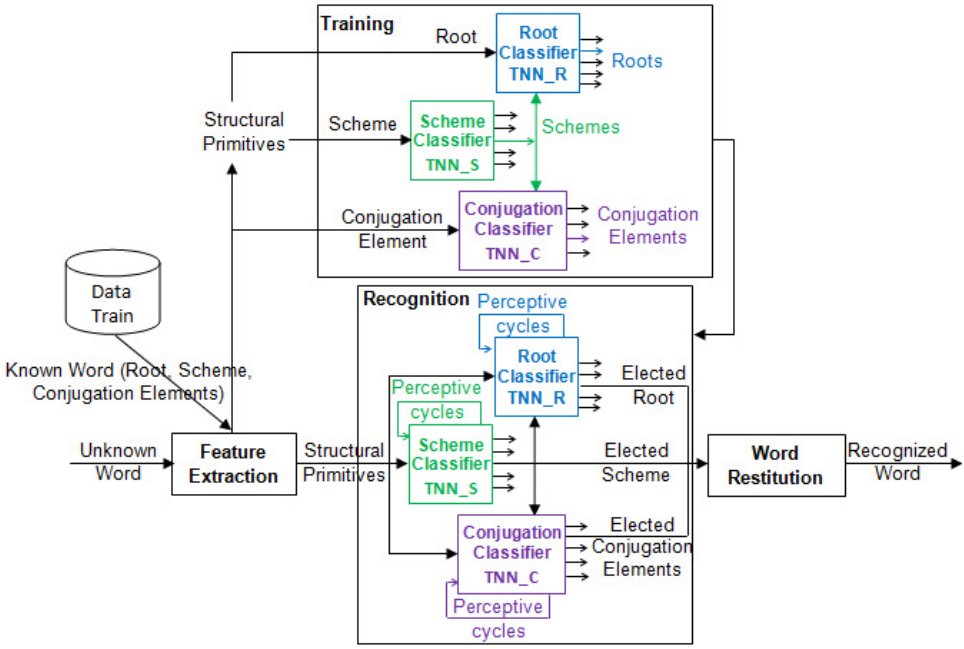


Fig. 5. Overview of the basic modules of the proposed system.

TNN is chosen not only for their great learning capacities, as any artificial neuron network, but also for their transparency. In TNN, neurons are no longer black boxes since a concept is associated to each neuron. Such network provides fast collision analysis and result correction. Notice that we have already experiment, in previous work, the use of TNN on a reduced vocabulary[1] and experimental results show that TNN performs considerably well. We believe that the benefit of using this model (compared to HMM which is an analytical method based on local perception) for the recognition of Arabic writing lies in the following aspects:

1. The training and recognition steps via this neuronal model aligns well with the reading process in humans: global vision, followed by a local vision in cases of ambiguity, as assumpted by Ref. 42 and shown in Fig. 6.
2. This type of model supports the use of as many layers as level decomposition: feature, letter and word as also shown in Fig. 6.
3. As cursive nature of the Arabic script makes the segmentation task delicate, so with this model whose inputs are global primitive, we just need to extract the most relevant structural primitives to recognize the entire word.

In next subsections, we will describe the feature extraction, training and recognition steps and demonstrate the interest of the collaborative combination between the three classifiers.

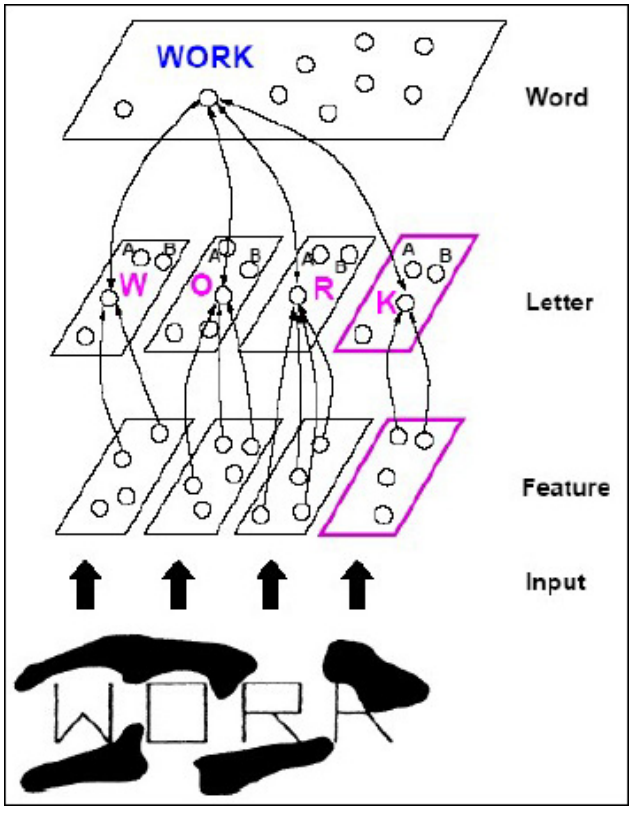


Fig. 6. McClelland and Rumelhart model.

6.1. *Feature extraction*

In this subsection, we define selected features and explain how to extract them.

6.1.1. *Selected features*

The selected features must be reliable, independent, and small in number and reduce redundancy in the word image. We consider a feature as an ascender (A), a descender (D), a loop (L), diacritic (Da and Db for respectively diacritic above and below) or nothing (O), in view its position in a PAW: in the beginning (B), in the middle (M), at the end (E) or isolated (I). This association: feature-position is made to be more precise when describing features. Figure 7 gives an illustrative example of extracted features from the word يستعرضان.

Let us consider a primitive as a feature combination. For example, the primitive LB means a loop in the beginning like in letter م, the primitive $DDaI$ signifies an isolated descender with diacritic above like in ش and the primitive ALI refers to an isolated ascender with a loop like in the letter ط (see Table 1). In Fig. 8, the word

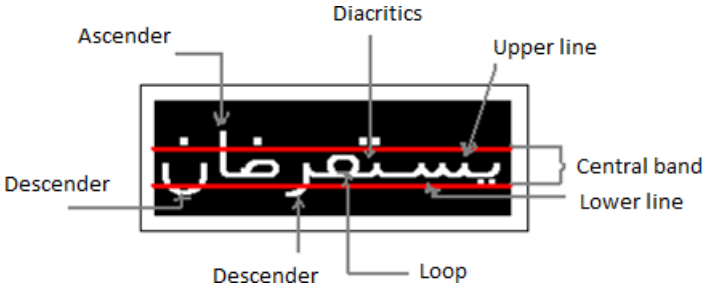


Fig. 7. Word global features extracted from the word يستعرضان.

Table 1. Primitives and examples of associated letters.

Primitive	Example	Primitive	Example	Primitive	Example
AB	ل	DbI	ب	DDbE	ج
AI	ا	OB	ع, حـ	LDaB	ق, فـ
AE	ا	OI	د	LDaI	ف
DI	ر, ح, ع	ADI	ل	DDaE	خ
DE	ى	ABI	ط	LDaE	ة
LB	ص, م	ADaI	آ	DDbI	ي
LI	ه	ADbI	ا	LDaM	ض
LM	ھ, هـ	DDaI	خ	LDaE	ف
LE	ه	DDaI	ش	ALDaM	ظ
DaB	خ, ز, ت, ث	DLE	و	DLDaI	ق
DaI	ذ	DLI	ص	DLDaE	غ
DbB	ي, ب	DDbI	ج	ALDaI	ظ

متاجرون is described, from right to left, by a list of structural primitives considering their positions according to the PAWs.

6.1.2. *Extraction steps*

The feature extraction process is tightly related to the adopted segmentation approach. Segmentation is a well-known problem in word recognition due to its high variability especially when dealing with a large vocabulary for semi-cursive scripts as Arabic. In order to build a feature vector sequence to describe each word, we use implicit word segmentation where the image is divided from right to left into many vertical frames. To deal with the morphology complexity of Arabic handwritten letters, we have adopted a nonuniform segmentation. Frames do not necessarily have the same width and the boundaries of each frame are based on minima and maxima

Fig. 8. Structural primitives of the word متاجرون.

Fig. 9. Central band computing.

analysis of the vertical projection histogram. We explain here the followed steps for feature extraction.

1. Determining the central band (base line, upper line and lower line), as illustrated in Fig. 9, by:

   (a) Horizontal projection of the image after discarding diacritic.
   (b) Determination of the peak and both significant local minima.

2. Decomposing the word into: PAWs, diacritics and loops.
3. Delineating areas of each PAW (see Fig. 10) and saving their borders by:

   (a) Computing PAW skeleton,
   (b) Vertical projection of the skeleton,
   (c) Locating the climbs and the flats of the projection and
   (d) Fusion of areas of the same letter, especially at the end of the word.

   Note that vertical projection can prove not effective in the event of short words, in the presence of vertical ligatures and overlapping but decomposable words result from tri-consonantal and quadric-consonantal roots; therefore they are made, at least, of three letters in addition to the affixed letters (prefixes, infixes and suffixes) which can be added to the word by the effects of derivation, conjugation and agglutination. In case of vertical ligature, as shown in the example of Fig. 11(a),

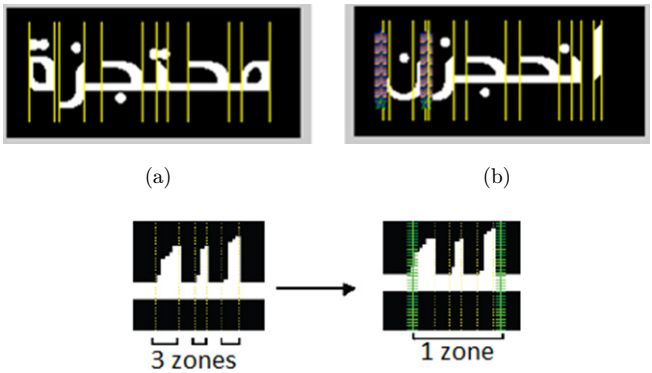


Fig. 10. Arabic word areas.

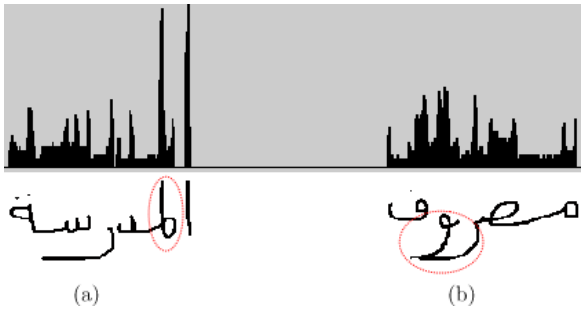


Fig. 11. Problems of ligature and overlapping in case of vertical projection.

the letter ل is legated to the letter ه in the word المدرسة which can be confused with the letter ط. We do not seek to separate between the legated letters. When training word roots, schemes and conjugation elements from different samples of words, we only give the extracted primitives as input to the TNNs and by supervised training, we each time specify which desired neuron will be activated. In case of overlapping between successive letters, for example the letters ر and و in the word مصروف (see Fig. 11(b)), we build on the fact that a PAW does not contain more than one descender what generally allows to separate them in final and isolated descenders. It is important to know that global approach makes it possible to recognize the overall word even if one of its letters is badly represented what accredit the word superiority principle of McClelland and Rumelhart (the word is considered as a whole not as a sum of small parts).

As illustrated in Fig. 12, to fusion areas of the same letter placed at the end of the word, for example the ل in the word رسول (see Fig. 12(a)), the significant climb which appears at the end of the histogram (see Fig. 12(c)), will not be extracted as new primitive, rather it will be absorbed by the previously extracted primitive (ADI: Ascender Descender Isolated which refers to the letter ل) thanks to the presence of descender at the end of the word. When a slight climb is observed in the histogram, for letters at the end of PAWs (see example of the letter ب at the

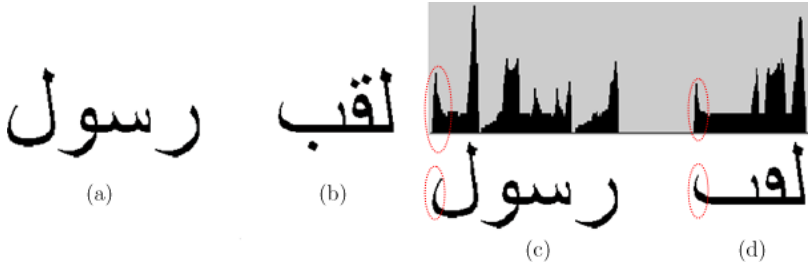
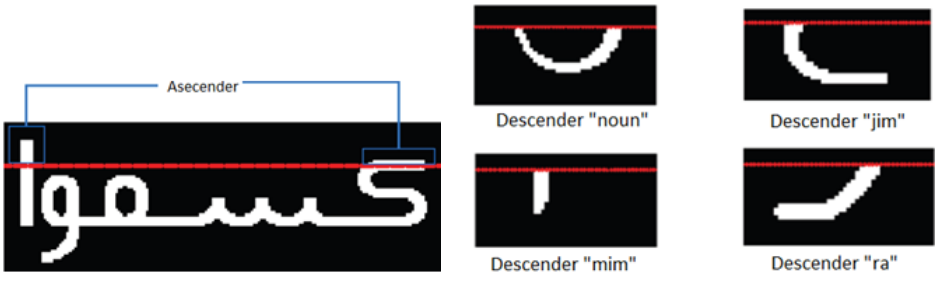


Fig. 12. (a) Example of word with descending letter at the end, (b) example of word with letter at the end of PAW and (c) and (d) vertical projection of words given as examples.



(a) (b)

Fig. 13. Ascender and descender extraction (a) Ascender extraction and (b) different types of descenders.

end of the PAW لقب in Fig. 12(b)), it will be also referred to the previous primitive (DbE: Diacrtic below at the End which refers to the letter ب).

4. Finding ascenders and descenders (see Fig. 13(a)): Since ascenders are placed above the central band, a vertical projection is applied on the PAW superior part to locate ascender boundaries. Note a PAW may contain many ascenders. Descenders are found below the central band. A PAW can only have one descender at its end. Descenders are of different types: descender "noun" (*Dn*), "mim" (*Dm*), "jim" (*Dj*) and "ra" (*Dr*), as shown in Fig. 13(b). Their extraction is made by specifying their types. A descender is of type *Dn* if it has two intersection points with the baseline. Otherwise, it can be *Dm*, *Dr* or *Dj*. To distinguish them, their intersection point position is considered: *Dj* is selected if the intersection point is placed on left side and *Dr* is selected if the point is rather placed on the right side, otherwise, it is a *Dm*.
5. Generating the sequence of primitives by superposing areas with loops, diacritics, ascenders and descenders as shown in Fig. 14.

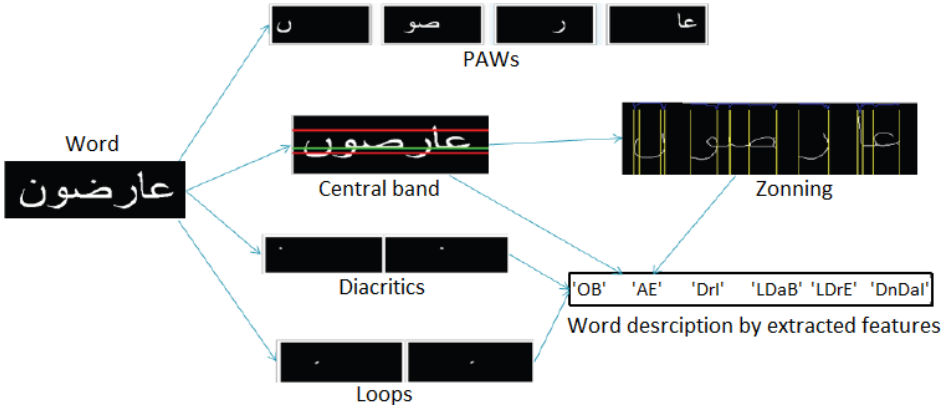


Fig. 14. Word description.

Experiments, carried on 17 271 word samples, written in Arial, AGA Abasan Regular and AF_Buryidah fonts with different sizes, achieve 97% of extraction success.

6.2. *Training and recognition*

Being inspired by the perceptive model of Ref. 42, as previously shown in Fig. 6, all of the used networks are composed of three layers: primitives, letters and words. But each network is specialized in a specific morphological concept (root, scheme and conjugation).

6.2.1. *Training*

For each network, we apply a supervised learning: each sample from the training data, is a pair consisting of an input pattern and a desired output value. The supervised learning algorithm (here back-propagation algorithm) analyzes the training data and produces a classifier which should predict the correct output value for any valid input pattern. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

Unlike traditional recognition system where the training of the word is made via different samples of that word, TNN_R tries to learn a root from different words derived from that same root, and not from different samples of that root. TNN_R learns (1) how to ignore access letters and (2) to only consider root letters of the word (see Fig. 15).

TNN_R is composed of three layers (primitives, letters and roots) which respectively contain 50, 104 and 101 neurons (see Fig. 16).

Opposite to TNN_R, TNN_S learns how to ignore the root letters and focus on access schemes letters to find out the scheme that the word follows (see Fig. 17).

TNN_S has three layers which correspond to word primitives, letters and schemes. They respectively contain 50, 28 and 22 neurons. 28 is the number of determinative letters of Arabic used schemes and 22 is the number of schemes. Figure 18 displays TNN_S architecture.

TNN_C focuses on access conjugation letters to learn the conjugation elements (number, gender, time, etc.) as illustrated in Fig. 19.

TNN_C has three layers (primitives, letters and conjugation elements) which respectively contain 50, 30 and 12 neurons. 30 is the number of letters that can be

Fig. 15. Training of the same root صرف from words with different schemes and conjugations.

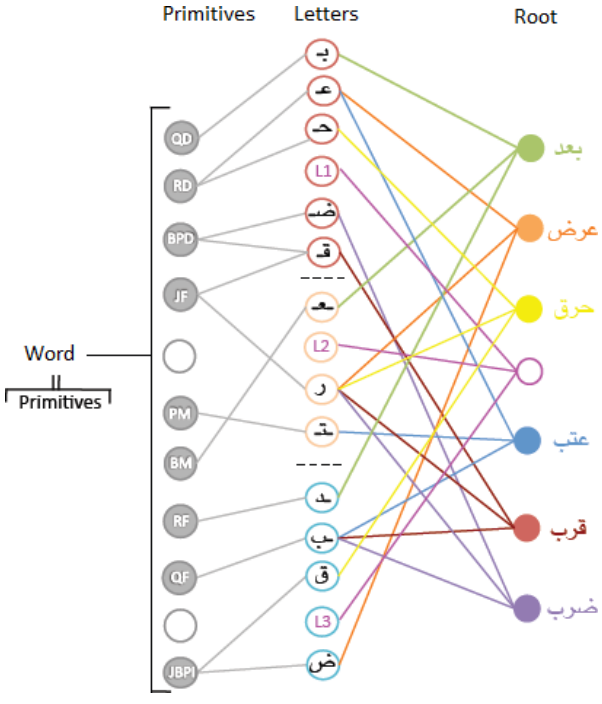


Fig. 16. TNN_R architecture.

used as prefixes or suffixes at verb conjugation (for example: the prefix أ is added to the radical when the verb is conjugated with I at the present time and the suffix ون is added to the radical when the name is conjugated in the plural masculine). 12 is the number of conjugation elements such as the number (singular, dual and plural), gender (male, female), time, etc. Figure 20 gives an example of conjugation elements training from the word اختلفن.

Experiments, carried on the three TNNs separately trained, showed that TNN_S usually provides the highest scores. That is why it should be beneficial to rely on this network to help supervising the two others. So, (1) we provide word structural primitives to each TNN, (2) we give, to TNN_R the word root, to TNN_S the word scheme and to TNN_C the word conjugation elements, so that each one adjusts its training parameters according to the desired output, and (3) we allow the word

Fig. 17. Training of the same scheme from words with different roots and conjugations.

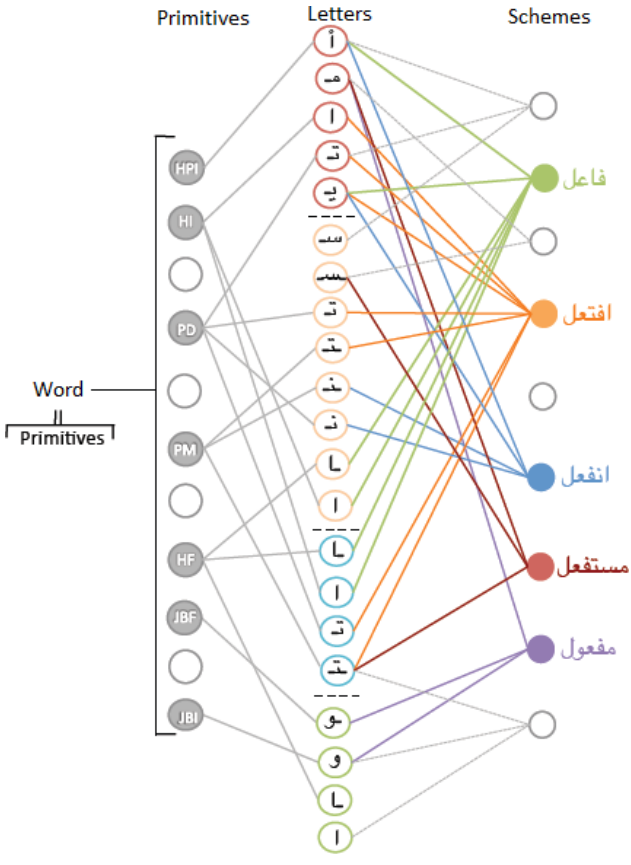


Fig. 18. TNN_S architecture.

scheme to supervise root and conjugation classifications (as shown in the training step of the proposed system overview, Fig. 5).

6.3. *Recognition*

To recognize a word, the next steps are followed: (1) the scheme classifier is supplied with the word primitives, (2) perceptive cycles are applied on this classifier to discard bad candidates, (3) the same word primitives are supplied to the root and

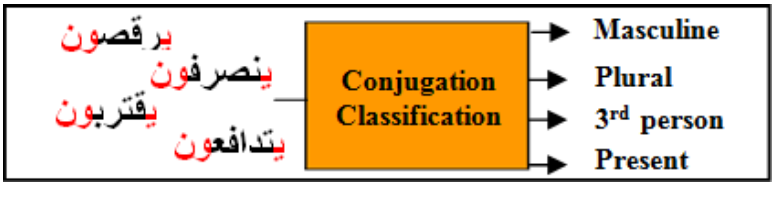


Fig. 19. Same conjugation training from words with different roots and schemes.

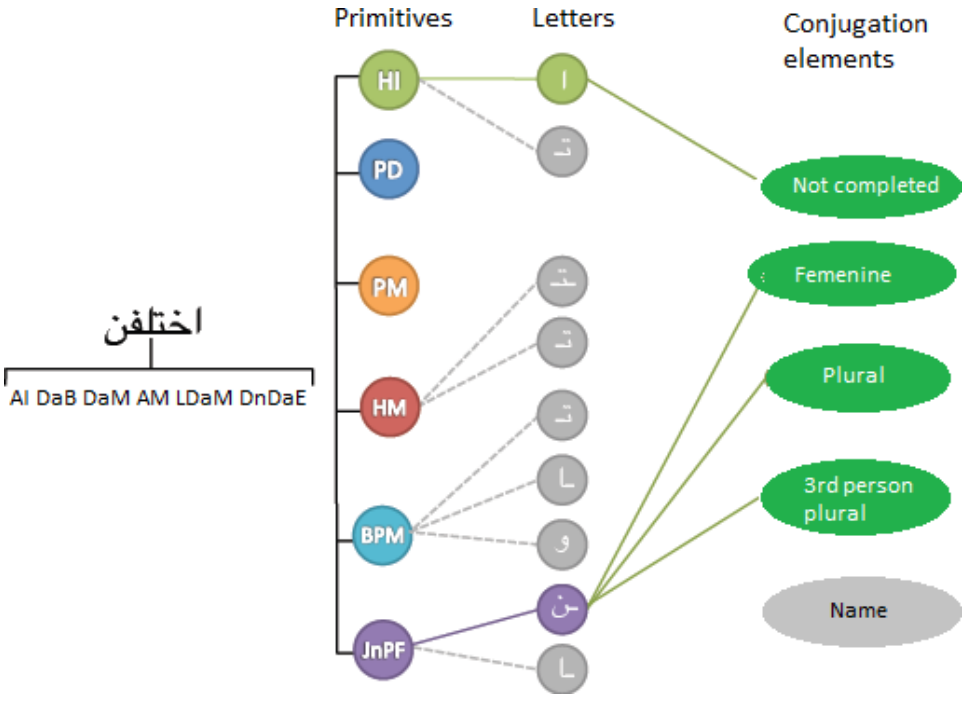


Fig. 20. TNN_C architecture, example of conjugation elements training from the word اختلفن.

conjugation classifiers which will be supervised using the maintained scheme candidate, (4) perceptive cycles, applied separately on root and conjugation classifiers, will refine the vision to reject bad candidates, (5) linguistic constraints will be used to reject other confusions, by confronting root and scheme classifiers outputs, since a root does not fit any scheme and vice versa and (6) the word will be reconstituted from the root, the scheme and the conjugation selected candidates (see Fig. 21).

To better explain how perceptive cycles benefit from network transparency to decide about the correct output, we give an example, in Fig. 22, of applying perceptive cycles on TNN_R. It shows that a first propagation (see (1)) of a word structural primitives leads to two concurrent roots سخن and عتب. Since each root is characterized by three determinative letters, local feedback could lead to the right root when deciding about its determinant letters. Indeed, a first perceptive cycle seeks for some details concerning the first letter of the root (see (2)), in order to decide which letter between س and ع has to be retained. For that, the system goes back to word to check if the vertical projection of the corresponding zone is flat or presents three acute peaks. In this case, the shape of the projection is flat, thus the س letter neuron, in the second layer, will be pruned. By the same way, a second perceptive cycle (see (3)) checks whether the second letter is ت or خ and will lead to the pruning of خ letter neuron (since the shape of the letter, to be identified, is acute).

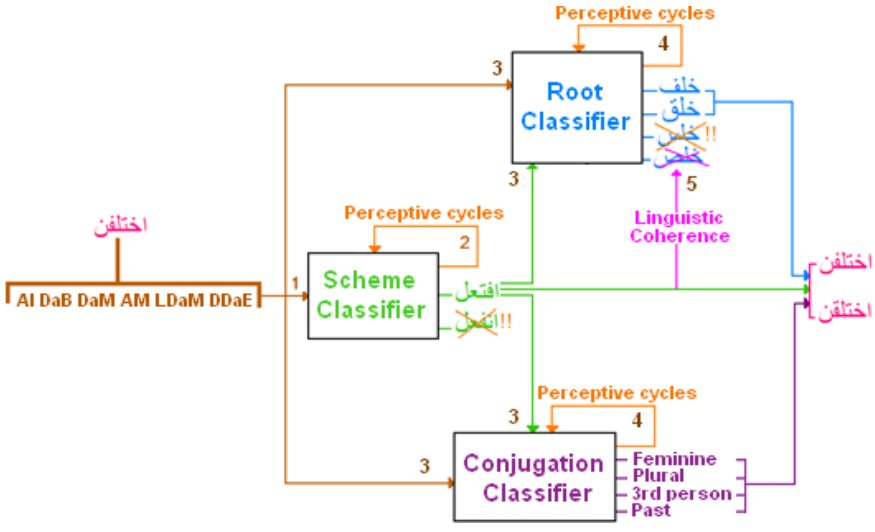


Fig. 21. Recognition process with classifier collaboration and perceptive cycles.

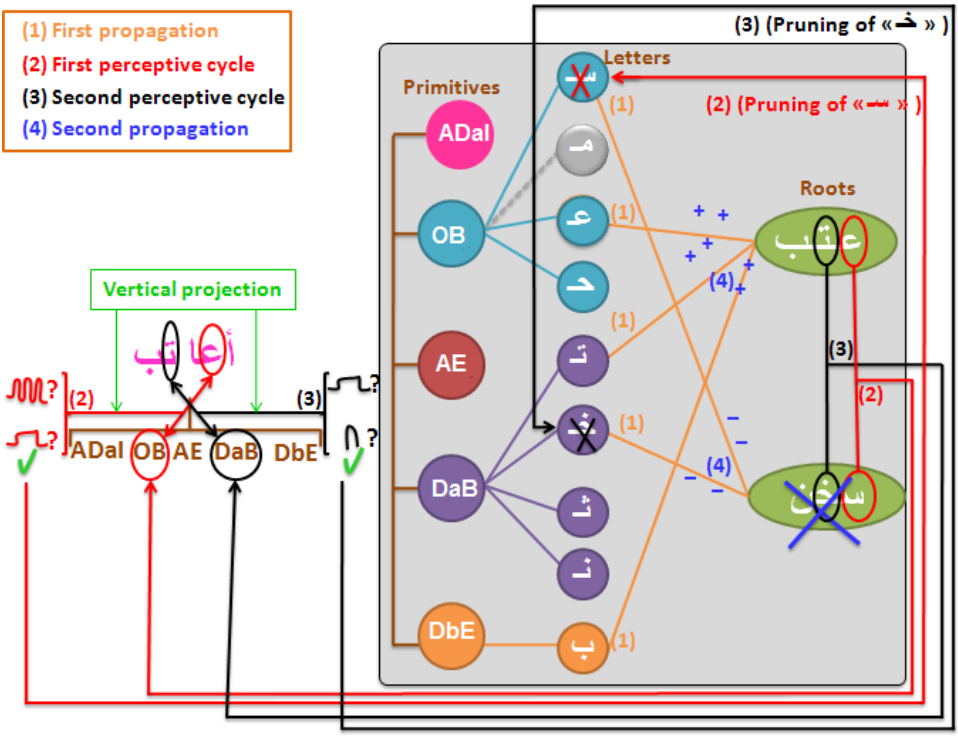


Fig. 22. Example of perceptive cycles on TNN_R.

Finally, a second forward propagation (see (4)), from the second layer, will exclude the root سخن and enhance the vote to عتب.

6.4. *Word restitution*

To restore the radical from the proposed roots and schemes, we calculate, for each word, the maximum of products of the majority likelihoods (*a priori*) of the scheme with the *a posteriori* probability of the root knowing that scheme as follows (R: root, S: scheme and Radical: derivation of R according to S).

$$P(\text{Radical}) = P(S) * P(R/S). \tag{1}$$

Figure 23 shows an example of word recognition by combining the outputs of the three networks. Note that the combination has rejected the scheme فاعل, the first candidate TNN_S (although its activation is equal to 0.8), and retained the scheme فعال, the second candidate (activated to 0.73). In fact, if TNN_R has been supervised by the scheme فاعل, it proposed the root عدل with accuracy of 0.61 (P(عدل/فاعل)). But, if it has been supervised by the scheme فعال, it proposed the root بعد with accuracy of 0.92 (P(بعد/فعل)). By multiplying the probabilities and comparing: $(0.8 * 0.61 = 0.48 < 0.67 = 0.73 * 0.92)$, the word بعادين (combined to 0.67) takes over the word يعادلن (combined to 0.48).

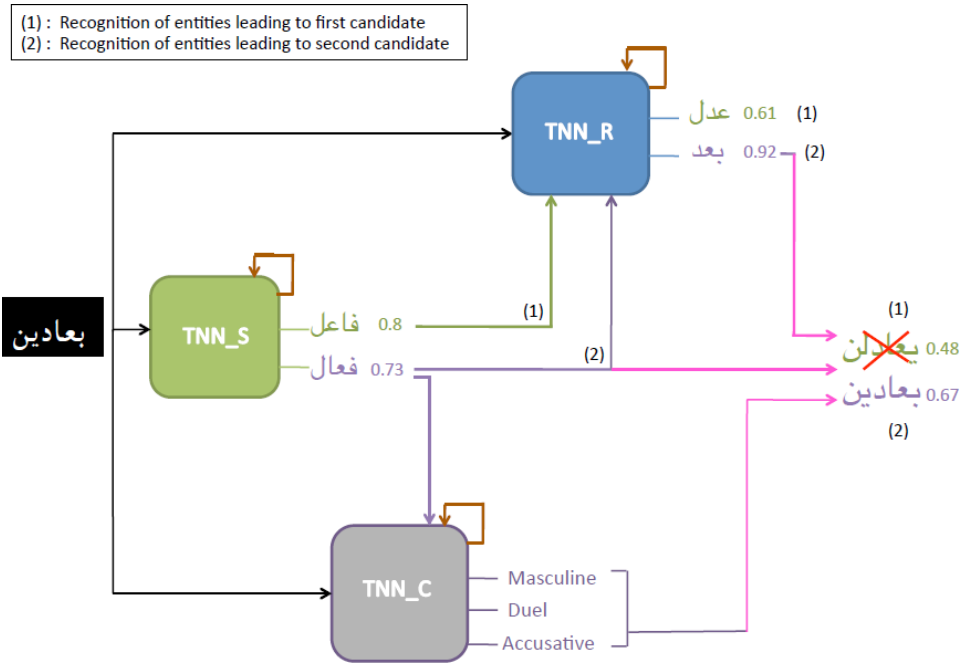


Fig. 23. Example of combination for the recognition of the word بعادين.

7. Results and Discussion

To our knowledge, also confirmed by Refs. 7 and 35, no database for Arabic printed decomposable words is freely available for researchers. Hence, different researchers of Arabic open vocabulary and text recognition use different data and the recognition of the different techniques may not be comparable. For this reason, we build our database of decomposable words which are derived from only healthy tri-consonantal roots. Taking as input the words in the lexicon, the images of our database are synthetic, generated using three different fonts: Arial, AGA Abasan Regular and AF_Buryidah written in different sizes (three sizes: 16, 17 and 18) as shown in Fig. 24. One can clearly observe variations in height of letters; in distance between letters, in height and shape of ascender letters (e.g. the letter گ) and in depth and shape of descender letters.

We carried experiments on a vocabulary of 2000 then 5757 respectively generated from 51 and 101 roots. Words are derived using 22 schemes taking into account various conjugation elements (e.g. masculine, female, singular, duel, plural, present, past, etc.).

7.1. *Network stability test on a vocabulary of 1700*

Figures 25(a) and 25(b) respectively present the obtained recognition rates of TNN_S and TNN_R using different sizes of vocabularies. As one can see, both networks behave in a stable manner with growth in size of the vocabulary. This stability is explained by the fact that enlarging vocabulary, the number of neurons in the output layers of TNN_R and TNN_S remains fixed (51 and 22 for TNN_R and TNN_S, respectively) and that new words contribute to learn more and more neurons of these TNNs.

7.2. *Experiments on vocabulary of size 2000*

We have expanded the lexicon by generating more words from 51 roots to achieve 2000 different words with various schemes and inflections. We have built a database

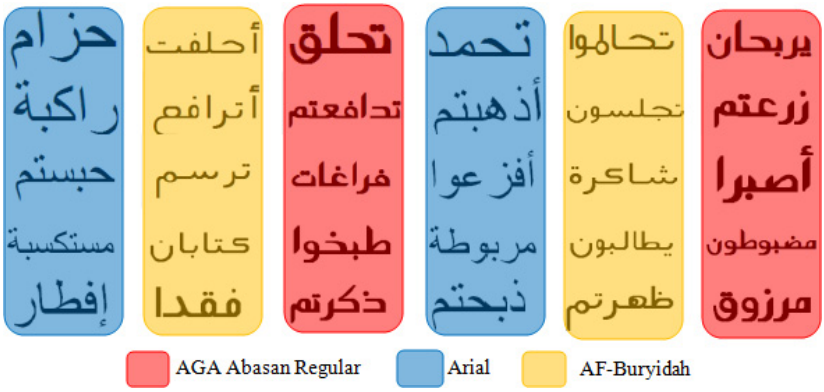
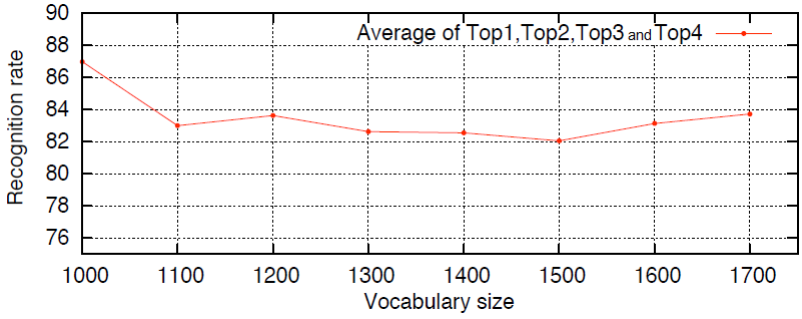
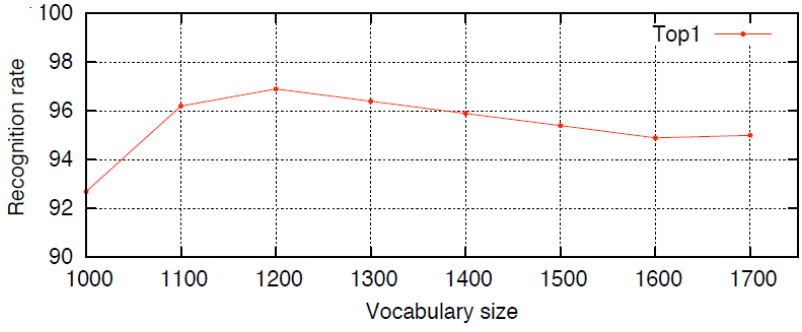


Fig. 24. Examples of handled words classified by fonts.



(a)



(b)

Fig. 25. Stability test of (a) TNN_R and (b) TNN_S.

of 6000 samples. Around 4000 samples were used for training and 2000 for recognition. In Table 2, we mention, for each network, the exact number of training and test samples used by root, scheme or conjugation elements. Note that 78, 181 and 333 samples of words have been served to respectively learn a root, a scheme and a conjugation element.

These proportions were used in the following experiments to compute the recognition rate for each network on a vocabulary of size 2000 but in different situations:

1. before perceptual cycles,
2. after perceptual cycles on TNN_S,
3. after cycles of perceptual on TNN_S and TNN_R and
4. after cycles of perceptual on TNN_S, TNN_R and TNN_C.

Table 2. Training and test corpus sizes for each network.

	Number of Trained Outputs	Training Corpus Size	Test Corpus Size
TNN_R	51 roots	$51 * 78 = 3978 \cong 4000$	$51 * 39 = 1989 \cong 2000$
TNN_S	22 schemes	$22 * 181 = 3982 \cong 4000$	$22 * 90 = 1980 \cong 2000$
TNN_C	12 conjugation elements	$12 * 333 = 3996 \cong 4000$	$12 * 166 = 1992 \cong 2000$

Table 3. Network recognition rate, without perceptive cycles, for a vocabulary of size 2000.

	TNN_R (%)	TNN_S (%)	TNN_C (%)
Top1	90.05	94.71	96.32

Table 4. Network recognition rate with perceptive cycles only on TNN_S, for a vocabulary of size 2000.

	TNN_R (%)	TNN_S (%)	TNN_C (%)
Top1	93.33	98.48	98.57
Gain	3.25	3.77	2.25

Notice that the quality of recognition results is tightly related to quality of the feature extraction. Recall that experiments, carried on 17 271 word samples, achieved 97% of extraction success. Table 3 shows the recognition rate of TNN_R, TNN_S and TNN_C for a vocabulary of size 2000.

By applying perceptive cycles only on TNN_S, the recognition rates have been significantly improved as illustrated in Table 4. This is explained by the fact that improving TNN_S can only enhance TNN_R and TNN_C since these latter heavily depend on results of TNN_S which plays a leading role, as aforementioned.

Table 5 presents obtained results after applying perceptive cycles on the three networks.

In Table 6, we summarize the recognition rate before and after cycles in the case of collaboration.

Table 7 highlights the gain provided by the collaboration, by giving the system scores before and after collaboration.

Table 5. Network recognition rate, with perceptive cycles on the three TNN, for a vocabulary of size 2000.

	TNN_R (%)	TNN_S (%)	TNN_C (%)
Top1	94	98.48	98.68
Gain	3.95	3.77	2.36

Table 6. Network recognition rates, with and without collaboration, with and without perceptive cycles for a vocabulary of size 2000.

	TNN_R (%)	TNN_S (%)	TNN_C (%)
Without collaboration	80.6	94.71	—
With collaboration, without perceptive cycles	90.05	94.71	96.32
With collaboration, with perceptive cycles	94.0	98.48	98.68

Table 7. Gains from network collaboration and perceptive cycles for a vocabulary of size 2000.

	TNN_R (%)	TNN_S (%)	TNN_C (%)
Gain from network collaboration	9.45	—	>0
Gain from perceptive cycles	3.95	3.77	2.36
Gain from collaboration and perceptive cycles	13.3	3.77	>2.36

Table 8. Training and test corpus sizes for each network.

	Number of Trained Outputs	Training Corpus Size	Test Corpus Size
TNN_R	101 roots	$114 * 101 = 11\,514$	$57 * 101 = 5757$
TNN_S	22 schemes	$523 * 22 = 11\,506$	$261 * 22 = 5742$
TNN_C	12 conjugation elements	$959 * 12 = 11\,508$	$479 * 12 = 5748$

7.3. *Experiments on vocabulary of size 5757*

Once again, we have expanded the vocabulary a bit more, generating more words, this time, from 101 roots, to have 5757 different words in different schemes and inflections. We used the same fonts (Arial, Regular and AF_Buryidah Abasan AGA) with different polices leading to a database of 17 271 samples. A total of 11 514 samples have been used for the training and 5757 for the recognition. Table 8 gives, for each network, more details about distribution of samples between training and recognition.

Table 9 displays, for each TNN, the training duration, the iteration number and the error. As indicated by final TNN errors, the three TNNs converge since they yield to errors inferior than 575.7 which is the product of sample number (11 514) and the threshold ($= 0.05$).

As displayed in Table 9, the training can take hours depending on the number of roots but once completed, the recognition is instantaneous since it is a simple forward propagation of the three TNNs. Note that the integration of linguistic knowledge is made during the training step which also explains this duration. Systems that use linguistic knowledge as post processing, require additional time to refer to a dictionary or a statistical model of the language before being able to recognize words.

Recall that experiments, carried on 17 271 word samples, achieved 97% extraction success. The same sample distribution was used in different experiments to compute network recognition rates for a vocabulary of size 5757: first, before and after network collaboration, then with only perceptive cycles and finally with collaboration

Table 9. Convergence of the three TNNs.

	Training Duration (s)	Error	Iteration Number
TNN_R	9267	188.337	<80
TNN_S	2781	319.378	<500
TNN_C	3444	495.071	<1000

Table 10. Network recognition rates, with and without collaboration, with and without perceptive cycles for a vocabulary of size 5757.

	TNN_R (%)	TNN_S (%)	TNN_C (%)
Without collaboration	90.73	95.65	—
With collaboration, without perceptive cycles	94.91	95.65	97.0
With collaboration, with perceptive cycles	95.62	98.75	97.95

Table 11. Gain from network collaboration and perceptive cycles for a vocabulary of size 5757.

	TNN_R (%)	TNN_S (%)	TNN_C (%)	Word
Gain from network collaboration	4.18	—	> 0	
Gain from perceptive cycles	0.71	3.1	0.95	
Gain from collaboration and perceptive cycles	4.89	3.1	> 0.95	95.33

and perceptive cycles. As illustrated in Table 10, the recognition rates confirm not only efficiency of the proposed system but also the contribution of network collaboration in handling with a large vocabulary of Arabic words.

Table 11 displays the obtained gains from network collaboration and perceptive cycles.

The curve, shown in Fig. 26, confirms once again the network stability against vocabularies increasingly large.

Notice that between 1000 and 2000, we used 51 roots. Beyond 2000 and up to 5757 we moved to 101 roots. It turned out that the majority of roots problem (similar roots, examples given in Fig. 27) are among the first 51 roots which explains the disturbance at the beginning of the curve. TNN_R (using 101 roots), TNN_S and TNN_C are stables because the number of schemes and conjugation elements does not change with the expansion of the vocabulary. We expect that by enlarging the vocabulary beyond 6000 words, the model will maintain its stability as the number of schemes and conjugation elements remain unchanged and that the addition of 100 words will not cost more than adding one neuron at the output layer of TNN_R (the number of layers will be the same, also for the number of neurons in the intermediate layers).

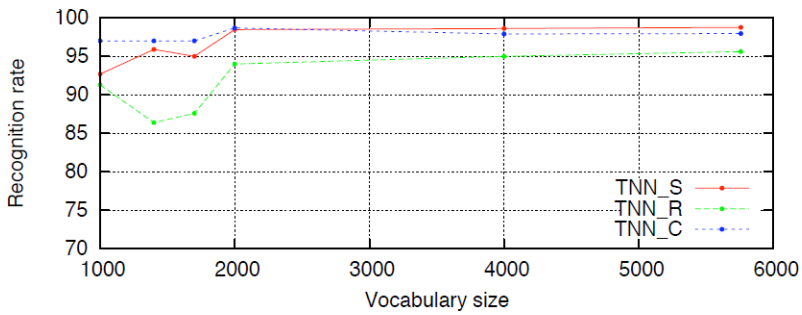


Fig. 26. Network stability against vocabularies increasingly large.

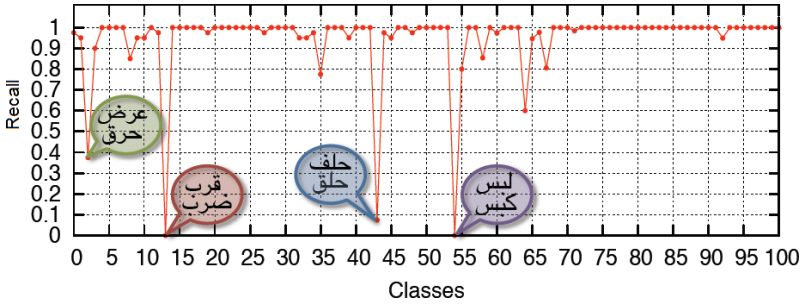


Fig. 27. Recall of each root.

7.4. *Analysis of confusion cases*

We also evaluate the proposed system in terms of precision and recall. Figure 27 illustrates the obtained results by class as a curve. A class corresponds to one of the 101 root. The more the recall of a class is close to 1, the more the classification is good. For example, the recall of the class 94, which corresponds to the root عدل is equal to 1, since all the 168 words that were derived from are well recognized. By cons, the recall of the class 55 which corresponds to the root لعب is 0.8 since only 32 samples among 40 have been well classified. The rest have been associated to the root كسب because of the letter ـعـ without loop (described by the primitives *OB* or *OM*). Consequently, the words derived from the roots لعب and كسب become the same as they have similar descriptions in terms of structural primitives.

We also note that most of classes have a maximum recall (equal to 1) and only 10 classes, including 101, have a recall less than 0.8. The majority of these misclassifications are understandable because the concerned roots are similarly described. The recall can then drop to zero. This is the case, for example, of the root قرب. Its recall is equal to zero because all its words have been associated to the root ضرب. This is due to the same description of their first letters which are in fact different. The same case repeats itself for the roots كبس and لبس. Cases where recall is low, but not zero, does not arise from the same roots but rather from the same words that may have derived from these roots. In fact, the most samples of the class 43, which corresponds to the root حلف, have been assigned to the root حلق despite their last letters are differently described. Although the root حلق is described by a descender letter at its end, the words, which can be derived from this root, may not retain this description (when a suffix is joined to the root at its conjugation for example). In this case, the structural primitives of derived words from both roots can be the same especially as we do not consider the number of diacritic when describing words. Another example concerns the roots عرض and حلق.

Despite these misclassification cases, we can conclude that TNN_R provides good results since an average recall of 0.94 is achieved. If we exclude the four misclassification cases, the recall grows to 0.98.

Table 12. Recognition rate of 150 intruder words.

	TNN_R (%)	TNN_S (%)	TNN_C (%)
Top1	90.15	96.12	94.9

7.5. *Word restitution*

The total top1 of word restitution is 95.33% for a vocabulary of size 5757. Although this rate is satisfactory, it is important to underline the bad influence of identical roots, in terms of structural primitives, on their training and therefore the total rate. If we do not count the four roots, whose recall is low or zero, we obtain a total rate of top1 equal to 97.73%.

7.6. *Resolution of corpus explosion*

Our system is able to recognize words that samples were not learned before: words that do not belong to the vocabulary and we consider intruders. In Table 12, we display the recognition rate of 150 words, without applying perceptive cycles, which do not belong to the learned vocabulary. For example, the words مخالفان and أدركت are not part of the learned vocabulary but as their roots خلف and درك and schemes مفاعل and أفعل have been recognized from other words, they have been recognized. This reveals the generic aspect of the proposed system from its ability to recognize more words without additional learning.

7.7. *Previously added improvements*

In order to carry out the root, scheme and conjugation element network training more efficiently, we automatically developed some enhancement taking into account the letter order, sister letters, network supervision, neuron bursting and ambiguity resolution.

7.7.1. *Letter order*

Letter order plays a key role in word recognition. To better exploit neural network training capacities, their inputs can be set in the interval [0,1] instead of activations only equal to 0 or 1. Thus, network inputs can be used to refer to possible positions of letters in the words. As illustrated, in Fig. 28, to learn the root صرف from the word انصرف, the activation of the letter ص is automatically set to 0.33 since this letter appears on the third position in that word.

7.7.2. *Sister letters*

We call sister letters, letters which are described by the same primitive. For example, ص, ه and ط are sister letters since they are described by the same primitive: *LM* (Loop in the Middle). As shown in Fig. 28, to learn the root صرف, TNN_R automatically activate on the input layer, not only proper letters of the word, but also their sister

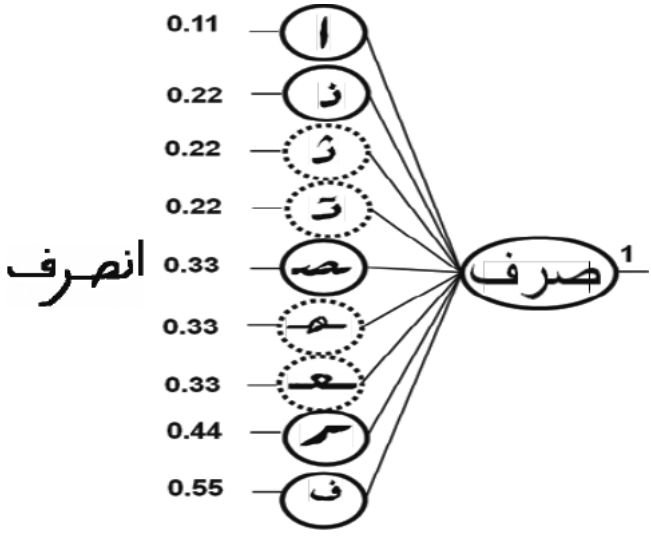


Fig. 28. The use of the letter order and sister letters.

letters (see dotted circles). This avoids competition between sister letters when learning the root. Obviously, the sisters of one letter have the same activation than the letter itself.

7.7.3. *TNN_R and TNN_S supervision*

As TNN_R is designed to focus on the root letters and to ignore access letters (scheme and conjugation letters), we inhibit neuron letters that can never be root letters in the handled vocabulary. Opposed to TNN_R, we inhibit in TNN_S, neuron letters that can never be access letters. As shown in Fig. 29, using the word يتعانقون to learn the scheme تفاعل, we only inhibit the letters ع and ق as they are root letters. Note that ز, which is a root letter, has not been deactivated since it can be also a scheme letter in the following examples of schemes: انفعل, منفعل and انفعال.

7.7.4. *Neuron bursting*

When observing TNN_S behavior for unrecognized words, we noticed that the last occurrence of a letter overwrites the previous occurrences of the same letter. This can easily disrupt TNN_S training and therefore the recognition of schemes, since occurrence of a letter can be either a scheme letter, or a conjugation letter. For example, in Fig. 30, the letter ا of the word حازمان, which follows the scheme فاعل, appears twice: (1) as a scheme letter (second position from the right) and (2) as a conjugation letter (5th position). The first instance of ا is overwritten by the second one which will activate the letter to 0.55. Therefore, we lose a critical letter (ا activated to 0.22) of the scheme فاعل. To solve this problem, we proceeded by neuron bursting. It is about to automatically burst each neuron letter of scheme of

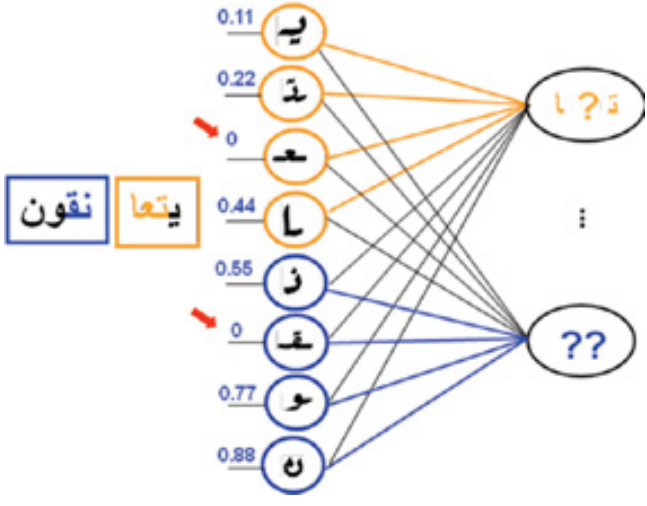


Fig. 29. Example of TNN_S supervision.

TNN_S in a set of neurons. The number of burst neurons depends on the letter scheme. For example, the letter ا has five decisive possible positions in a word, so it will be broken up into five neurons. The letter ز has only two decisive positions (1st and 3rd), and therefore only two neurons will be broken.

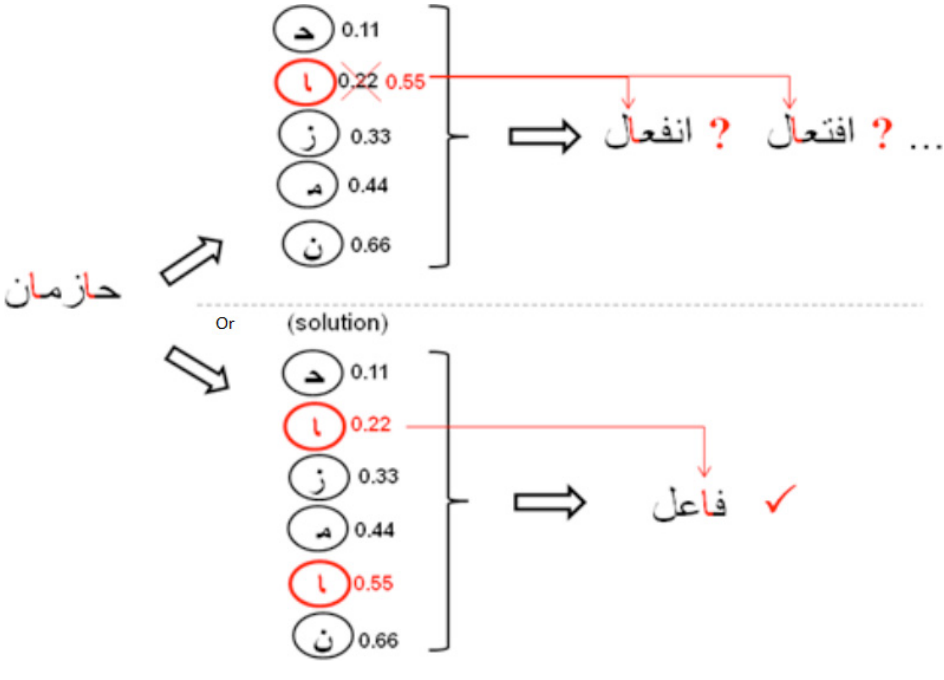


Fig. 30. Neuron bursting.

Table 13. Gain in recognition rate after improvements.

Improvements	TNN_R (%)	TNN_S (%)
Letter order, sister letters and network supervision	8.8	11.5
Neuron bursting and ambiguity resolution (at the training phase)	10	22.9
Total gain	18.8	33.5

7.7.5. *Ambiguity resolution*

Thanks to network transparency, we find that some scheme letters vote competitively when they have the same position in the word. This can lead to confusion cases. For example, when the scheme letters ت and ن, which correspond to the primitive *DaM* (Diacritic above in the middle), are in the second position, as in the words يتحارب and ينصرف, they will simultaneously and respectively vote for schemes تفاعل and انفعل. Therefore, their contributions become weak, so other neurons can take over and collisions may appear. For that, we recourse to some scheme characteristics to decide which letter will exclude the other. For example, when learning the scheme تفاعل from the word يتحارب, the letter ت will remove the letter ن thanks to the presence of the letter scheme ا which is determinant in the scheme تفاعل.

In Table 13, we recapitalize the recognition gain, taking into account the previous improvements.

7.8. *Comparison to systems based on segmentation-free approach*

As noted by Ref. 49, while handwritten Arabic tasks are rather well covered, few printed Arabic databases and competitions are available. So far, most of the printed Arabic systems have actually been benchmarked on private or small-scale databases, making their comparison rather difficult. As mentioned before, we used a private database which contains 17 200 printed Arabic words covering a vocabulary of 5757 decomposable words. We also carried experiments on the Arabic Printed Text Image (APTI) database[47] in the aim to compare with systems reported in ICDAR 2011 competition. This database is synthetically generated using a lexicon of 113 284 words, 10 Arabic fonts, 10 font sizes and four font styles. It is created in low resolution (72 dots/inch). It contains a mix of decomposable and nondecomposable words. The total number of single word images is above 45 million. The use of APTI database has also allowed us to evaluate the system robustness using digitally low resolution word images. In addition, it gave us the opportunity to validate our proposed neural-linguistic approach and demonstrate the interest of linguistic knowledge integration in the recognition process of a large vocabulary of Arabic words based on a global approach, which is known to be limited to reduced vocabulary. To this end, we considered systems tested on the APTI database which are proposed in Refs. 24, 25 and 36 and reported in ICDAR 2011 competition.[49] We also considered system reported in Ref. 48 which is declared out of competition. Notice that all of these systems are based on a segmentation-free approach as described below. Table 14

Table 14. Comparison to systems based on segmentation-free approach.

System	Mean Word Recognition Tate (%)
IPSAR System[36]	77.5
UPV-PRHLT-REC1[24]	84.4
UPV-PRHLT-REC2[25]	84.4
DIVA-REGIM[48]	98.9
Our system[9,13–15]	78.2

summarizes these comparisons. The evaluation is reported as mean of word recognition rates according to the first APTI protocol for ICDAR 2011 Competition: test mono font and mono size. The test images presented to the systems are the one rendered using the font "Arabic Transparent", style plain and size 24.

IPSARec[36] is a cursive Arabic script recognition system based on Hidden Markov Model Toolkit (HTK). This is a portable toolkit for speech recognition system which is customized to recognize characters. IPSARec is an omnifont, unlimited vocabulary recognition system. It does not require segmentation. The proposed system proceeds on three main stages: extracting a set of features from the input images, clustering the feature set according to a pre-defined codebook and finally, recognizing the characters.

The UPV-BHMM Systems[24,25] also are based on Bernoulli HMMs (BHMMs), that is, HMMs in which conventional Gaussian mixture density functions are replaced with Bernoulli mixture probability functions. Also, in contrast to the basic approach followed in Ref. 24, in which narrow, one-column slices of binary pixels are fed into BHMMs, the UPV-BHMM systems are based on a sliding window of adequate width to better capture image context at each horizontal position of the word image. This new, windowed version of the basic approach is described in Ref. 25. Parameter estimation and recognition were carried out using the Expectation-Maximization (EM) algorithm.

The DIVA-REGIM system[48] is based on HMMs. One of its main characteristics is to be open vocabulary, i.e. able to recognize any Arabic printed word. The used HMM sub-models correspond to Arabic characters completed with a selected set of their corresponding variations. The feature vector is extracted from each analysis window. Using a simple right-left sliding procedure of the analysis window, no segmentation into letters is made and the word image is transformed into a sequence of feature vectors. During training time, the EM algorithm is used to iteratively refine the component weights, means and variances to monotonically increase the likelihood of the training feature vectors. At recognition time, an ergodic HMM is built from all character models.

Experimental results show the capability of these systems to recognize a large vocabulary of printed Arabic words in mono font and mono size context using digitally low resolution word images. For UPV-PRHLT-REC1, UPV-PRHLT-REC2 and DIVA-REGIM systems, we observe good results and slightly worse for either

IPSAR System and our system. Both UPV-BHMM Systems have the same behavior and show the best results with an average of 84.4%. We believe that word recognition rate (78.2%) achieved by our system is a very promising result considering the vocabulary size. It is shown that giving slightly higher weight to linguistic knowledge in the recognition process provides solution to handle with large vocabularies.

8. Conclusion and Future Prospects

In this work, we proposed a novel approach to exploit the morphological aspect of Arabic language when building a system based on classifier combination for the recognition of large vocabulary of Arabic words. We focused on a vocabulary of decomposable words which are derived from healthy tri-consonantal roots. The proposed approach is inspired from the cognitive neural system of McClelland and Rumelhart,[42] to represent word in layers: from the local (layer of primitives) to global (layer of possible words) and vice versa. We conceived and implemented a neural-linguistic system based on the collaboration of three TNNs (for root, scheme and conjugation training and recognition). The real challenge, raised by this work, is to propose a collaborative and intelligent cooperation of these three TNNs to ensure the training and recognition of a maximum of words. Experiments on 5757 sized vocabulary confirm the proposed system to be well suited for the recognition of a wide canonical vocabulary. When comparing our system to segmentation-free base systems, reported in ICDAR 2011 competition and using APTI database, we conclude that it reaches satisfactory recognition rate considering the vocabulary size. That is to say that giving slightly higher weight to linguistic knowledge offers not only promising results, but also solutions to handle wider vocabulary. It also demonstrates the usefulness of classifier collaboration in the recognition system.

As future work, we plan:

1. to resume experiments on wider and wider lexicons, by enlarging them exponentially,
2. to extend perceptive cycles in order to be able to resolve ambiguities between any letters (not only letters presenting flat or acute shape) using "Hu moments" for example,
3. to build and test with handwritten Arabic word database especially since we used global features that are easy to extract even from handwritten word and
4. to test the classifier combination by the use of different classification methods (structural, statistical, stochastic, etc.) to enhance system reliability, profiting from classifier behavior differences and thus from their potential complementarily.

References

1. B. Al-Badr and R. Haralick, A segmentation-free approach to text recognition with application to Arabic text, *Int. J. Doc. Anal. Recogn.* **1**(3) (1998) 147–166.

2. S. Al-Fedaghi and F. Al-Anzi, A new algorithm to generate Arabic root-pattern forms, in *Proc. 11th National Computer Conf.*, King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia (1989), pp. 4–7.
3. I. Aljarrah, O. Al-Khaleel, K. Mhaidat, M. Alrefai, A. Alzu'bi and M. Rabab'ah, 'Automated system for Arabic Optical Character recognition with lookup dictionary, *J. Emerg. Technol. Web Intell.* **4** (2012) 362–370.
4. M. Aljlayl and O. Frieder, On arabic search: Improving the retrieval effectiveness via a light stemming approach, in *Proc. Int. Conf. Information and Knowledge Management (CIKM)*, November 2002 (ACM), pp. 340–347.
5. I. A. Al-Sughaiyer and I. A. Al-Kharashi, Arabic morphological analysis techniques: A comprehensive survey, *J. Am. Soc. Inf. Sci. Technol.* **55**(3) (2004) 189–213.
6. A. Amin and W. Mansoor, Recognition of printed Arabic text using neural networks, in *Proc. 4th Int. Conf. Document Analysis and Recognition, ICDAR '97*, Washington, DC, USA (1997), pp. 612–615.
7. S. Barkallah, S. Ben Chobba and S. Kanoun, Intgration de connaissances linguistiques dans le processus de reconnaissance hors-ligne de mots dcomposables imprims arabes: Etude Comparative, *Proc. CIFED'12* (2012), pp. 37–54 (in French).
8. I. Bazzi, R. Schwartz and J. Makhoul, An omnifont open-vocabulary OCR system for English and Arabic, *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(6) (1999) 495–504.
9. I. Becheikh, Combinaison de classifieurs linguistiques pour la reconnaissance d'un vocabulaire large de mots arabes, Ph.D. Thesis, Faculté des sciences économiques et de gestion de Sfax, Tunisia (2012).
10. I. Becheikh and A. Kacem, Neural network for the recognition of handwritten tunisian city names, in *Proc. ICDAR'07*, Curitiba, Paran, Brazil, September 2007 (IEEE Computer Society), pp. 1108–1112.
11. A. Belaïd and C. Choisy, Human reading based strategies for off-line Arabic word recognition, in *Proc. SACH'06*, Summit on Arabic and Chinese Handwriting, Univ. of Maryland, College Park (2006), pp. 27–28.
12. N. Ben Amara and A. Belaïd, Modlisation pseudo bidimensionnelle pour la reconnaissance de chanes de caractres arabes imprims, in *Proc. CIFED'98*, Qubec, Canada (1998), pp. 131–141.
13. I. Ben Cheikh, A. Belaïd and A. Kacem, A novel approach for the recognition of a wide Arabic handwritten word lexicon, in *Proc. ICPR08*, Florida (2008), pp. 1–4.
14. I. Ben Cheikh, A. Kacem and A. Belaïd, A neural perceptive model for the recognition of a large canonical Arabic word vocabulary, in *Proc. ACIT'09* Ymen (2009).
15. I. Ben Cheikh, A. Kacem and A. Belaïd, A neural-linguistic approach for the recognition of a wide arabic word lexicon, in *Proc. DRR'10, Electronic Imaging Symp., SPIE Proceedings*, Vol. 7534, San Jose, CA, USA (2010), pp. 1–10.
16. A. Ben Hamadou, Vrifying et Correction Automatiques par Analyse Affixale des Textes Ecrits en Langage Naturel, Ph.D. Thesis, Faculty of Sciences of Tunis (1993) (in French).
17. A. Benouareth, A. Ennaji and M. Sellami, Arabic handwritten word recognition using HMMs with explicit state duration, *EURASIP J. Adv. Signal Process.* **2008** (2008) 1–13.
18. F. Biadsy, R. Saabni and J. El-Sana, Segmentation-free online Arabic handwriting recognition, *Int. J. Pattern Recogn. Artif. Intell.* **25** (2011) 1009–1033.
19. M. Cheriet and M. Beldjehem, Visual processing of Arabic handwriting: Challenges and new directions, in *Proc. SACH'06*, India (2006), pp. 1–21.
20. G. Dimauro, S. Impedovo, G. Pirlo and A. Salzo, Automatic bankcheck processing: A new engineered system, *Int. J. Pattern Recogn. Artif. Intell.* **11** (1997) 467–503.
21. M. M. Fahmy and S. Al Ali, Automatic recognition of handwritten Arabic characters using their geometrical features, *Stud. Informat. Contr. J.* **10** (2001) 467–503.

22. N. Farah, L. Souici and M. Sellami, Classifiers combination and syntax analysis for Arabic literal amount recognition, *Eng. Appl. Artif. Intell.* **19**(1) (2006) 29–39.
23. F. Farooq, D. Jose and V. Govindaraju, Phrase-based correction model for improving handwritin recognition accuracies, *Pattern Recogn.* **42** (2009) 3271–3277.
24. A. Gimenez and A. Juan, Embedded Bernoulli mixture hmms for handwritten word recognition, in *Proc. ICDAR* (2009), pp. 896–900.
25. A. Gimenez, I. Khoury and A. Juan, Windowed bernoulli mixture hmms for arabic handwritten word recognition, in *Proc. ICFHR 2010*, Kolkata, India (2010), pp. 533–538.
26. H. Goraine, M. Usher and S. Al-Emami, Off-line Arabic character recognition, *Computer* **25**(7) (1992) 71–74.
27. D. Guillevic and C. Y. Suen, HMM-KNN word recognition engine for bank cheque processing, in *Proc. Int. Conf. Pattern Recognition*, Brisbane, Australia (1998), pp. 1526–1529.
28. K. M. Hassibi, Recognition of Arabic machine-printed cursive text, in *Proc. Int. Conf. Pattern Recog.* (1996), pp. 1207–1214.
29. L. Heutte, Reconnaissance de caractres manuscrits: Application la lecture automatique des chques et des enveloppes postales, Ph.D. Thesis, Universit de Rouen, France (1974) (in French).
30. L. Heutte, P. Pereira, O. Bougeois, J. Moreau, B. Plessis and P. Courtellemont, Multibank check recognition system: Consideration on the numeral amount recognition module, *Int. J. Pattern Recogn. Artif. Intell.* **11** (1997) 595–617.
31. Y. Hilal, Tahlil Sarfi Lil Arabia, in *Proc. Computer Processing of Arabic Language*, Kuwait (1985).
32. W. Kammoun and A. Ennaji, Reconnaissance de Textes Arabes Vocabulaire Ouvert, in *Proc. CIFEC*, France (2004) (in French).
33. S. Kanoun, Identification et Analyse de Textes Arabes par Approche Affixale, Ph.D. Thesis, Universit des Sciences et techniques de Rouen, France (2002) (in French).
34. S. Kanoun, A. Alimi and Y. Lecourtier, Affixal approach for Arabic decomposable vocabulary recognition: A validation on printed word in only one font, in *Proc. ICDAR'05*, Seoul (2005), pp. 1025–1029.
35. S. Kanoun, M. A. Alimi and Y. Lecourtier, Natural language morphology integration in off-line Arabic optical text recognition, *IEEE Trans. Syst. Man Cybern. B, Cybern.* **41**(2) (2011) 579–590.
36. M. S. Khorsheed, Offline recognition of omnifont arabic text using the hmm toolkit (htk), *Pattern Recogn. Lett.* **28** (2007) 1563–1571.
37. M. S. Khorsheed and W. F. Cloksin, Spectral features for Arabic word recognition, *Proc. ICASSP'00*, Vol. 6 (San Diego State University, 2000), pp. 3574–3577.
38. A. L. Koerich, R. Sabourin and C. Y. Suen, Large vocabulary off-line handwriting recognition: A survey, *Pattern Analy. Appl.* **6** (2003) 97–121.
39. Y. Lu and M. Shridhar, Character segmentation in handwritten words — An overview, *Pattern Recogn.* **29** (1996) 77–96.
40. S. Madhvanath, V. Govindaraju and V. Ramanaprasad, Reading handwritten US census forms, in *Proc. Int. Conf. Document Analysis and Recognition*, Montreal, Canada (1995), pp. 82–85.
41. S. Masmoudi Touj, N. Essoukri Ben Amara and H. Amiri, A hybrid approach for offline Arabic handwriting recognition based on a planar hidden markov modeling, in *Proc. 9th Int. Conf. Document Analysis and Recognition*, Vol. 2, Washington, DC, USA, (IEEE Computer Society, 2007), pp. 964–968.
42. J. L. McClelland and D. E. Rumelhart, An interactive activation model of context effects in letter perception, *Psychol. Rev.* **88** (1981) 375–4076.

43. H. Miled, C. Olivier, M. Cheriet and Y. Lecoutie, Coupling observation/letter for a markovian modelisation applied to the recognition of arabic handwriting, in *Proc. 4th Int. Conf. Document Analysis and Recognition, ICDAR'97*, Washington, DC, USA (1997), pp. 580–583.
44. M. T. Parvez and S. A. Mahmoud, Offline arabic handwritten text recognition: A survey, *ACM Comput. Surv.* **45**(2) (2013).
45. S. Procter, J. Illingworth and F. Mokhtarian, Cursive handwriting recognition using hidden Markov models and a lexicon-driven level building algorithm, *IEE Proc. Vis., Image Signal Process.* **147**(4) (2000) 332–339.
46. F. Slimane, R. Ingold, A. M. Alimi and J. Hennebert, Duration models for Arabic text recognition using hidden Markov models, in *Proc. IEEE Int. Conf. on Computational Intelligence for Modelling, Control and Automation (CIMCA 08)*, Vienna, Austria (2008), pp. 838–843.
47. F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi and J. Hennebert, A new arabic printed text image database and evaluation protocols, in *Proc. ICDAR* (2009), pp. 946–950.
48. F. Slimane, R. Ingold, S. Kanoun, A. M. Alimi and J. Hennebert, Impact of character models choice on arabic text recognition performance, in *Proc. ICFHR* (2010), pp. 670–675.
49. F. Slimane, S. Kanoun, H. El Abed, A. M. Alimi, R. Ingold and J. Hennebert, Arabic recognition competition: Multi-font multi-size digitally represented text, in *Proc. ICDAR* (2011), pp. 1449–1453.
50. A. W. Senior and A. J. Robinson, An off-line cursive handwriting recognition system, *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(3) (1998) 309–321.
51. S. N. Srihari, Recognition of handwritten and machine-printed text for postal address interpretations, *Pattern Recogn. Lett.* **14** (1993) 291–302.
52. S. Srihari, H. Srinivasan, P. Babu and C. Bhole, Handwritten arabic word spotting using the cedarabic document analysis system, in *Proc. Symp. Document Image Understanding Technology (SDIUT-05)*, College Park, MD (2005), pp. 123–132.

Afef Kacem Echi received her M.Sc. and Ph.D. degrees in Computer Science from the National School of Computer Sciences of Tunis in 1997 and 2001, respectively. Since 2000, she has been an assistant in the Computer Science Department at the Faculty of Sciences of Monastir, and was appointed Assistant Professor there in 2002. Dr. Kacem is a responsible member of the research area: Analysis and Recognition of Handwriting and document in LaTICE: Laboratory for Technologies of Information and Communication at the High School of Sciences and Techniques of Tunis. Dr. Kacem authored the book "*Vers la Rétro-conversion de Documents Mathématiques*", published by Editions universitaires europeennes (September 19, 2011).

Imen Ben Cheikh received her Ph.D. degree in Computer Sciences, in 2012, from the Faculty of Economic Sciences and Management of Sfax-Tunisia. Since 2010, she has been an assistant in the computer science department at the High School of Arts and Multi-Media. Dr. Ben Cheikh is a member of research teams of LaTICE: Laboratory for Technologies of Information and Communication at the High School of Sciences and Techniques of Tunis. She has been working on document analysis and writing recognition.

Abdel Belaïd received his Ph.D. in Computer Science in 1979 and his D.Sc. degree in 1987 from Henry Poincaré University, Nancy I, France. After working a few years as Assistant Professor in this University, he joined the National Center for Scientific Research CNRS as a Research Scientist in 1984. In 2002, he became Professor at the University Nancy 2. He initiated the Cognitive Science Master and led it until 2008. He teaches various aspects of Computer Science, Human Man interfaces and Multimedia. His areas of research include pattern recognition, document analysis and handwriting recognition where he has authored over 150 articles in various national and international journals and conference proceedings. He is the co-author of a book, "*Pattern Recognition: Methods and Applications*". He leads a research group READ at the UMR LORIA 7503 working on document analysis and text recognition.