A Neural-Linguistic Approach for the Recognition of a Wide Arabic Word Lexicon

Imen Ben Cheikh, Afef Kacem, Abdel Belaïd

A Neural-Linguistic Approach for the Recognition of a Wide Arabic Word Lexicon

I. Ben Cheikh[a], A. Kacem1 and A. Belaïd[b]
[a]UTIC-ESSTT, 5 Avenue Taha Hussein, BP56 Mnara, Tunis, Tunisie
Imen.becheikh@gmail.com, Afef.kacem@esstt.rnu.tn;
[b]LORIA, Campus scientifique, B.P. 239, 54606 Vandœuvre-Lès-Nancy, France
abelaid@loria.fr

ABSTRACT

Recently, we have investigated the use of Arabic linguistic knowledge to improve the recognition of wide Arabic word lexicon. A neural-linguistic approach was proposed to mainly deal with canonical vocabulary of decomposable words derived from tri-consonant healthy roots. The basic idea is to factorize words by their roots and schemes. In this direction, we conceived two neural networks TNN_R and TNN_S to respectively recognize roots and schemes from structural primitives of words. The proposal approach achieved promising results. In this paper, we will focus on how to reach better results in terms of accuracy and recognition rate. Current improvements concern especially the training stage. It is about 1) to benefit from word letters order 2) to consider "sisters letters" (having same features), 3) to supervise networks behaviours, 4) to split up neurons to save letter occurrences and 5) to solve observed ambiguities. Considering theses improvements, experiments carried on 1500 sized vocabulary show a significant enhancement: TNN_R (resp. TNN_S) top4 has gone up from 77% to 85.8% (resp. from 65% to 97.9%). Enlarging the vocabulary from 1000 to 1700 by 100 words, again confirmed the results without altering the networks stability.

Keywords: Arabic writing recognition, wide vocabulary, neural networks, linguistic knowledge integration, canonic vocabulary.

1. INTRODUCTION

Many researches have been undertaken in the field of typed and handwritten Latin, Chinese and Japanese writing recognition. Little progress, however, has been made in Arabic writing due to its cursive nature. Besides, several approaches (holistic, pseudo-global and analytic) have been proposed according to vocabulary size (reduced, slight extended or wide).

The recognition of a wide vocabulary of Arabic words poses challenges from not only the characteristics of Arabic language, but also the limits of existing approaches and paucity of training data available. Because of Arabic morphological complexity, effective Arabic words go past 60 billions [1] what makes their automatic processing unrealistic and constitutes handicaps for dictionary building, automatic spelling... To remedy such problem, reducing this number becomes mandatory. To this end, words morphological analysis and factorization seem to be one solution.

Current trends are for incorporating linguistic knowledge, at various levels (lexical, syntactic and semantic), either keeping them as an independent stage, or integrating them partly or totally in the recognition chain [1]. In this context, we proposed, in a previous work [2], a neural-linguistic approach based on two neural networks: TNN_R and TNN_S to respectively recognize the root, from which the word derives, and the scheme, the word follows, by the use of linguistic knowledge. Carried experiments demonstrated that results are promising. Now, we will mainly focus on how to improve the approach in order 1) to achieve better results, in terms of accuracy and recognition rate and 2) to handle with more and more large vocabularies.

In section 2, we will illustrate how the Arabic linguistic knowledge was considered in writing recognition in previous related research works. Next, we will remind the concepts underlying our approach. Afterwards, we will explain our improvement strategy. By conducting experiments, we will finally demonstrate the networks stability, face to wider and wider vocabularies, and investigate their shortcomings to plan for further improvements as future work.

2. ARABIC LINGUISTIC KNOWLEDGE INTEGRATION

Many studies highlight the richness and the stability of Arabic in terms of morpho-phonologic peculiar to this language [1,3,4,5]. An Arabic word is decomposable or not. If it derives from a root, it is said to be decomposable in morphemes (root, prefix, infix and suffix). A word is, then, composed of root letters and access (non-root) letters. In [1], Cheriet proposed to exploit this word vision using any recognition approach. He suggested analysing errors using linguistic clues and in rejection cases, extracting the root and doing template matching to infer the rest. He affirmed that one must focus on what kind of linguistic knowledge is important and how and where it is more appropriate to incorporate. By adopting the same vision of the word, an "affixal approach", proposed by [4] then reconsidered by [5] for the recognition of Arabic typed texts, consists in the segmentation of words in letters and the recognition of their morphological entities. The authors used several linguistic concepts (Affixal and semantic restrictions) to guide the recognition process.

Our vision of the word is slightly different. In fact, 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...) to a brief scheme. The derivation of the root with the brief scheme gives rise to the radical (see Fig. 1). 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. Thus, according to our vision, to recognize a word, we just need to identify its root and conjugated scheme (conjugation elements), without segmenting it in letters. This is the main idea which will allow us to factorize words and to handle with a wide vocabulary while using a holistic approach to avoid effective segmentations.

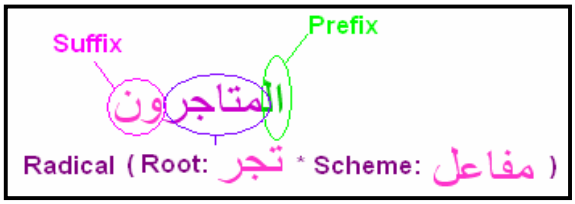


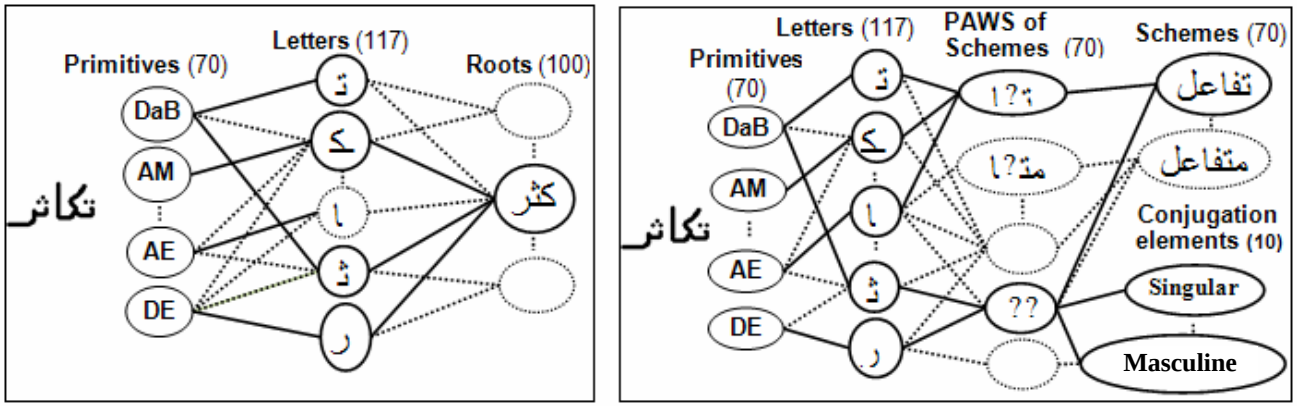
Fig. 1. Our Arabic word vision.

3. NEURAL-LINGUISTIC APPROACH

In [2], we proposed an approach for the recognition of a wide Arabic canonical vocabulary. The idea is to integrate linguistic knowledge to a perceptive model by the use of transparent neural networks (TNN), having as input global primitives of words. Such model allows not only to avoid segmentation, due to cursive nature of Arabic, but also to fasten result analysis and error correction since neurons are no longer black boxes; they are rather associated with concepts. It is important to notice that 1) we have already experimented the use of TNN on a reduced vocabulary as reported in [6] and 2) the considered system was inspired from a human vision system developed, first, by Mc. Culloch, used then for Latin by M. Côté, and reconsidered for Arabic by S. Maddouri [7] operating, first, globally by extracting the most obvious primitives, then while refining its vision following several perceptive cycles.

Within this approach, the recognition is founded on two transparent neural networks TNN_R and TNN_S, having as input the word structural primitives, but TNN_R tries to provide its root when TNN_S tries to supply its scheme. Dealing with large vocabulary of 8000 words for example (derived from 100 roots using different schemes and conjugations), we have succeeded in reducing necessary neurons from 8000 to 180 (100 neurons for roots + 80 neurons for schemes and conjugation

elements) by adopting the following architectures for TNN_R and TNN_S (see Fig. 2). TNN_R is a three-layer network which learns how to focus on root letters and ignores access ones, reserved for schemes, while TNN_S is a four-layer network which learns how to ignore root letters and focus on schemes ones.



(a) (b)

Fig. 2. (a): TNN_R Architecture, (b): TNN_S Architecture

Note that dissociating roots from schemes was necessary since these two entities do not require the same information to be learned. In fact, word PAWs (Piece of Arabic Word) are useless for the training of its root (e.g. PAWs of both words "كافح" and "كفاح" correspond to the root "كفح") but useful for the training of its scheme (e.g. PAWs of "كافح" lead to the scheme "فاعل" whereas PAWs of "كفاح" match the scheme "فعا ل").

For the training of TNN_R and TNN_S, which are transparent networks, instead of adjusting their connections weights manually, we have chosen to dissociate them into mono-layer-Perceptrons to make their automatic training possible. Previous experimentations, carried on 1500 sized vocabulary are justly considered as a proof of the concept. Reached results were encouraging but not satisfying enough (top4 for TNN_R and TNN_S are resp. equal to 77% and 65%). Next, we will explain our improvement strategy.

4. NEURAL-LINGUISTIC APPROACH IMPROVEMENT

The improvement of our approach mainly concerns the training stage:

- Exploiting word letters order information,
- Introducing the "sisters letters" concept,
- Supervising TNN_R and TNN_S behaviours,
- Splitting up of neurons, and
- Solving ambiguities during the training stage.

The improvement might be extended to the recognition stage in order to handle collisions using "Perceptive cycles" and "Linguistic constraints".

4.1. Letters order information

Since neural networks have great learning capacities, we think to better exploit network inputs by the use of values in the interval [0,1] instead of only 0 or 1. These values refer to possible letter positions in the word. As shown in Fig. 3, to learn the root "صرف" of the word "انصرف", the activation of the letter " ص ", will be 0.33 as it refers to the third position.

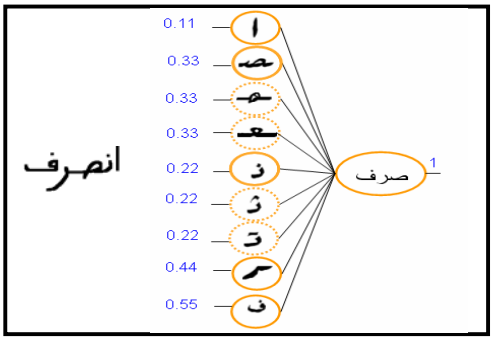


Fig. 3. The use of letters order and sisters letters

4.2. "Sisters letters" concept

Sisters of a letter are those which are described by the same primitive. For example, " ص ", " م " and " ه" are sisters letters, since all of them present the same primitive : "LM": Loop in the Middle. As shown in Fig. 3, to learn the root "صرف" of the word "انصرف", we activate, in inputs layer, not only letters composing the word but also sisters of each letter (see dotted circles). This avoids sisters letters to be concurrent in the root training. Of course, sisters of one letter have the same activation as the letter itself.

4.3. TNN_R and TNN-S supervising

Since TNN_R is conceived to focus on root letters and ignore access ones, we will inhibit neurons of letters which can never be root letter. Oppositely, for TNN_S, we will inhibit neurons of letters which can never be access letter. For example, as illustrated in Fig. 4, in the training of the scheme "تفاعل", composed of " ا ? ت " and "??" ("?" refers to a root letter), we will inhibit the root letters " ق " and " ع ". Note that the letter " ن ", which is here a root letter, is not deactivated since it could be a scheme letter like in the schemes "انفعل", "منفعل" and "انفعال".

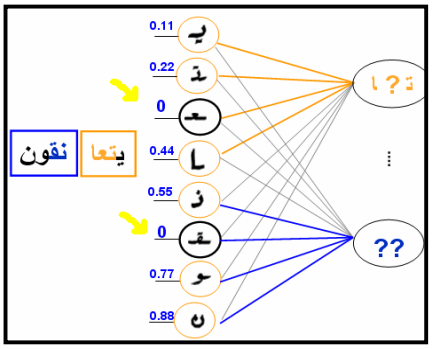


Fig. 4. TNN_S supervising

Note that the use of letters order, sisters letters and networks supervising contributes to considerably increase the recognition rates especially for TNN_R (top4 from 77% to 85.8% with top1=76.1%). For TNN_S, the obtained results have also grown but not enough (top4 from 65% to 76.5% with top1=35%). That incites us to look for further improvements for TNN_S.

4.4. Neurons splitting up

Observing TNN_S recognition behaviour face to miss-recognized words, we find that the last occurrence of a letter overwrites the previous ones. For example, in the word "حازمان" which matches the scheme "فاعل ", the letter " ا " occurs twice: 1) as a scheme letter (2nd position) and 2) as a conjugation letter (5th position). The first occurrence gets over by the second one which will set the activation of " ا " to 0.55. Therefore, we lose a determining letter (" ا " activated to 0.22) of the scheme "فاعل ". By the same way, the last occurrence of a scheme letter, in a word, overwrites the previous ones. For example, in the word "اقتراب" which corresponds to the scheme "افتعال", the letter "ا" is, in both positions, a scheme letter.

To address this problem, we believe in the neurons splitting up. It is about to split up each scheme letter neuron of TNN_S into a set of neurons. The number of split neurons depends on the scheme letter. For example, " ا " has five possible determinant positions in a word, thus it will be split into five neurons, while " ت " has only two significant positions (1st and 3rd), then only two split neurons. Note that scheme letter neurons splitting up permitted to enhance the results: top4 from 76.5% to 95% (top1 from 35% to 68%).

4.5. Ambiguities solving at the training stage

Through TNN_S transparency, we realized that some schemes' letters vote competitively when they have the same position in the word. This led to cases of confusion. For instance, if the schemes' letters " ت " and " ذ ", which have the primitive "DaM" (Diacritics above in the Middle), are in the second position (e.g. in the words "يتحارب" and "ينصرف"), they will vote, simultaneously and respectively, to the schemes "تفاعل " and "انفعل". Therefore, their contributions weaken what allows other neurons to take over and cause collisions. Further improvement for TNN_S training is, then, necessary. This is to ensure that these letters vote exclusively, since two different letters cannot be in the same position of the same word. For that, we will take advantage from some schemes' features to decide which one will exclude the other. For example, when learning the scheme "تفاعل " from the word "يتحارب", the letter " ت " will exclude the letter " ذ ", thanks to the presence of the scheme letter " ا ". This improvement has enabled us to reach a top1 equal to 95.4% for TNN_S.

5. EXPERIMENTATIONS

Considering the above proposed improvements, we conducted experimentations aiming 1) the survey of the behaviour of the model with regard to more and more large vocabularies and 2) the visual inspection of the networks, for a given vocabulary, to identify their weaknesses and to act well accordingly.

5.1. Model behaviour face to vocabulary size

To observe TNN_R and TNN_S behaviours, we used 8 different sized vocabularies ranging from 1000 to 1700 by injecting 100 new words each time. We executed the recognitions using test corpus specific to each vocabulary.

As shown in Fig. 5(b), TNN_R behaves in a stable manner with the growth of the vocabulary size. Note that a good top4 is sufficient, since we plan to apply "perceptive cycles" to get the right root among the first 4 candidates.

The more the vocabulary is large, the better TNN_S learns as illustrated by Fig. 5 (b). This result was expected since the schemes number is fixed. It is important to note that the training of a 1700 sized vocabulary gives better results than the 1600 sized one.

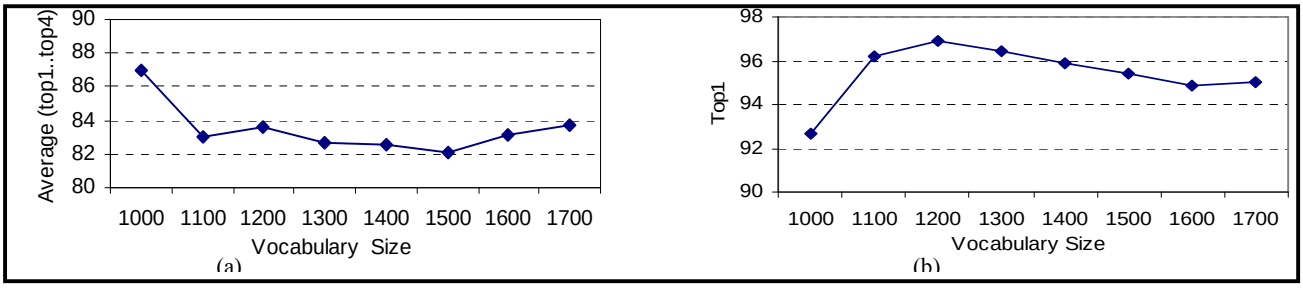


Fig. 5. TNN_R (a) and TNN_S (b) behaviors face to vocabulary size

5.2. Model visual inspection

Considering a vocabulary with a given size (for instance 1700), it is about to investigate TNN_R and TNN_S limits.

TNN_R Investigation: We noted that TNN_R does not manage to learn the roots of some words. Observing some collisions, we found that they are due to root letters which are described by the same primitive, as illustrated in Fig. 6. This figure shows two examples (a,b) of words in which, one primitive could make appear many letters (see red continue circles and arrows). Thus, for one word, more than one root could be candidates (see green and blue dotted arrows and squares).

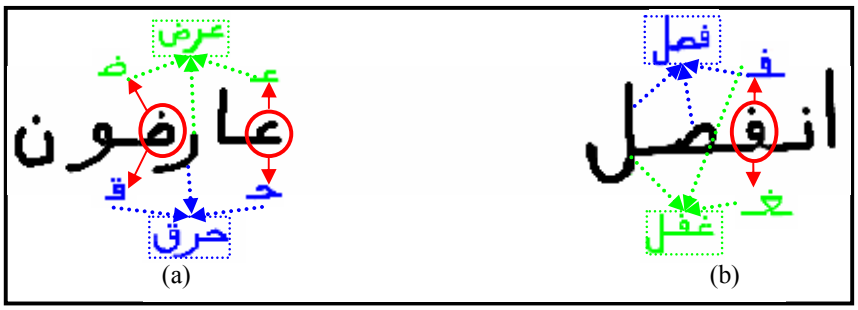


Fig. 6. Root collisions due to similar letters.

TNN_S Investigation : By the same way, we concluded that some schemes have not been well learned. As explained in Fig. 7, we found that many ambiguities arise from letters which are, at once, root and access letters. This figure gives an example of a word which can be seen in two different views because of letters which are root and access letters at once. Consequently, two schemes are concurrent: according to view1, green continue encircled letters contribute to a scheme whereas, in view2, blue dotted ones contribute to another.

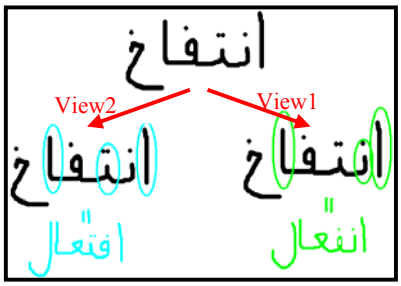


Fig. 7. Scheme collisions due to root/access letters.

6. CONCLUSION AND FUTURE WORK

This work shows that giving slightly higher weight to linguistic information offers not only better results, but also solutions to handle wider vocabulary. In comparison with our previous work [2], the taking into account of the above improvements (letters order, "sister letters", neurons splitting up, networks supervising and ambiguity solving) has proven to significantly help the model reach its classification goal. Results are provided and show that, even without considering yet the suggested improvement proper to the recognition stage (perceptive cycles and linguistic constraints), our approach always produces higher success rates than the current approaches with regard to the vocabulary size as illustrated in table 1. Note that we are comparing our pseudo-global "neural-linguistic" approach to other ones which 1) use linguistic knowledge like [4] and [5] and/or 2) are conceived for large vocabularies like [8], even though the experimentation lexicon of some of them is not wide enough.

To demonstrate the robustness of our approach face to wider vocabulary, we tested it on different sized lexicons (from 1000 to 1700) and succeeded to prove that the model maintains a stable behavior. These experiments confirm neural-linguistic approach to be well suited for the recognition of a wide vocabulary of Arabic words. For the recognition stage, we believe that "perceptive cycles" and "linguistic constraints" should bring a distinct improvement of the current results.

Table 1. Proposed approach vs. current approaches

Approach	**Writing**	**Vocab. Size**		**Top1**	**Top2**	**Top3**	**Top4**
Analytic [4]	Typed	1000		74	81.2	83.9	85
Analytic [5]	Typed	1423		81.3	95.7	96.4	99.7
Analytic [8]	Handwritten	25		88,7	-	-	-
Pseudo-global (our approach)	Typed	1700	TNN_R	77.5	83.8	86	87.6
			TNN_S	95	97.3	97.8	98.7

REFERENCES

[1] M. Cheriet and M. Beldjehem, Visual Processing of Arabic Handwriting: Challenges and New Directions, SACH'06, India.

[2] I. Ben Cheikh, A. Belaid and A. Kacem, A Novel Approach for the Recognition of a Wide Arabic Handwritten Word Lexicon, ICPR'08, Florida.

[3] A. Ben Hamadou, Vérification et Correction Automatiques par Analyse Affixale des Textes Ecrits en Langage Naturel. PHD, Faculty of Sciences of Tunis (1993).

[4] S. Kanoun, A. Alimi and Y. Lecourtier, Affixal Approach for Arabic Decomposable Vocabulary Recognition: A Validation on Printed Word in Only One Font, ICDAR'05, Seoul.

[5] W. Kammoun and A. Ennaji, Reconnaissance de Textes Arabes à Vocabulaire Ouvert, CIFED, France (2004).

[6] I. Ben Cheikh and A. Kacem. Neural Network for the Recognition of Handwritten Tunisian City Names. ICDAR'07, Brasil.

[7] S. Maddouri and al, Local Normalization Towards Global Recognition of Arabic Handwritten Script, Document Analysis and Systems (DAS'00), Brasil.

[8] S. Touj, N. Ben Amara and H. Amiri, A Hybrid Approach for Off-line Arabic Handwriting Recognition Based on a Planar Hidden Markov Modeling, ICDAR'07, Brazil.