Markovian Modeling of Arabic Cursive Handwriting : an Analytical Approach

H. Miled[1]♦*, M. Cheriet*, C. Olivier♦♠
♦ PSI-La3i, UFR des Sciences, Université de Rouen, France
* LIVIA, École de Technologie Supérieure, Montréal, Canada
♠ SIC-IRCOM, URM CNRS 6615 Université de Poitiers, France
[1] e-mail: miled@la3i.univ-rouen.fr

Abstract

Because of the nature of cursive handwriting, our approach segments a word into graphemes. Each extracted grapheme is represented by a vector of 19 features then is transformed into an observation. This stage can be considered as a vector quantization (VQ). A Markovian modeling of characters enables the management of the sequence of observations obtained (from the VQ) in order to estimate different word model parameters. This analytical modeling is based on the three levels of abstractions of the written Arabic: character, pseudo-word (intrinsic notion to Arabic handwriting), and word. In order to optimize the performance of the classifier, we studied two different observation alphabets. The first is selected by an entropy criterion, while the second is obtained by a clustering algorithm (k-means). The performances of these two alphabets are given, showing the superiority of the entropy one: recognition rate of 79.5% is achieved (top rank) using a lexicon of 232 classes and a test set of 5900 words.

keywords: Arabic handwriting recognition, Mutual Information, features selection, word description, HMM modeling.

1. Introduction

The optical reading of printed Arabic has been an interesting challenge for the last twenty years [1]. Handwritten Arabic recognition has not been studied that much [2-3], which we attribute essentially to the fundamental difference, in terms of rules and alphabets, between Arabic and the other scripts (such as Latin, Chinese, and Indian). Arabic, handwritten or printed, is semi-cursive (Figure 2). An Arabic word is a sequence of disjoint connected components called pseudo-words, and similarly each pseudo-word is a sequence of completely cursive characters. Arabic, contrary to Latin, is written from right to left.

The Arabic alphabet is composed of 28 different characters. The shape of a character varies according to its position (at the beginning, in the middle, at the end of a pseudo-word, or isolated). This rule is true except for six characters (marked with a "*" in Figure 1) which can never be attached to their successors. The Arabic character set can be divided into 18 sub-sets. Each sub-set contains characters with identical dominant shape, also called character main body. The different characters in each sub-set can be distinguished by the number and position of their dots (Figure 1).

There are various Arabic cursive handwriting styles. We have grouped them into two large categories: Oriental and North African. Each of these categories has its own signature regarding the shape of the different characters. This does not prevent them from having common aspects, like the presence of what we call ligatures. These ligatures are particular occurrences of links between two consecutive characters (Figure 2). Ligatures are discussed in [4], but only in the case of printed Arabic.

Handwriting recognition is based mainly on two types of approaches: global and analytical. The choice of the approach depends essentially on the type of problem to be addressed, and on the size of the lexicon. In our application, we study a relatively complex problem with a large number of word classes (232), for each of which we only have a limited number of samples. In such a case, the choice of an analytical approach seems appropriate. In order to describe a word as a sequence of features, we develop a segmentation module [5] to cut the word into graphemes. After vector quantization, these graphemes represent classification features. The engine used in our analytical module is a Hidden Markov Model (HMM). HMMs have proved their capacity to model phenomena, which evolve with respect to time, such as

speech [6-7] and writing recognition [8-9]. In reality, HMMs are largely used in different domains of pattern recognition. They have also demonstrated their capacity and their flexibility to model multi-writer handwriting, despite the large variability of writing styles [10]. A study [11] analyzes the optimal order of cursive recognition models.

This paper is divided into six main parts. First, we introduce the two fundamental notions of Arabic handwriting: pseudo-word and tracing. Second, we summarize our segmentation method, described in [5]. Third, we explain the extraction of primitives and the classification of graphemes. Fourth, we present Markovian modeling, with the estimation phase of the different parameters. Fifth, we give the performance of the classifier we developed. Finally, we terminate with a general conclusion.

2. Pseudo-word vs. tracing

When writing, the writer is constrained to go from one pseudo-word to another, each time he meets one of the six characters mentioned in the introduction. Unfortunately, many writers do not respect this constraint. The detection of pseudo-words in a word is not a simple task; parts of the connected components are either portions of pseudo-words, or concatenations of different pseudo-words (Figure 3). We adopt a new notion in our study and call it a tracing.

To distinguish between the notions of pseudo-word and tracing, we give the two following definitions:

- ***Definition 1:*** The pseudo-word is a *logical entity* composed of a sequence of characters, a theoretical sequence of characters as it should appear in reality.
- ***Definition 2:*** The tracing is a *physical entity* composed of a sequence of characters as produced by the writer.

Consequently, a tracing can be a pseudo-word (Figure 3-a), a concatenation of pseudo-words (Figure 3-b) or a portion of a pseudo-word (Figure 3-c). These last notions, intrinsic to Arabic cursive handwriting, are important in the modeling of Arabic as script having its own characteristics.

3. Segmentation

The connected components of the word are first extracted and then preclassified into two categories: tracings and diacritics [12]. Only the tracings are segmented, by a method based on the extraction and analysis of the signal describing the general word shape. This method is described in [13], and takes in consideration the Arabic handwritten calligraphic characteristics to reduce sub-segmentation cases due to ligatures.

A	B	C	A	B	C	A	B	C
a	ا	ا *	h	ط	ط ظ	o	ھ	ھ
b	ب	ب ت ث	i	ع	ع غ	p	و	و *
c	ح	ح ج خ	j	ف	ف	q	ى	ى ي
d	د	د * ذ *	k	ك	ك	r	ء	ء
e	ر	ر * ز *	l	ل	ل	s	ى	ق
f	س	س ش	m	م	م			
g	ص	ص ض	n	ں	ن			

- A: code of character sub-set ;
- B : the main body of characters of each sub-set
- C : characters belonging to each sub-set .

Figure 1: The Arabic alphabet: the characters are grouped by dominant shapes (main body); six characters (marked with a "*") can never be attached to their successors.

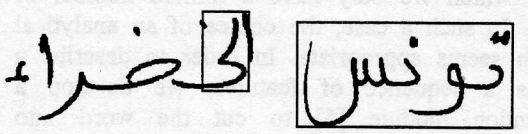
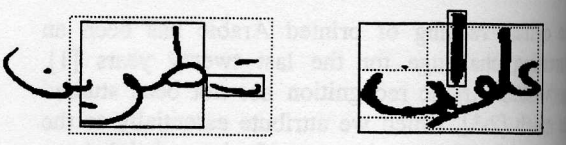


Figure 2 : Arabic handwritten text sample: the left box contains a ligature, and the right box contains a word composed of two pseudo-words.



(a) (b)

Figure 3 : Differences between the notions of pseudo-word and tracing: (a) the first pseudo-word on the right is composed of two tracings; (b) the first tracing is composed of two attached pseudo-words

The components obtained after the segmentation phase are called graphemes. A grapheme is defined as an elementary information zone that can be: a character, a portion of a character (over-segmentation) or a set of characters (under-segmentation).

In cursive Arabic handwriting, a pseudo-word is a set of characters or portions of a character (graphemes) attached with links. These links reflect the writing style and increase the variability of grapheme shapes. Each segmented grapheme is composed of a main body and of small links (Figure 4-b). In order to stabilize the grapheme's morphologic representation, we filter out the links. In this way, the grapheme is restricted to the component representing its main body (Figure 4-a).

We have studied statistically the limits of under-segmentation and over-segmentation of this module (very important for future recognition phases). We used a database tagged in graphemes of 4720 Tunisian City names. This study proved that 98% of the graphemes were composed of a component smaller than or equal to a character; 2% of the remaining graphemes contained at most two characters. A second statistical study shows that only 0.3% of the characters "*SIN*" and "*CHIN*" are segmented in four portions. The other characters are segmented at most in three portions. From these statistics, we formulate the following hypotheses:

- ***Hypothesis 1:*** a character can be segmented in a maximum of three parts (*limit of over-segmentation*);
- ***Hypothesis 2:*** a grapheme can contain at most two characters (*limit of under-segmentation*).

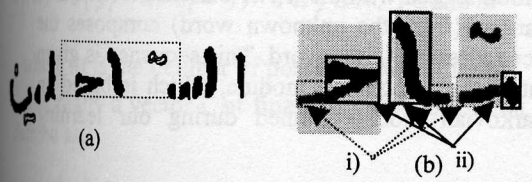


Figure 4: Filtering of links and detection of grapheme's main body: (a) example of graphemes after segmentation (b) zoom on the boxed area i) links ii) main body of graphemes.

In our approach, only the information extracted from the tracings zone is used to describe the word. We don't take into consideration the extracted diacritics because they are unstable (they generally have a small shape which can be easily damaged by the acquisition phase), and they increase the variability of the handwriting (due to the personality of the writer). However this approach decreases the quantity of information about the unknown shape (word) in the image.

4. Graphemes

4.1 Description of graphemes

The transfer of a grapheme (a shape) into a Markov Model observation is an essential phase. At this level, each grapheme is represented by a vector of the different measurements used by the classifier to compare an unknown grapheme to known ones. The classifier will base itself on one measurement to assign the grapheme to a class (the nearest one). A characteristic vector composed of different types of features can optimize the performance of a classifier [14]. In our application, the graphemes are described by two families of characteristic [15]. Each family is called a sub-vector. The first sub-vector is composed of structural features corresponding to human observations: closed loops, open loops, ascenders, descenders, normalized size; [16-17] report the informative and discriminating power of this type of visual features. The second sub-vector is composed of the first 10 Fourier Descriptors [18-19]. Finally, each grapheme is represented by a vector of 19 normalized characteristics. Let us remember also that these characteristics don't take into account the diacritics.

4.2 Choice of observation alphabet

In our study we distinguish between two notions: physical and logical entities. In this section, the physical entities are the graphemes, and the logical entities are denoted observations.

We study the distribution of graphemes (physical entities) over the observation alphabet (logical entities). A first distribution is done manually, while a second one uses an automatic clustering algorithm.

In the first method, we manually tag a set of graphemes, based on their visual shapes (the observations). Our tagged database contains approximately 45,000 graphemes distributed over more then 100 different classes; the number of classes is not fixed in advance. The number of samples in each class varies from five (in the case of an irregular or exotic grapheme) to some thousands.

In order to have statistically and informatively representative observation classes, we must reduce the number of these classes. The final set of observations participating in word description is chosen based on the amount of information that these observations bring to the character identification. The entropy criterion is based on the mutual information between each class of

observation and the alphabet of the characters with respect to the sequences observed in the training set.

To compute this mutual information, we propose to transform each sequence describing a segmented character into a binary sequence indicating the presence or the absence of a given observation k in each possible position in the character description. For example, take the following sequence representing a character from the database:

a/x/f/

Given that we are studying the grapheme "x", the sequence generated would be the following:

(0,1,0)

A code is attributed to each generated sequence and denoted s_j^k.

The mutual information between the alphabet of characters Ω and the set of sequence codes $\{s_j^k\}$ generated by an observation k can be written as:

$$MI(\Omega,k) = \sum_{m_i \in \Omega} \sum_{s_j^k} P(m_i, s_j^k) \log \frac{P(m_i, s_j^k)}{P(m_i) P(s_j^k)}$$

where m_i describes all characters of the alphabet Ω.

In this study, we make the assumption that our observations are independent of each other. Consequently, the mutual information obtained does not reflect the correlation between the different graphemes. This is a weakness of such a method.

The entropy criterion gives different classes of observations ordered by an increasing degree of informative power. In order to obtain the final alphabet, we select the most informative observation's classes. Each of the remaining graphemes (not selected) is attributed to the nearest class, using a k nearest neighbors classifier (k-NN).

To obtain an automatic system at this level, we have implemented a second k-means-clustering algorithm to decide, from their spatial distribution, the assigning of graphemes to the different classes of observation. The grapheme distribution is a priori unknown but it is based on distance criteria. The only manual intervention from the supervisor, at this level, is choosing the number of observation classes.

4.3 Observations vs. Graphemes

The decision on the class given to each segmented grapheme is made using a k-NN classifier. After the extraction of the characteristics, the grapheme is represented by an input vector for the k-NN. This phase can be considered as vector quantization (VQ), which transforms a sequence of graphemes (physical entities) into a sequence of HMM observations (logical entities). The value of k is chosen based on our best experimental results. The variation of results based on the value of k is so small that it is insignificant; nevertheless a value of k $\in$ {5,6,7} is relatively optimal.

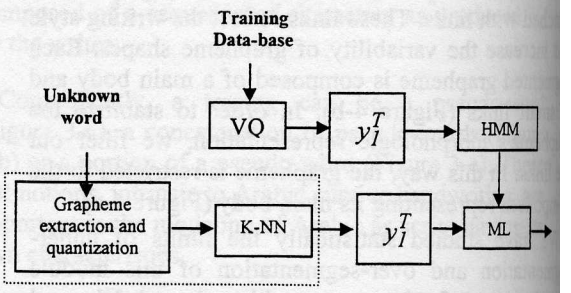


Figure 5: Block scheme of the training and recognition processes.

Figure 5 presents two processes: training and recognition. During training, the words segmented into graphemes represent, after the VQ phase, the sequences of observations which will be used for the estimation of the different parameters of the models. The classes of the observation alphabet are selected either by entropy criterion or by clustering. During recognition, the problem is different: the features extracted from the segmented graphemes are compared (Euclidean distance) with the various reference vectors (representations of graphemes from the training set). Each of these graphemes is transformed into an observation tagged with the k-NN classifier. The set of observations (from the unknown word) composes the sequence representing the word. This sequence is given as input to the recognition module, which is based on the Markovian models trained during our learning phase.

5. Markovian Modeling

A Markovian process is a stochastic chain $q_1^t = (q_1, q_2, q_3, \ldots, q_t)$ (finite number of states) which verifies the Markov property:

$$\forall t \quad P(q_t = i / q_{t-1} = j, q_{t-2} = k, \ldots) = P(q_t = i / q_{t-1} = j)$$

Any stochastic discrete process is a Markov process of order k if it verifies the extended Markov property:

$$\forall t \quad P(q_t / q_1^{t-1}) = P(q_t / q_{t-k}^{t-1})$$

If this process is composed of a finite set of N different states $i \in \lfloor 1, N \rfloor$, we then have a *Markov chain.*

A Markov Model is completely described by the number of states N and the following parameters A and π:

1- The transition probability matrix $A = [a_{ij}]$

where a_{ij} is the probability of being in state j at time t while knowing that we were in state i at time $t-1$:

$$a_{ij} = P(q_t = j / q_{t-1} = i) \quad 1 \le i \le N,\ 1 \le j \le N$$

2- The state initial probability vector Π

where $\pi_i = P(q_1 = i) \quad 1 \le i \le N$

the following condition has to be respected:

$$\forall i \sum_{j=1}^{N} a_{ij} = 1 \text{ .and } \sum_{j=1}^{N} \pi_j = 1$$

A Markov Model λ with parameters A and Π is noted as: $\lambda = (A, \Pi)$.

Contrary to a MM, the states of an HMM cannot be observed directly. We observe symbols given by the states as the system evolves in time. These symbols are called observations. In an HMM, the observation is a probability of the hidden state. Only the symbols given by the model are acceptable, and from them we can evaluate the most probable states. An HMM describes a double stochastic process.

In addition to the parameters A and Π, an HMM is described by matrix B of observation probabilities of the symbols following the model states: $B = [b_j(k)]$ where $b_j(k)$ is the probability that state j generates the symbol k ($k \in [1, M]$);

$$b_j(k) = P(y_t = k / q_t = j) \quad 1 \le j \le N;\ 1 \le k \le M$$

where M is the number of possible observations.

Eventually a vector Γ of final state probability can be defined as:

$$\Gamma = [\gamma_i] \text{ where } \gamma_i = P(q_T = i) \quad 1 \le i \le N$$

An HMM λ is totally defined by the parameter A, B and Π; we can add the final probability vector Γ to reinforce the decisions. An HMM λ is noted as:

$$\lambda = (A, B, \Pi, \Gamma)$$

A HMM is a soft elastic model. It tolerates variability in the writing, and adapts perfectly to our type of data. The only problem is that we have a relatively large lexicon ($N_\omega = 232$ classes of words) and very few samples per word class. To overcome this problem, we are modeling less complex components that are better represented in the database than the word component.

The main subject of modeling is the character component. Let us remember also that the word is a sequence of pseudo-words, which we are taking into account in our model to get a better modeling of Arabic handwriting.

In the following section, we discuss the modeling chosen for Arabic cursive handwriting. This modeling works from three hierarchical levels of abstraction: the character, the pseudo-word and the word which represents the entity to be recognized.

5.1 Character Level Modeling

The modeling process is directly linked to the segmentation phase. The model developed has to take into consideration all the possible segments of a character. This segmentation does not depend on the character as a logical entity, but rather depends on its shape. Based on this, we go from a model per character to a model per family of characters (Figure 2); this reduces the number of models from approximately 30 to only 18 (after the elimination of diacritics and especially dots). This choice has advantages as well as disadvantages. The main advantage is the increase in the number of sample considered in the estimate of the model parameters. The disadvantage is in the degradation of the refinement of the word description. With the hypotheses given in section 1.1, the character model becomes a right-left 3-state model. The transitions between states correspond to the set of graphemes which can be generated from the segmentation of a given character (Figure 4). The states of this model are called μ-states. The transition probabilities between the different states are estimated on the entire database and increase considerably the number of samples per characters (models). At this level we estimate two main probabilities Pr_1 and Pr_2; the other transition probabilities being directly deducible (Figure 6). The transitions are of order 1.

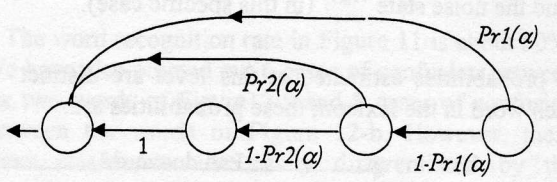


Figure 6: character model (α).

5.2 Pseudo-word Level Modeling

A pseudo-word is an ordered succession of characters (see section 1). The pseudo-word model can be viewed as a concatenation of the character models composing the word (Figure 7). The segmentation phase depends

essentially on the set of bi-grams. The segmentation engine behaves the same way each time it encounters the same bi-gram in a pseudo-word. The parameter of interest at this level is the probability of segmentation by bi-grams. This probability can be computed over an entire database.

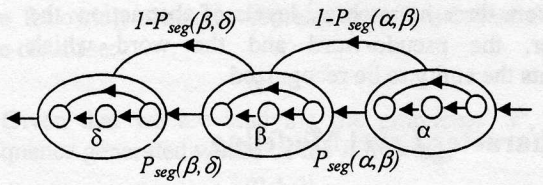


Figure 7: Example of model for a 3-character pseudo-word (αβδ): $P_{seg}(\alpha,\beta)$ is the probability of segmenting the bi-gram (α,β), while "$1-P_{seg}(\alpha,\beta)$" represents the probability that the two characters α and β will be joined in the same grapheme.

This model must take into consideration two fundamental problems: touching pseudo-words, and confusing diacritics taken as pseudo-words, because they were included in the sequence of pseudo-words. These diacritics are represented by a noise state "*" inserted between two pseudo-words, but could also appear at the beginning or end of the word (Figure 8).

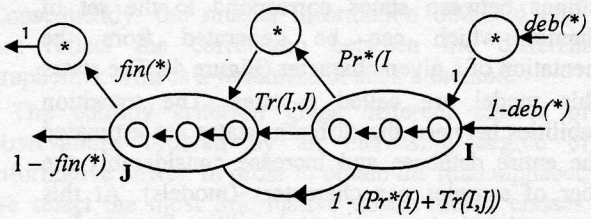


Figure 8: The transition model between the pseudo-words of the sequence composed of I, J and the noise state ''*'' (in this specific case).

The probabilities estimated at this level are distinct for each word in the lexicon; these probabilities are:

i) deb(*) is the probability that the first element of the sequence of the pseudo word is a diacritic; ii) fin(*) is the probability that the sequence ends with a diacritic; iii) Pr*(I) is the probability that pseudo-word I is followed by a diacritic; vi) Tr(I,J) is the probability that pseudo-word I is followed by pseudo-word J and that there is at least one segmentation point between them.

5.3 The Word Model

The final model is a fusion of the three models defined earlier. Its states are all the combinations of the μ-states (from character models), encountered while tagging the training set. During training, we compute all transition probabilities defined above and for each of the three hierarchical models. The estimation of the different parameters of the word final model is done as follows:

The transition probability matrix A: we assume the terms of matrix A to be a_{ij} (transition between two consecutive states i and j of the final model) the product of two entity probabilities [20]: i) $a_{inter}(i,j)$ is the transition probability between the last μ-state of state q_i and the first μ-state of state q_j; ii) $a_{intra}(j)$ is the production probability of state q_j as it was observed. Figure 9 illustrates by an example the computation of these two probabilities. Therefore:

$$a_{ij} = a_{inter}(i,j).\ a_{intra}(j)$$

The probability vectors of the initial states π_i and final states γ_i are estimated as follows:

$$\pi_i = \begin{cases} deb(*) & \text{if i is state ''*''} \\ (1 - deb(*)).a_{intra}(i) & \text{otherwise} \end{cases}$$

$$\gamma_i = \begin{cases} fin(*) & \text{if i is state ''*''} \\ 1 - fin(*) & \text{otherwise} \end{cases}$$

The symbols (observations) are directly produced by the word final states. The matrix B of produced symbols is simply estimated statistically on the entire training set.

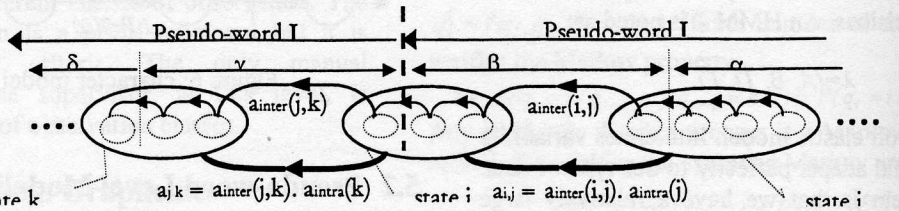


Figure 9: An example of computation of transition probabilities: *i)* $a_{inter}(i,j) = 1-Pr_1(\beta)$ *and* $a_{inter}(j,k) = (1-Pr_1(\gamma))$ *; ii)* $a_{intra}(j) = (Pr_2(\beta))\ (1-(Pr^*(I)+Tr(I,J)))$ *and* $a_{intra}(k) = (Pr_2(\gamma))\ (1-P_{seg}(\gamma,\delta))$

5.4 Decision

The classifier is a maximum likelihood (ML) module (figure 10). This classifier takes the word as being a sequence of observations $y_1^T = (y_1, y_2, ..., y_T)$. For each candidate word class i, this classifier computes $P(y_1^T / \lambda_i)$ which corresponds to the probability of getting the sequence of observations y_1^T by module λ_i. These probabilities ($P(y_1^T / \lambda_i)$) are evaluated by the *Baum-Welch* algorithm. Finally, the word is attached to the class k for which the model λ_k maximizes the emission probability y_1^T :

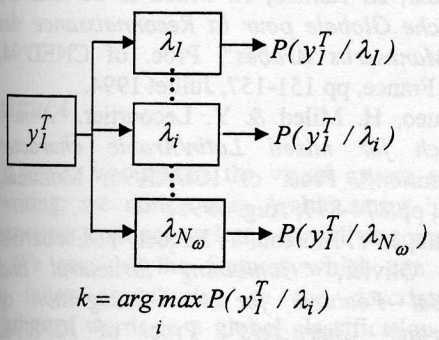


Figure 10 : The ML based word classifier: N_ω is the size of the lexicon

6. Results

Our lexicon counts 232 different word classes. Each class is written by approximately forty-five writers. We are training the models on 4720 tagged words, and the system performance is evaluated on a test set of 5900 words. The results are very promising in both alphabets. The first results have shown that the performance varies with respect to the size of the alphabet [15]. We decided to use a fixed size of forty observation classes. For the alphabet selected by entropy criteria, we choose the more informative graphemes (see section 2.2) while, in the automatic case, only the desired number of classes was fixed. The performances obtained were comparable, but the first alphabet still shows slightly better results (Table 1). When comparing performances from the two alphabets, we notice that the difference in the result increases considerably at the fifth rank to reach approximately 2.25%.

	top 1	top 2	top 3	top 4	top 5
MI	**79.5%**	**86.9%**	**89.6%**	**91.1%**	**92.2%**
K-means	79.1%	86.4%	88.2%	89.3%	89.9%

Table 1: Performance of the system, using two types of observation alphabet.

The analysis of the main recognition errors (confusions) shows two principal causes:

The first is the less trained words, because they are constituted by either rare characters or complex shape characters (Figure 11).

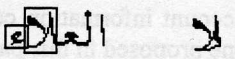
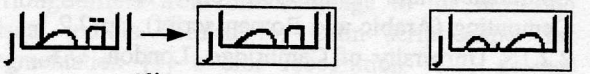


Figure 11: Rare or complex shape characters

The second is the confusion between word classes which have the same handwritten main shape, and which can be distinguished by diacritics (Figure12).



Kitar Samar

(a)

Sidi Aïche

Sidi Hassine

(b)

Figure 12: The most common confusion cases in our data base due to similar main shapes of the words in each case.

The word recognition rate in Figure 11 is about 50%. We have also pointed out 6 cases of confusion between the two words of Figure12-a and 8 cases of confusion between the words of Figure 12-b. However, these word classes can be easily differentiated by the diacritics information.

7. Conclusion

In this paper we first introduced the notion of pseudo-word. The second part of the paper discusses the segmentation and extraction of graphemes, as well as their vector quantization. The graphemes obtained are composed of either portions of characters or a set of touching characters. We adopted HMM for the

management of sequences of segmented graphemes. This modeling was the subject of the third section. We presented an analytical approach of the words with three hierarchical levels merged into each model. The training of the various parameters of models and the basic principles of the word classifier have also been presented and discussed. The results obtained justify the choice of observation alphabet obtained by entropy criterion over the clustering technique. The analysis of the confusion cases proves the importance of the diacritics to improve the recognition rates and the quality of decision of our classifier.

We are trying to coordinate this approach with a global word recognition module which takes into consideration the diacritics as visual indexes. The latter module takes into account information complementary to that used by the one proposed in this paper.

References

[1] P. Ahmed, M. A. A. Khan, *"Computer recognition of Arabic script based text: the state of the art."*, Proc. of the 4th International Conference and Exhibition on Multi-lingual Computing (Arabic and Roman script), pp 2.2.1-2.2.15, University of Cambridge, London, U.K., April 1994.

[2] A. Amin, *"Off Line Arabic Character Recognition- A survey"*, Proc. of ICDAR'97, Ulm, Germany, pp 441-445, Aug. 1997.

[3] B. El-Badr & S. A. Mahmoud, *"A Survey and Bibliographie of Arabic optical text recognition"*, Signal Processing, vol. 41, pp 49-76, 1995.

[4] A. Amin, H. B. Al-Sadoun, *"A New Segmentation Technique of Arabic Text"*, Proc. of ICPR'92, pp. 441-445, Sept. 1992.

[5] C. Olivier, H. Miled, K. Romeo, Y. Lecourtier, *"Segmentation and Coding of Arabic Handwritten Words"*, Proc. of ICPR'96, Vienna, Autria, vol. 3, pp 264-268, Aug. 1996.

[6] L. R. Rabiner, *"A tutorial on hidden Markov models and selected applications in speech recognition"*, Proc. of IEEE, vol. 77, no. 2, pp 257-285, 1989.

[7] J. Nouza, *"Feature selection method for Hidden Markov Model based speech recognition"*, Proc. of ICPR'96, Vienna, Austria, vol. 2, pp 186-190, Aug. 1996.

[8] H. Bunke, M. Roth & E. G. Schukat-Talamazzini, *"Off-line cursive handwriting recognition using Hidden Markov Models"*, Pattern Recognition, vol. 28, no. 9, pp 1399-1413, 1995.

[9] N. Ben Amara & A. Belaid, *"Printed PAW Recognition Based on Planar Hidden Markov Models"*, Proc. of ICPR'96, Vienna, Autria, vol. 2, pp 220-224, Aug. 1996.

[10] A. El Yacoubi, *"Modèlisation Markovianne de l'écriture manuscrite: Application à la reconnaissanc des adresses postales"*, Thèse de Doctorat de l'université de Rennes 1, 1996.

[11] C. Olivier, T. Paquet, M. Avila & Y. Lecourtier, *"Optimal Order of Markov Models applied to Bankchecks"*, International Journal of Pattern Recognition and Artificial Intelligence, vol. 11, no. 5, pp 789-800, 1997.

[12] A. Ameur, K. Romeo, H. Miled & M. Cheriet, *"Approche Globale pour la Reconnaissance des Mots Manuscrits Arabes"*, Proc. of CNED'94, Rouen, France, pp 151-157, Juillet 1994.

[13] K. Romeo, H. Miled & Y. Lecourtier. *"A new approach for mixed Latin/Arabic character segmentation"*. Proc. of ICDAR'95, Montréal, Canada, pp 874-877, Aug. 1995.

[14] L. Heutte, J. V. Moreau, T. Paquet, Y. Lecourtier & C. Olivier, *"Combining Structural and Statistical Features for the Recognition of Handwritten Characters"*, Proc. of ICPR'96, Vienna, Autria, vol. 2, pp 210-214, Aug. 1996.

[15] H. Miled, C. Olivier, M. Cheriet & Y. Lecourtier, *"Coupling Observation/Lettre for a Markovian Modelisation Applied to the Recognition of Arabic Handwriting"*, Proc. of ICDAR'97, Ulm, Germany, vol. 2, pp 580-583, Aug. 1997.

[16] M. Cheriet & C. Y. Suen, *"Extraction of key letters for cursive script recognition"*, Pattern Recognition Letters, vol. 14, n. 12, pp 1009-1017, 1993.

[17] M. Côté, E. Lecolinet, M. Cheriet & C.Y. Suen, *"Building a Perception Based Model for Reading Cursive Script"*, Proc. of ICDAR'95, Montréal, Canada, vol. 1, pp 893-901, Aug. 1995.

[18] S. A. Mahmoud, *"Arabic Character Recognition Using Fourier Descriptors and Character Contour Encoding"*, Pattern Recognition, vol. 27, no. 6, pp 815-224, 1994.

[19] F. P. Kuhl & C. R. Giardina, *"Elliptic Fourier features of a closed contour"*, Computer Graphics and Image Processing, vol. 20, pp 236-258,. 1981.

[20] M. Y. Chen, A. Kundu & J. Zhou, *"Off-line handwritten word recognition using an hidden Markov model type stochastic network"*, IEEE PAMI, vol. 16, no. 5, pp 481-496, 1994