Structural Features Extraction for Handwritten Arabic Personal Names Recognition

Afef KACEM, Nadia AOUÏTI
LATICE-ESSTT
University of Tunis, Tunisia
Afef.kacem@esstt.rnu.tn, Nadia.aouiti@gmail.com

Abdel BELAÏD
LORIA-CNRS
Nancy, France
abdel.belaid@loria.fr

Abstract—**Due to the nature of handwriting with high degree of variability and imprecision, obtaining features that represent words is a difficult task. In this research, a features extraction method for handwritten Arabic word recognition is investigated. Its major goal is to maximize the recognition rate with the least amount of elements. This method incorporates many characteristics of handwritten characters based on structural information (loops, stems, legs, diacritics). Experiments are performed on Arabic personal names extracted from registers of the national Tunisian archive and on some Tunisian city names of IFN-ENIT database. The obtained results presented are encouraging and open other perspectives in the domain of the features and classifiers selection of Arabic Handwritten word recognition.**

Keywords-component; feature extraction; Arabic handwritten recognition; personal names

I. INTRODUCTION

Handwriting recognition still lacks a good recognition rate since it depends much on the writer and because we do not always write the same word in exactly the same way. Because of the huge variability of the handwriting style and the noise affecting the data, it is almost impossible to directly recognize handwritten word from its bitmap representation. Thus, the need of features extraction method that allows extracting a feature set from the word image is obvious for classification. In fact, features extraction is a preprocessing step that aims at reducing the dimension of the data while extracting relevant information. In this step, each word is represented as a feature vector, which becomes its identity. These features, as mentioned by [1], must be reliable, independent, small in number, and reduce redundancy in the word image.

Features extraction methods are based on two types of features: statistical and structural. Major statistical features, used for word representation, are derived from distribution of points: zoning, projections and profiles, crossing and distances. Words can be represented by structural features with high tolerance to distortions and style variations. This type of representation may also encode some knowledge about word structure or may provide some knowledge as to what sort of components make up that word. Structural features are based on topological and geometrical properties of the word, such as aspect ratio, cross points, loops, branch points, strokes and their directions, inflection between two points, horizontal curves at top or bottom, etc.

Due to the high variability in unconstrained handwritten script words, obtaining these features is a difficult task. To achieve acceptable results, the context has to be restricted by a given lexicon of all possible words. This paper describes a features extraction method based on structural features and explores the use of these features in case of handwritten Arabic personal names recognition. The outline of the paper is as follows. In section II, we explore a number of features extraction methods in use in the field of Arabic handwriting recognition. In section III, we describe the used lexicon. In section IV, we propose a features extraction method that captures characteristics such as loops, legs, stems and diacritics in the script. In section V, we give an overview of the obtained results. We, finally draw, in section VI, a conclusion with some outlooks.

II. RELATED WORKS

In the field of Arabic Handwritten Recognition, some advances have been accomplished during the last years. Observing Arabian manuscripts reveal the complexity of the task, especially for the features choice (discontinuity of the writing, multiple connections of sub word, complex ligatures, etc.) [2]. This leads to particularize the environment (restriction of the vocabulary and the number of writers), and imposes the cooperation of several types of features in order to reduce the complexity level [3, 4].

As previously mentioned, commonly used features in Arabic handwriting recognition are structural or statistical. Structural features are intuitive aspects of writing, such as loops, branch-points, end-points and dots. Statistical features are numerical measures computed over images or regions of images. They include but not limited to pixel densities, histogram of chain code directions, moments and Fourier descriptors.

Among the different type of features, [5] adopted global structural features: - The number of connected components, of descenders, of ascenders, of unique dot below the baseline, of unique dot above the baseline, of two dots

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.276

below the baseline, of two dots bound above the baseline, of 3 bound dots, of Hamzas (zigzags), of Loop, of tsnine (by calculation of number of intersection in the middle of the median zone) and Concavity features with the four configurations. They also used statistical features: the density measures or "zoning". Two subdivisions of the word image are applied. For each zone, two statistical measures that are the density of black pixels and the variance are calculated. Parameters such as lower and upper baselines are used, in [6] to derive a subset of baseline dependent features. Thus, word variability due to lower and upper parts of words is better taken into account. In addition, the proposed system learns character models without character pre-segmentation. Experiments have been conducted on the benchmark IFN/ENIT database of Tunisian handwritten country/village names.

Notice that many Arabic letters, pieces of words or even words share common primary shapes, differing only in the number of dots, and whether the dots are above or below the primary shape, structural features are a natural method for capturing dot information explicitly, which is required to differentiate such letters, words or parts of words. This perspective may be a reason that structural features remain more common for the recognition of Arabic script than for that of Latin scripts. This paper proposes the extraction of structural features for the recognition of handwritten Arabic personal names.

III. LEXICON DESCRIPTION

We have been restricted by the lexicon of personal names from count registers of Tunisian national archives. These registers are old, noisy and high degraded documents. They consist of rows; each of them is composed of a list of personal names. Rows are of different length. Due to the writing style, horizontal and/or vertical ligatures are easily introduced between parts of words and attachment occurred between the words of successive rows. Registers are written by a single author who used line support, images of multiple instances of the same word are likely to look similar. This reduces the amount of handwriting variations that have to be compensated for. Notice that, for some letters, the shape changes (see Figure 1(a) and (b) for the letter يand (c) and (d) for the letter د).

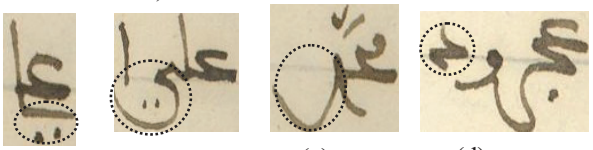


Figure 1. Same letters with different shapes.

Other letters are written in a tilted way so they can be easily confused with others letters such the letters 'ا' and 'ل' with the letter 'ر' as illustrated in Figure 2.

Figure 2. Tiled letters confused with other letters.

Registers are written in Arabic which is cursive: the letters are joined together along a writing line. Due to the style of the writing, vertical and/or horizontal ligatures are easily introduced between the parts of words (See Figure 3).

Figure 3. Horizontal and vertical ligatures.

Discontinuity can be seen between letters of the same word or inside letter itself as shown in Figure 4

Figure 4. Letter discontinuity.

These historical registers are also written using old scripts. So, for some letters, the number and/or position of their diacritic points were changed. For example, the Arabic letter 'ق' is written with a single diacritic point above the letter body instead of two points and the letter 'ف' is written with a single diacritic point below the letter body (see Figure 5).

Figure 5. Letters: 'ي', 'ف' and 'ق' in old Arabic script.

In Arabic, diacritics are essential to differentiate certain letters. But, sometimes, diacritic points can be merged into one component so two or three diacritic points can be presented by the same way (see Figure 6).

Figure 6. Confusion in number of diacritic points.

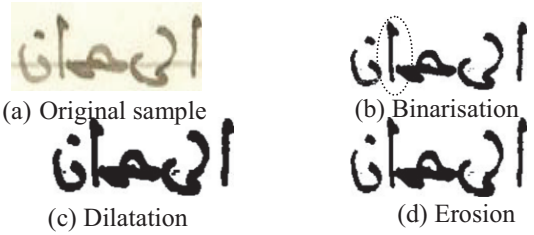
Some diacritic points are displaced (see Figure 7(a)) and others are confused with small letters (see Figure 7(b), letter 'ة' and 7(c), letter 'ي').

(a) قلعية (b) حميدة (c) الزكري

Figure 7. Diacritic points vs small letters.

IV. Proposed Feature Extraction Method

The preliminary task is to do pre-processing since words tend to be highly degraded as they are taken from historical documents under many imperfections and noise. It mainly considers gray to binary conversion, noise removal and smoothing. In Figure 8, binary conversion is followed by dilatation and erosion.



(a) Original sample (b) Binarisation

(c) Dilatation (d) Erosion

Figure 8. Word pre-processing.

Besides pre-processing, recognition system is based on how words are represented. In this work, structural features are extracted to represent patterns.

As features extraction method is tightly related to the adopted segmentation approach and knowing that segmentation is a difficult problem in handwritten word recognition due to the high variability, especially when dealing with semi cursive scripts as Arabic, we proceeded without any word segmentation. It is about to detect presence of letters without delimiting them and thus have a global vision of words while avoiding segmentation problems. To this end, some global and structural features are extracted considering their positions in the word (at the beginning, in the middle or at the end of the word, in the upper, central or lower bands of the word) as shown in Figure 9.

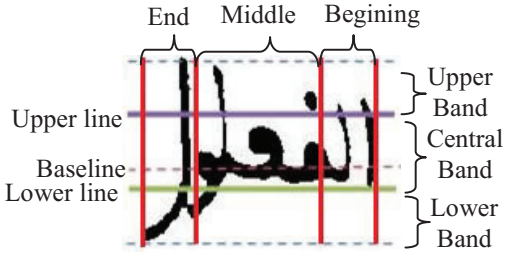


Figure 9. Possible positions of the extracted features

Words are partitioned into three bands: the upper, the central and the lower bands after baseline location. Baseline is quite tricky to locate especially in case of Arabic script which is, in contrast to the Latin script, has not major accumulation of black pixels in a line. This is mainly due to letter extensions or horizontal ligatures. As horizontal projection histogram was not helpful (see Figure 10) to locate baseline, we referred to line support, used by the author to write words (see Figure 11).

Figure 10. Failure of the horizontal projection for baseline location.

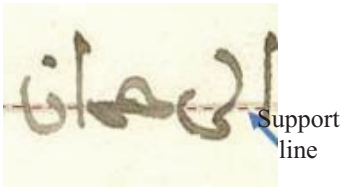


Figure 11. Base line location using support line.

The central band is delimited by the upper and the lower lines. These lines are located using the baseline which divides word image into inferior and superior parts. For these parts, we respectively compute the upper and the lower bands. 50% of the superior part and 30% of the inferior part are respectively considered for the upper and the lower bands because 1) letter stems are generally higher than their legs and 2) words are written by a single author and we note that the height of the letters, without stems, does not exceed 50% of the upper band of the word image.

Afterwards, connected components are respectively extracted from the upper, the lower and the central bands to locate letter stems, legs, diacritics and loops. Dividing then words into three zones, from right to left, serves to classify extracted features according their position in the word: in the beginning (the first quarter), in the middle (the second and the third quarters since Arabic word is generally elongated in the middle) and at the end (the last quarter) of the word.

Word description is then performed from right to left as a sequence of structural features gathered from each band. Next, we will explain how to extract loops, stems, legs and diacritics and how to distinguish between different shapes of stems, legs and diacritics.

A. Loops

To find loops, the system extracts connected components of the entire mirror image of the word (see Figure 12).

Figure 12. Loop extraction of the name اللطيف

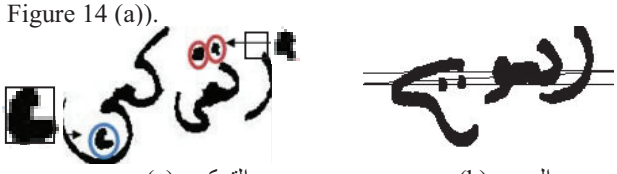
Due to the writing style, false loops can be detected and others can be disconnected or mouthfuls (see Figure 13).

(a) أخوهما (b) بن عماموا

Figure 13. False disconnected and mouthful loops.

B. *Diacritics*

Diacritics may occur in the upper and/or the lower bands of words, at the beginning, middle and/or the end of words. The number of diacritic varies from one to three points. Diacritics do not cross the baseline and they are reduced in area (i.e. width*height) and have high density (i.e. number of black pixels/area). The number of diacritics depends on the aspect ratio of their connected components (i.e. width/height) because two or three diacritic points can be attached and then considered as only diacritic point (see Figure 14 (a)).



(a) التركي (b) السويح

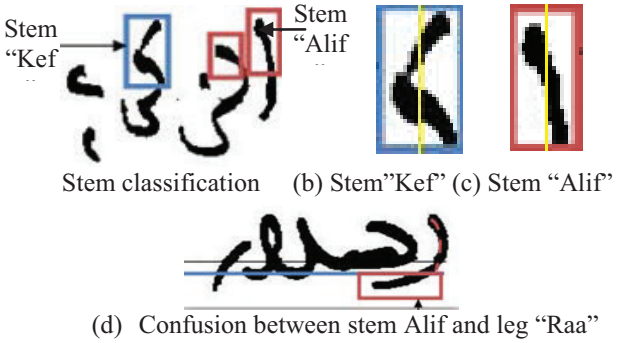
Figure 14. Diacritic extraction.

Algorithm *Diacritic_Extraction*

1. Define a range of inclination *R* (see Figure 14(b)).
2. Extract connected components.
3. For each component $C \notin R$,
   if *C* do not cross baseline then return *diacritic*.
4. For each component $C \in R$,
   if *Area(C)* ≤ threshold then return *diacritic*.

C. *Stems and Legs*

Stems and legs are respectively located in the upper and lower bands of words. Stems can be of two types: “stem_alif” (ا) and “stem_kef” (ک) while legs can be classified as “leg_noun” (ن), “leg_raa” (ر), “leg_haa” (ح). Stems and legs classification is based on aspects ratio, density of their connected compounds and the number of their contact points with the upper and the lower lines of the central band (see Figures 15 and 16).



Stem classification (b) Stem”Kef” (c) Stem “Alif”

(d) Confusion between stem Alif and leg “Raa”

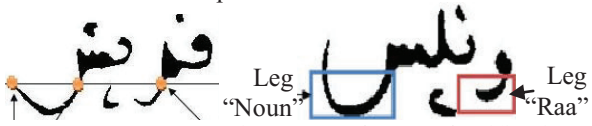
Figure 15. Stem extraction.

Algorithm *Stem_Extraction*

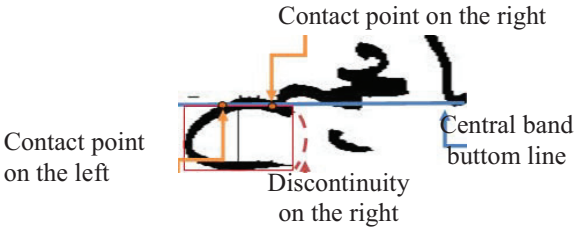
1. Extract connected components in the upper band.
2. For each component compute
   Ratio(C)=high(C)/width(C).
3. if Ratio(C)>1
   then compute number of run length pixels nbr-rlp
   if nbr_rlp<4 then return *stem alif* (see Figure15(c))
   else return *stem kef* (see Figure 15(b))
   else return *stem kef*

As the letter "Alif" (ا) exceeds sometimes the lower line, it can be easily confused with the letter “Raa” (ر). To avoid such confusion, the system goes back the pixels and checks if they are attached to pixels of a component classified stem "Alif" in which case, leg “Raa” is not considered as illustrated in Figure 15(d).

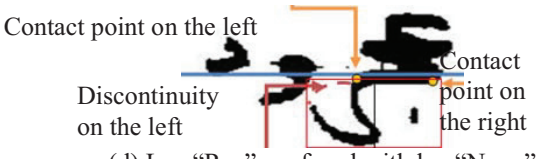
As shown, in Figure 16, legs “Raa” or “Haa” can be confused with leg “Noun” as both of them cross the lower line twice. To distinguish between them, the system checks black pixel discontinuity on the right and the left sides of their connected components.



(a) Contact points (b) Leg classification



(a) Leg “Haa” confused with leg “Noun”



(d) Leg “Raa” confused with leg “Noun”

Figure 16. Leg extraction.

Algorithm *Leg_Extraction*

1. Extract connected components in the lower band.
2. For each connected component *C* compute number of contact points *nbr-contact* with the lower line.
   if *nbr-contact*=1 then compute the position of the contact point according the middle of *C*.
   if *position*=right then return *leg “Raa”*
   else return *leg “Haa”*
   else compute number of run length pixels *nbr-rlp*
   if *nbr_rlp*≤3 then return *stem “Noun”*
   else compute pixel discontinuity

if *discontinuity*=right
then return *leg "Haa"*
else return *leg "Raa"*

Table I summarizes the extracted structural features and classifies them based on their positions in the word.

TABLE I. EXTRACTED STATISTICAL FEATURES

Description	Code
Loop at the Beginning of the word	LB
Loop in the Middle of the word	LM
Loop in the End of the word	LE
One diacritic Point Up at the Beginning	1PUB
Two or three Points Up at the Beginning	2PUB
One diacritic Point Up in the Middle	1PUM
Two or three Points Up in the Middle	2PUM
One diacritic Point Up at the End	1PUE
Two or three Points Up at the End	2PUE
One diacritic Point Down at the Beginning	1PDB
Two or three Points Down at the Beginning	2PDB
One diacritic Point Down in the Middle	1PDM
Two or three Points Down in the Middle	2PDM
One diacritic Point Down at the End	1PDE
Two or three Points Down at the End	2PDE
Stem "Alif" at the Beginning	SAB
Stem "Alif" in the Middle	SAM
Stem "Alif" at the End	SAE
Stem "Kef" in the Beginning	SKB
Stem "Kef" in the Middle	SKM
Stem "Kef" the End	SKE
Leg "Noun" at the Beginning	LNB
Leg "Noun" in the Middle	LNM
Leg "Noun" at the End	LNE
Leg "Raa" at the Beginning	LRB
Leg "Raa" in the Middle	LRM
Leg "Raa" at the End	LRE
Leg "Haa" at the Beginning	LHB
Leg "Haa" in the Middle	LHM
Leg "Raa" at the End	LHE

Figure 17 illustrates an example of word description. The extracted features of the name: الصقر are as follows: SAB, SAB, LM, 1PUM, LNE.

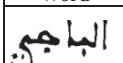
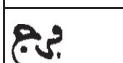
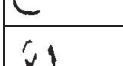
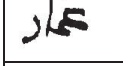
Figure 17. Features extraction for the name: الصقر

V. EXPERIMENTS

To evaluate features extraction results, we compute the *Levenshtein* distance, or edit distance, which is a string metric for measuring the amount of difference between two sequences. This distance is defined as the minimum number of edits needed to transform one sequence into the other, with the allowable edit operations being insertion (case of feature extracted in superfluous), deletion (case of not extracted feature), or substitution (case of not correctly extracted feature) of a single feature.

In Table II, *E, T* and *D* respectively refer to sequences of extracted features and truth description features and the *Levenshtein* distance.

TABLE II. EXAMPLES OF FEATURES EXTRACTION RESULTS

Word	E	T	D
	SAM, SAB, SAB, LM, LRE, 1PDE, 1PDE, 1PDM, 1PDB	SAM, SAB, SAB, LM, LRE, 1PDE, 1PDE, 1PDM, 1PDB	0
	LE, LHE, LNB, 1PDM, 1PDB	LE, LHE, LNB, 1PDM, 1PDB	0
	SAM, 1PUB, LHE	SAM, 1PUB, LHE	0
	SAM, 2PUE, 2PUE, 1PDM	SAM, 2PUE, 2PUE, 2PDM	1
	SAM, LRM, LHM, 2PUB, 2PUB	SAM, LRM, 2PUB, 2PUB	0
	SAM, LRE	SAM, LRE	0
	SKM, SKB, LRM, LRB, 2PUM, 2PDM, 1PUB	LRM, LRB, 2PUM, 2PDM, 1PUB	2
	SAB, LM, LNM, LRM, 1PUB	SAB, LM, LNM, LRM, 1PUB	0
	SAM, LBC, LRM, 2PDE, 2PUE, 2PDM	LBC, LRM, 2PDE, 2PUE, 2PDM	0
	SAM, SAB, LNE, 2PDM, 1PDM, 1PDB	SAM, SAB, LNE, 2PDM, 1PDM, 1PDB	0
	SAE, LNE, 1PUE, 1PDM, 2PUB	SAE, LNE, 1PUE, 1PDM, 2PUB	0

	SAM, LHE, 2PUB	SAM, LHE, 2PUB	0
	LRE, LRM, LRM	LRE, LRM, LRM	0

Notice that for the name 'عطية', although only one diacritic point was extracted, instead of two, but it was located in the right position. For the name 'قريش', wanted features are correctly extracted but wrongly stems were detected in superfluous. Most of features extraction errors can be attributed to the writing style and the poor quality of some data samples. Table III displays evaluation results of structural feature extraction using two databases: personal names, extracted from registers of the national archive of Tunisia, and Tunisian city names from the public database IFN-ENIT.

TABLE III. EXTRACTION EVALUATION RESULTS

Data test	Average Recall	Average Precision	F-Measure
Personal names (116)	0.89	0.89	0.89
IFN-ENIT (534)	0.78	0.82	0.80

As shown, in Table III, despite the difficulties specific to the Arabic and ancient handwriting of personal names, the extraction results are better than those concerning Tunisian city names of the IFN-ENIT database. This can be explained by large morphological variations of the handwriting as Tunisian city names are written by several people while registers, we deal with, are written by the same person.

VI. CONCLUSION AND FUTURE WORKS

The processing of Arabic handwritten writing, with its morphology problems, imposes a cooperation of several types of primitives according to the big variability of the word shapes. Among the different type of features, we adopted structural features. In this paper, the following global structural features are detailed: legs, stems, loops and diacritics considering their number, types and positions in the word. A feature set made to feed a classifier can be a mixture of such features. To reduce the size of the feature set, feature subset selection can be applied on the extracted features. In fact, the performance of a classier can rely as much on the quality of the features as on the classifier itself. A good set of features should represent characteristics that are particular for one class and be as invariant as possible to changes within this class. As future work, extracted structural features will be tested on the database IFN-ENIT then they will be processed by an HMM for Handwritten Arabic Personal Names Recognition.

REFERENCES

[1] A. Benouareth, A. Ennaji, and M. Sellami, "Arabic Handwritten Word Recognition using HMMs with Explicit State Duration," EURASIP Journal on advances in signal processing ISSN 1687-6172 2008, vol. 2008, no12, pp. 1-13

[2] A. Amin, "Off-line Arabic character recognition: The state of the art," Pattern Recognition. 31 (1998), pp. 517-530.

[3] N. Ben amara, and A. Belaid, "Printed PAW Recognition based on planar Hidden Marcov Model," Proc. International Confernce on Pattern Recognition (ICPR 98), pp. 220-226.

[4] L. M. Lorigo and V. Govindaraju, Pattern Analysis and Machine Intelligence. IEEE Transactions on May 2006, (5), pp. 712-724.

[5] N. Azizi, N. Farah, M. Tarek Khadir, and M. Sellami, "Arabic Handwritten Word Recognition Using Classifiers Selection and features Extraction/Selection," Recent Advances in Intelligent Information Systems, ISBN 978-83-60434-59-8: , pp. 735–742.

[6] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, "Arabic handwriting recognition using baseline dependant features and hidden Markov modeling," Proc. International Confernce on Document Analysis and recognition (ICDAR 05), pp. 893 – 897.