International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:4, 2012

Unconstrained Arabic Online Handwritten Words Segmentation using New HMM State Design

Randa Ibrahim Elanwar, Mohsen Rashwan, Samia Mashali

***Abstract*—**In this paper we propose a segmentation system for unconstrained Arabic online handwriting. An essential problem addressed by analytical-based word recognition system. The system is composed of two-stages the first is a newly special designed hidden Markov model (HMM) and the second is a rules based stage. In our system, handwritten words are broken up into characters by simultaneous segmentation-recognition using HMMs of unique design trained using online features most of which are novel. The HMM output characters boundaries represent the proposed segmentation points (PSP) which are then validated by rules-based post stage without any contextual information help to solve different segmentation errors. The HMM has been designed and tested using a self collected dataset (OHASD) [1]. Most errors cases are cured and remarkable segmentation enhancement is achieved. Very promising word and character segmentation rates are obtained regarding the unconstrained Arabic handwriting difficulty and not using context help.

***Keywords*—**Arabic, Hidden Markov Models, online handwriting, word segmentation

I. Introduction

HANDWRITTEN words recognition is one of the research areas having a lot of open issues. Handwritten words recognition is considered as much more challenging problem rather than printed word recognition. This can be attributed to the huge variability of handwritings among writers which make the problem much complicated especially if the help of natural language resources is absent. Natural language resources themselves like public datasets, lexica, language models, etc. are still not available for some languages or some problems. For example, handwritten datasets for Latin are much more available and intense rather than those for Arabic. Also, handwritten datasets for offline recognition problem are much more available rather than those for online recognition problem. This is due to the earlier beginning and continuity of those researches that motivated researchers to build and provide such resources. The language characteristics also can hold back achieving significant results in the recognition problem solution, for example, diacritics presence and cursiveness of Arabic leaving the generalization issue open for upcoming researches. A word recognition algorithm attempts to associate the word image to choices in a lexicon. Typically, a ranking is produced. This is done either by the analytic approach of recognizing the individual characters or by the holistic approach of dealing with the entire word image.

Randa I. Elanwar is Assistant Researcher in computers and systems dept., Electronic Research Institute, Cairo, Egypt (phone: 202-33310515; e-mail: eng_r_i_elanwar@yahoo.com).

Mohsen A. Rashwan is Professor of Digital Signal Processing, Electronics and communication dept., Cairo University, Cairo, Egypt (e-mail: Mohsen_Rashwan@rdi-eg.com).

Samia A. Mashali is Professor of Digital Signal Processing, Electronic Research Institute, Cairo, Egypt (e-mail: samia@eri.sci.eg).

The latter approach is useful in the case of touching printed characters and handwriting. A higher level of performance is observed by combining the results of both approaches [2].

Unlike analytical methods, holistic methods are constrained to applications with a small lexicon size as in bank check processing systems where the lexicons do not have more than 30–40 entries. For unconstrained word recognition, the analytical approach is preferred with the help of contextual information.

In an analytic approach, the segmentation of words into segments that relate to characters is required. Nevertheless, this is not a trivial task due to problems such as touching, overlapping, or broken characters. Moreover, this operation is made more difficult because of the ambiguity encountered in handwritten words. Therefore, most successful analytical methods employ segmentation-based recognition strategies where the segmentation can be explicit or implicit.

Segmentation based approaches try to segment a given word into smaller entities. However, as it is extremely difficult, if not impossible, to segment a given word into its individual characters without knowing the word's identity, they usually split a word into entities that don't necessarily correspond to exactly one character each, and they consider a number of possible segmentation alternatives at the same time. Typically, an oversegmentation of the given input word is attempted. That is, the image of a character that occurs within a word may be broken into several constituents, also called graphemes. At the same time the segmentation procedure avoids merging two adjacent characters, or parts of two adjacent characters, into the same constituent. A large number of heuristics for achieving such kind of segmentation have been reported in the literature [3].

An advantage of segmentation based word recognition schemes is that the problem is reduced to isolated character recognition - a problem for which a number of quite mature algorithms have become available. On the other hand, segmentation and grapheme recombination are both based on heuristic rules that are derived by human intuition. The development of automatic procedures that are able to learn segmentation rules from training data and automatically infer the parameters that guide the search for fitting the optimal character hypotheses is still an open problem [3].

One approach for segmentation is by proposing a high number of segmentation points, offering in this way several segmentation options, the best ones to be validated using heuristic rules. This strategy may produce correctly segmented, undersegmented, or oversegmented characters. A lot of researchers followed this approach. Kavallieratou et al. [4], have developed a rules-based system for offline handwriting segmentation of Greek and English words. The possible segmentation points are extracted under certain rules

generated using Transformation-based learning (TBL). De Stefano et al. [5], segment online handwriting into elementary strokes. The method is based upon filtering the Discrete Fourier Transform (DFT) of the sequences x(n) and y(n) at different resolution and building a saliency map from the reconstructed sequences by the Inverse Discrete Fourier Transform (IDFT) for every scale. The map records significant curvature variations on the original curve. Another research by De Stefano et al. [6], decompose the online handwritten English words into shape primitives. Then, ligature detection is done by selecting the regions of the word that have horizontal density histogram zones which count 1. Segmentation points are located at the intersections between the ligature and the word middle-line and the baseline. Abdulla et al. [7], have presented a rules-based system for offline Arabic handwritten word segmentation where the image upper contour information is kept. The contour pixels are then divided into segments of which slope is calculated to find the writing direction changes '+' or '-'. These segments are combined to form bigger decisive segments (DS) according to certain rules which are searched to find appropriate feasible segmentation points (FSP) according to another set of rules. Kherallah et al. [8], have developed a simultaneous handwriting segmentation-recognition system for online Arabic handwritten words based on Freeman codes similarity. Handwritten scripts are segmented into simple strokes and represented as a super-position of time shifted versions of beta-elliptic models characterized by three parameters. 8-directional Freeman chain codes are extracted and matched using Euclidian distance calculation for recognition.

Another approach for segmentation followed by other researchers is to also to propose a high number of segmentation points and validate them by feeding feature vectors representing the segmented parts to some classifier (especially neural network 'NN') rather than using heuristic rules. Kurniawan et al. [9], have developed a word segmentation system for offline English words using contour analysis to locate segmentation points in cursive handwriting then combine a feed forward NN to validate them. Rehman Khan et al. [10], have used rules-based method to locate segmentation points in cursive offline handwritten English words, then, combining a feed forward NN for validation. Cavalin et al. [11], present an implicit segmentation-based method for recognition of offline English words through a two-stage hidden markov model (HMM) recognizer. The first HMM stage is a Segmentation-Recognition (SR) module and gives the N best segmentation-recognition paths. The verification stage re-ranks the N best segmentation-recognition paths by re-classifying the segmented characters using a powerful HMM isolated character recognizer.

As shown above, most researchers working on the segmentation problem solely with human expert evaluation rather than recognition, have used limited datasets of their own despite the availability of large public datasets like UNIPEN [12], IAMonDB [13], ADAB [14] (on-line) and CEDAR [15], NIST [16], IFN/ENIT [17] and IAM database [18] (off-line) and those who used public databases didn't benefit from it all, they used only 1000 to 2000 words for training and 300 to 400 words for test.

Kurniawan et al. [9], have used 1000 words of IAM database (6417 patterns of valid and invalid points) are used for training, 317 words (1902 segmentation points) are used for test. Rehman Khan et al. [10], have used training data consists of 2678 words (25072 patterns) and test data consists of 2936 patterns. Cavalin et al.'s [11] experiments are carried out using 18,624 unconstrained word images available in the IAM database, distributed as follows: 12651 for training, 3168 for validation and 2805 for testing. For De Stefano et al. [5], 1,000 words produced by the same writer provided by the Handwriting Recognition group at IBM T.J. Watson Research Center, were used. While in De Stefano et al. [6], a data set of 1600 words produced by 100 different writers is used. Abdulla et al. [7] have conducted their experiments on the IFN/INIT database and AHD/AUST database (self collected dataset containing 12300 Arabic handwritten words by 82 different writers). Kherallah et al. [8] have used 34500 words of ADAB database. 20000 words are used as data prototypes, the others are used for testing.

The evaluation result in the latter case is measured the word recognition rate (WRR). For the former systems solving the word segmentation problem without the presence of classifiers, human experts are usually asked to perform the classification for evaluation. The evaluation result is measured either by the word segmentation rate (WSR) or the segmentation points recognition rate (SPRR), also defined as character segmentation rate 'CSR'). Kurniawan et al. [9], have achieved recognition rate 82.63% (SPRR) for valid identification of 1,902 pattern of segmentation point. The neuro rule-based segmentation algorithm by Rehman Khan et al. [10] has achieved recognition rate of 91.21% for valid identification of 2936 segmentation points (SPRR). The top-1 word recognition results (WRR) achieved by Cavalin et al. [11] 97.4%, 93.9%, 86% and 78% for 10, 100, 1000 and 3717 word lexicon sizes respectively. For Kavallieratou et al. [4], experiments held on 500 English and Greek words by 250 writers (2:1 training-test ratio) gave 77.8% accuracy (WSR). De Stefano et al. [5] produce correct decomposition in 99.53% of the words. While in De Stefano et al. [6], an average correct segmentation of almost 68% over the 26 character classes (CSR) is achieved. Abdulla et al. [7] have got 90.58% and 95.66% word segmentation accuracy (WSR) for the IFN/INIT database and AHD/AUST database respectively.

In this paper we proposed a HMM-based word segmentation system for unconstrained Arabic online handwriting. Our system follows the analytical approach where words are broken down into characters by the segmentation-recognition HMM. The HMM proposed segmentation points are then validated using rules-based post stage without any contextual information help. The evaluation of the segmentation performance is done using human expert. Thus we have used a self collected dataset that we have presented in previous work [1]: the OHASD dataset, the first online handwritten Arabic sentence dataset. The dataset is

unconstrained, natural, simple and clear. Texts are sampled from daily newspapers and are dictated to writers using tablet PCs for data collection. The current version includes 154 paragraphs written by 48 writers. It contains 670 text lines, more than 3800 words and more than 19,400 characters. We divided the dataset to 110 documents for training, 14 for validation and 30 for test. The results achieved are very promising regarding the fact that no contextual information is used.

The paper is organized as follows: Section 2 gives a description of the word segmentation system. Experiments and results are presented in Section 3, and Section 4 draws some conclusions and proposes future work.

II. System Description

The system we propose is composed of several units. The first is a pre-processing unit where the input word strokes undergo the main preprocessing operations of smoothing, re-sampling, and normalization. The second unit is the complementary strokes removal (section A) where dots, hamza and other secondary (delayed) strokes are identified and filtered out from the main word strokes based on heuristic rules. The third unit is the feature extraction unit where local and vicinity features are computed for a window of samples moving in the samples writing order direction. These features are discussed in details in section B. Frames made up from the extracted features are passed to pre trained HMM to simultaneously segment and recognize the characters they represent. The HMM proposes characters with their boundaries are considered as segmentation points on the input word strokes. These points are passed to a validation post stage where different rules are applied in specific order to relocate their position on the word strokes for error reduction and segmentation enhancement. More about these rules is given in section C. Finally, secondary strokes are reassigned to their corresponding main character bodies. The details of each of the system units are given in the following sections:

A. Secondary Strokes Removal

Secondary strokes removal is an essential step in the online handwriting case, and especially for Arabic, because writers first cursively write the Arabic word then randomly add the secondary strokes to the main character bodies. That is why they are also called delayed strokes as they are not written in order with the main character body. Consequently, at the feature extraction stage, where features are extracted in the writing order of strokes, the features extracted from the secondary strokes will be piled successively at the end of the feature matrix rather than next to their main character feature vectors which is so confusing for classifiers.

Secondary strokes removal is achieved in two stages. The first filters out the significantly small size strokes like single dots. The second one filters out the relatively large strokes like Hamza, Madda, Delayed-Alef, Kaf-hat and stuck dots after baseline rough estimation. Geometric features from all the word strokes are computed to act as reference values to compare and decide.

An example of of the pre-processing and dot removal procedures for the word يحال can be seen in Fig. 1.

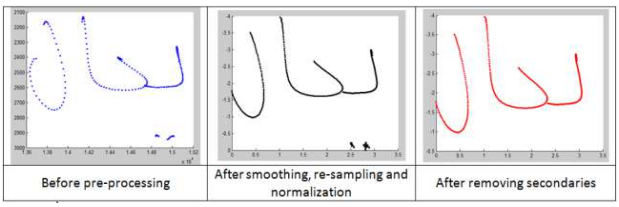


Fig. 1 Word shape before and after pre-processing

B. Features

The third unit in our proposed system is the feature extraction unit. We used new features together with other features found in literature. Features are computed for a window of samples moving in the samples writing order direction. These features have two types: Local and vicinity features. Local features are those computed for one sample relating it to another sample, whereas, vicinity features are those representing all samples within the window. Abdelazeem et al. [19] summarized features found in literature as:

1. Delta X and Y: The relative change of each sample's (P_t) x-value and y-value with the following sample which is represented with $\Delta x(t)$ and $\Delta y(t)$ as shown in Fig. 2.

2. Writing Direction: It describes the local writing direction using the cosine and sin of $\alpha(t)$, where $\alpha(t)$ is the angle between the line connecting the previous sample P_{t-1} and the next sample P_{t+1} and the positive direction of the x-axis as shown in Fig. 2.

3. Chain Code: An 8-direction chain code is used to quantize the change in direction between each two pair of consecutive samples along the trajectory as in Fig. 3.

4. Angle: The angle $\theta(t)$ between each two samples on the trajectory in radians, as shown in Fig. 4.

5. Aspect: It characterizes the height-to-width ratio of the bounding box of the vicinity of P_t as shown in Fig. 4. It's represented with A(t), where:

$$A(t) = \frac{\Delta y^*(t) - \Delta x^*(t)}{\Delta y^*(t) + \Delta x^*(t)} \qquad (1)$$

6. Curliness: Curliness C(t) is a feature that describes the deviation from a straight line in the vicinity of P_t, where:

$$C(t) = \frac{L(t)}{\max(\Delta x^*(t), \Delta y^*(t))} - 1 \qquad (2)$$

and L(t) is the length of the trajectory in the vicinity of P_t, i.e., the summation of lengths of all inter-sample line segments that are shown Fig. 4.

7. Slope: The slope S(t) of the straight line joining the first and last point in the vicinity of Pt as shown in Fig. 4, where:

$$S(t) = \tan\theta^*(t) \qquad (3)$$

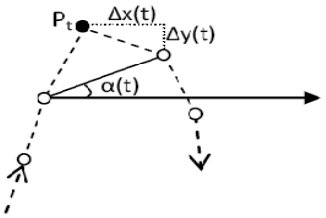


Fig. 2 Delta X and Y, and Writing Direction Features

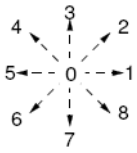


Fig. 3 Freeman directional chain code

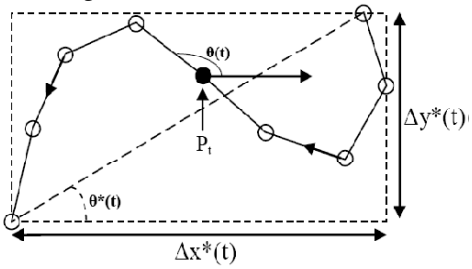


Fig. 4 Feature extraction in vicinity of sample P_t

We propose new local and vicinity features motivated by our will to use offline features without generating a bitmap for the handwritten word, thus, avoiding any error may be caused by this process.

The first new feature is local feature called the EYE feature, a three component directional feature: Moving Eye, PAW Fixed Eye, Word Fixed eye. Two types of directional features are suggested: instantaneous and relative. The instantaneous feature is, the same described above in Fig. 3 as 'writing direction', as if, an eye is tracing the word strokes sample-by-sample in their writing order and finding the next direction with respect to the current sample, we call it (Moving Eye). The relative feature is, as if, an eye is tracing the word strokes in their writing order, sample-by-sample in their writing order, and find the next direction with respect to one fixed sample: the very first sample written on the first stroke in the word, word head, we call it (Word Fixed eye) or the first sample on each written part of Arabic word, PAW head, we call it (PAW Fixed Eye).

The advantage of using both instantaneous and relative features is that: the moving eye represent the dynamic writing changes of the word (role of online feature) whereas the fixed eye represent the writing changes with respect to one or more fixed points preserving relative locations (role of offline feature).

The EYE feature has two representations: one using sin and cosine the direction angle, the other uses the polar representation, length and angle value in radians, of the direction angle, we call it Polar-EYE.

1. Moving eye: the cosine and sin of the angle between the line connecting P_{t-1} and P_t and the positive direction of the x-axis.

2. PAW fixed eye: the cosine and sin of the angle between the line connecting P_t and the stroke head and the positive direction of the x-axis.

3. Word fixed eye: the cosine and sin of the angle between the line connecting P_t and the word head and the positive direction of the x-axis.

4. Polar Moving Eye: the length of the line connecting P_t and P_{t-1}, and the angle it forms with the positive direction of the x-axis in radians.

5. Polar PAW fixed Eye: the length of the line connecting P_t and stroke head, and the angle it forms with the positive direction of the x-axis in radians.

6. Polar Word fixed Eye: the length of the line connecting Pt and the word head, and the angle it forms with the positive direction of the x-axis in radians.

The second new feature is vicinity feature called Chords angles, where the cosine and sin of the angles between the parallel chords connecting samples within a window and the positive direction of the x-axis as in Fig. 5.

And the Chords curviness feature, the ratio between the chord length and inter-sample distance sum between its two ends for all parallel chords as in Fig. 6.

$$F= [\frac{c3}{d3+d4}, \frac{c2}{d2+d3+d4+d5}, \frac{c1}{d1+d2+d3+d4+d5+d6}] \quad (4)$$

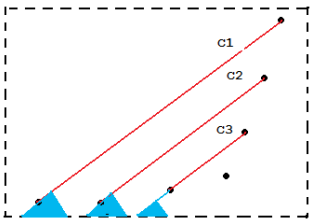


Fig. 5 Parallel chords angles features

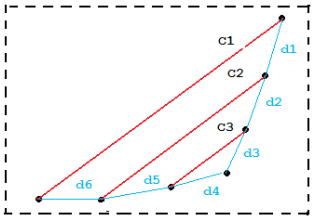


Fig. 6 Chords curviness feature

The best features are computed for each sample within a fixed size window moving in the direction of writing in the sample order. These features are used to build up feature frames fed to the next system unit, HMM classifier.

Hidden markov models designed and trained using frames of best features, are used to simultaneously segment and recognize the test word feature frames to their corresponding characters. We have HMMs for 68 unique models representing 28 Arabic characters (reduced to 19 after removing dots) in different positions together with 6 ligatures: Lam-Meem, Lam-Alef, Nabra-Hah, Meem-Hah, Nabra-Meem (لم،لا،لح،بح،مح،بم). System parameters: (1) HMM number of

states per model, (2) HMM number of Gaussian mixtures per state, (3) number of samples per window (window size), and (4) window overlapping are optimized using validation data set. The experimental result section details the optimization procedures and gives the best HMM structure used. The HMM output characters boundaries represent the proposed segmentation points (PSP) on the word strokes. These points are forwarded to next unit (rules-based validation post stage) for error reduction and segmentation improvement.

C. PSP Validation Stage

The last unit in our system is a multi-stage rule-based post stage that functions with the concept of relating each word segment to its predecessor and successor to validate the position of the segmentation point.

This role of this stage is mostly adjusting the segmentation points locations more than eliminating them because these points proposed by HMM are much smarter than those proposed by heuristics used in literature. Eight different rules are applied in specific order to solve the segmentation errors (bad segmentation, under segmentation and over segmentation):

Rule 1: Shifting PSP lying very closely end of a stroke to the stroke last sample, then either translate or eliminate the next PSP according to the next word segment size.

Rule 2: Eliminating PSP to merge very small size and closely located word segments on the same stroke.

Rule 3: Eliminating more than one PSP on low slope word segments (dashes).

Rule 4: If a PSP will cause 2 word segments bounding boxes to intersect, either translating the PSP back or forth to prevent intersection, or eliminating the PSP according to the next word segment size.

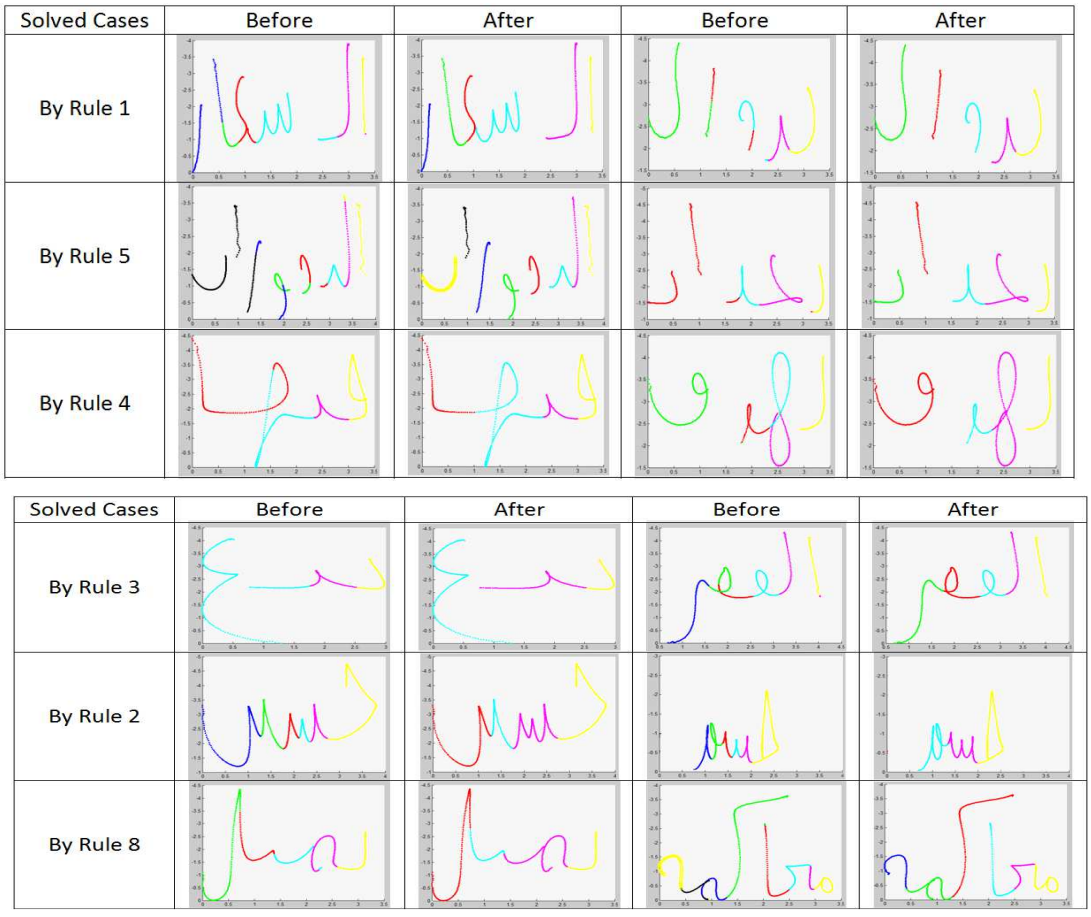
Rule 5: Adding PSP to a multi-stroke word segment having large non overlaps on x-axis between these strokes.

Rule 6: Shifting or eliminating PSP to merge vertically overlapping word segments on the same stroke and on nearby strokes (e.g. Kaf-hats).

Rule 7: Minor shifting of PSP location to the nearest valley on the stroke.

Rule 8: Shifting or eliminating PSP to merge touching strokes.

The sequence of applying these rules is: Rule 1, Rule 5, Rule 4, Rule 3, Rule 2, Rule 8, Rule 7, and finally Rule 6. Examples of segmentation errors corrected by applying the above rules can be seen in Fig. 7. The effect of these rules application on the segmentation result is detailed in the next section.



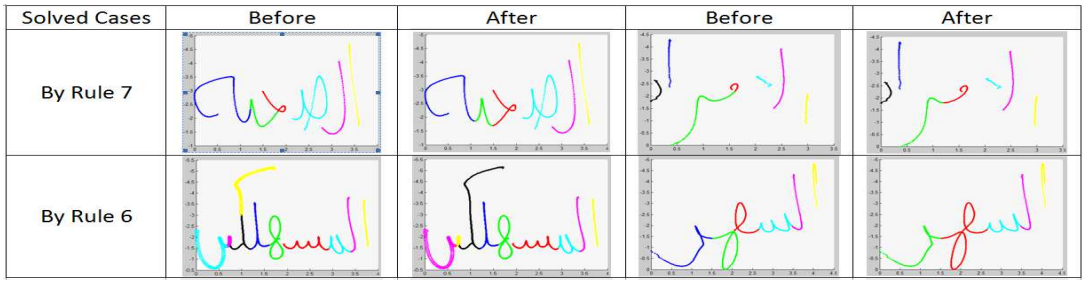


Fig. 7 Examples of segmentation errors corrected by the validation rules

III. Experimental Results

Experiments are conducted using the OHASD dataset, a self collected dataset presented in previous work [1], it includes 154 paragraphs written by 48 writers. It contains 670 text lines, more than 3800 words and more than 19,400 characters. The dataset is divided 110 documents for training, 14 for validation and 30 for test. These dataset divisions have 2802, 334 and 688 words respectively. We chose the human expert evaluation approach since our basic concern is the segmentation task rather than the recognition. Initial experiments are first established using about one third of the validation dataset (98 words) to first select best feature types and system parameters values optimization. Secondly the whole validation dataset is used for validation stage rules design.

TABLE I
The Feature Set Search Effect on HMM Results

Feat.	WRR	WSR	WUSR	WOSR	WBSR	CSR	CRR
F1	2.04	10.20	3.06	84.69	2.04	57.34	53.05
F2	4.08	10.20	1.02	83.67	5.10	60.05	58.01
F3	10.2	19.39	0	77.55	3.06	65.46	67.27
F4	3.06	8.16	3.06	85.71	3.06	52.37	63.21
F5	8.16	18.37	0	79.59	2.04	68.17	70.43
F6	9.18	15.31	4.08	78.57	2.04	70.88	70.20
F7	8.16	18.37	0	78.57	3.06	69.07	70.65
F8	8.16	15.31	0	82.65	2.04	70.20	66.14
F9	10.2	14.29	0	76.53	9.18	73.36	74.14
F10	10.2	18.37	0	77.55	4.08	72.23	71.78

F1: EYE
F2: EYE & Polar EYE
F3: EYE, Polar EYE, Chords angles
F4: EYE, Polar EYE, Chords angles & Chords curviness
F5: EYE, Polar EYE, Chords angles and Delta X,Y
F6: EYE, Polar EYE, Chords angles & Aspect
F7: EYE, Polar EYE, Chords angles & Curliness
F8: EYE, Polar EYE, Chords angles & Chain code
F9: EYE, Polar EYE, Chords angles, Aspect, Curliness & Chain code
F10: EYE, Polar EYE, Chords angles, Aspect & Curliness

Features are searched forwardly to find the best features combination as shown in table 1 where WRR is the word recognition rate, WSR is the word segmentation rate, WUSR is the word under segmentation rate, WOSR is the word over segmentation rate, WBSR is the word bad segmentation rate, CSR is the character segmentation rate and CRR is the character recognition rate.

The set achieving these conditions appears to be F10 = {EYE, Polar EYE, Chords angles, Aspect, Curliness} feature set.

In the following experiments, the system parameters are optimized in sequence using the winner feature set. Experiments to optimize the window size parameters turned up that 9-samples window with no overlapping is the best to use. Keeping the best window parameters and varying the number of HMM states shows that 20 states with 12 Gaussian mixtures per HMM has the best overall result.

Experiments showed that increasing the state number is improving WSR, WRR, CSR and CRR but also increases the under- and bad- segmentation in a faster rate. The number of HMM Gaussian mixtures variation didn't affect the results remarkably as expected, thus, we have thought to introduce a new parameter to the system, which is the location of HMM Gaussian mixtures. In other words, instead of having a HMM with all its states having equal number of Gaussian mixtures, we define a new HMM with variable Gaussian mixtures number per state. We have tried to vary the location of states having multiple mixtures along the HMM. Experiments showed that the best location for multiple-mixture states is at the beginning of HMM. Experimentally we found that HMM having 16 mixtures only for the first 8 states and a single Gaussian for the rest of states is the best HMM structure to be used as shown in tables II and III.

TABLE II
Varying State Number with All States Having the Same Mixture Number

States No.	WRR	WSR	WUSR	WOSR	WBSR	CSR	CRR
8	6.12	16.33	0.00	83.67	2.04	72.23	67.04
10	9.18	21.43	0.00	76.53	2.04	72.91	70.65
12	11.22	21.43	3.06	73.47	2.04	75.62	71.11
14	12.24	26.53	5.10	67.35	1.02	77.20	71.78
16	13.27	31.63	3.06	63.27	2.04	79.23	72.91
18	15.31	30.61	4.08	61.22	4.08	80.36	71.78
20	16.33	37.76	4.08	55.10	3.06	80.36	69.30
22	18.37	35.71	3.06	56.12	5.10	79.68	72.91
24	15.31	26.53	5.10	51.02	17.35	70.88	72.69
26	18.37	26.53	5.10	53.06	15.31	70.65	71.33
28	19.39	32.65	6.12	50.00	11.22	71.78	74.49
30	16.33	31.63	5.10	51.02	12.24	69.30	73.36
32	18.37	32.65	6.12	46.94	14.29	69.30	74.72
34	21.43	35.71	5.10	45.92	13.27	72.23	74.72
36	23.47	36.73	8.16	43.88	11.22	75.40	73.81

TABLE III
VARYING STATE NUMBER WITH STATES HAVING VARIABLE MIXTURE NUMBER
(16 MIXTURES FOR THE FIRST 8 STATES)

States No.	WRR	WSR	WUSR	WOSR	WBSR	CSR	CRR
12	8.16	17.35	0.00	76.53	6.12	75.40	71.78
14	10.20	17.35	3.06	74.49	5.10	73.14	69.30
16	14.29	28.57	5.10	64.29	2.04	76.98	71.56
18	12.24	34.69	5.10	57.14	3.06	76.98	69.07
20	13.27	30.61	4.08	58.16	7.14	78.33	69.30
22	16.33	32.65	6.12	56.12	5.10	79.91	72.23
24	15.31	35.71	7.14	51.02	5.10	77.65	68.62
26	17.35	38.78	8.16	45.92	7.14	80.36	70.88
28	14.29	41.84	8.16	42.86	7.14	82.17	72.23
30	17.35	39.80	11.22	36.73	11.22	79.23	72.91
32	20.41	42.86	9.18	38.78	9.18	80.59	73.36
34	18.37	43.88	8.16	34.69	13.27	79.23	70.65
36	20.41	53.06	11.22	26.53	9.18	84.65	72.01
38	18.37	41.84	12.24	29.59	16.33	78.56	72.23
40	19.39	45.92	18.37	26.53	9.18	79.91	70.43

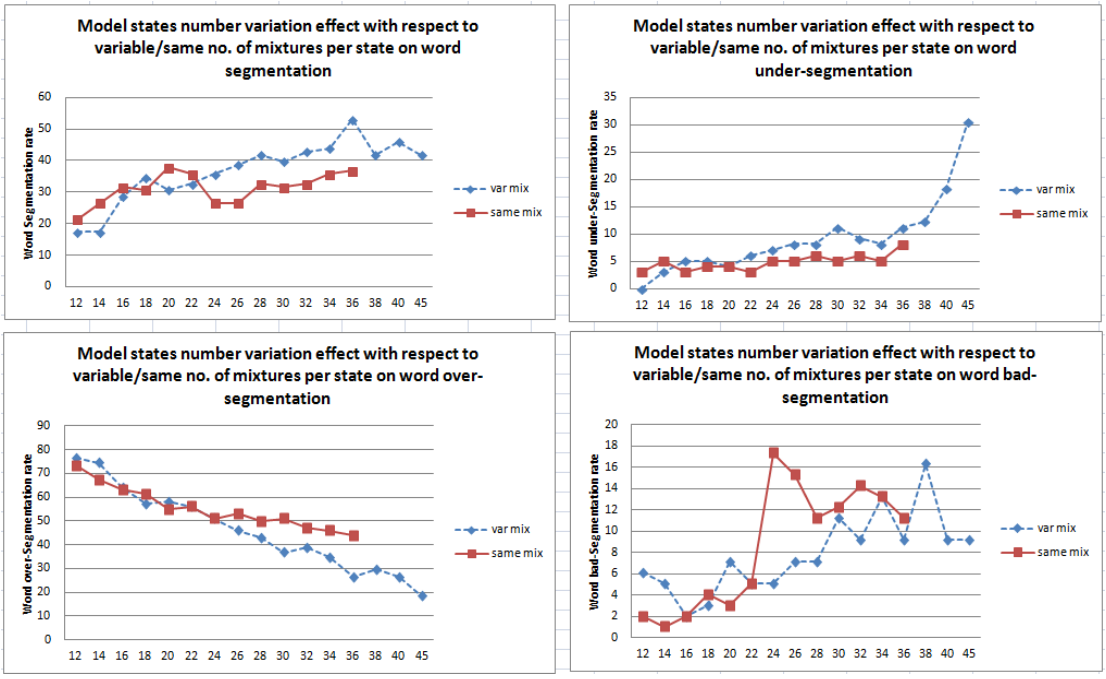


Fig. 8 The effect of location of variable-mixture states within the HMM

Moving to the PSP validation post stage, we have included the rest of validation data encounter more of the writers' variations for rules design. The effect of applying these rules is given in table IV. Unfortunately, we have limits of improvement. Actually, the words that could not be fixed are either: (1) Under-segmented words by HMM, (2) bad segmented words where PSP are located very far from their correct places, or(3) over/bad segmented words that turn into under-segmented word after PSP validation. The remaining step is the secondary strokes restoration stage. Spatial information are used to assign the secondary strokes to the main character having total or partial histogram overlap on x-axis or that having the nearest located boundaries. The results of this stage are given in the table 5 below.

TABLE IV
THE VALIDATION POST STAGE SEGMENTATION RESULTS

Symbol	WSR	WUSR	WOSR	WBSR	CSR
HMM reference result	46.41	18.56	23.95	11.08	80.75
R1	47.31	18.26	23.35	11.08	80.43
R1-R5	52.69	10.78	27.25	9.28	85.15
R1-R5-R4	56.29	11.98	23.05	8.68	87.19
R1-R5-R4-R3	57.78	11.98	22.46	7.78	87.89
R1-R5-R4-R3-R2	70.06	15.87	8.08	5.99	90.38
R1-R5-R4-R3-R2-R8	72.16	16.17	7.19	4.49	90.95
R1-R5-R4-R3-R2-R8-R7	73.95	17.07	6.59	2.40	91.01
R1-R5-R4-R3-R2-R8-R7-R6	78.44	17.96	1.80	1.80	91.97

TABLE V
UNITS FOR MAGNETIC PROPERTIES

		O1	O2	O3	O1	O2	O3
Writer No.	Words No.	WSR	WSR	WSR	CSR	CSR	CSR
1	51	49.02	76.47	72.55	82.77	93.31	90.12
2	99	44.44	83.84	56.57	81.58	94.07	81.10
3	40	52.50	85.00	60.00	84.62	92.82	81.31
4	46	32.61	65.22	45.65	69.70	87.01	73.82
5	76	47.37	77.63	60.53	81.50	91.33	81.87
6	22	63.64	77.27	31.82	87.38	91.26	69.44
Average	23	46.23	78.67	57.19	80.87	92.02	80.88

O1: HMM output
O2: PSP Validation stage output
O3: Dot restoration output

Unfortunately, spatial information was not enough to handle the severe secondary strokes location shift cases like those shown in the Fig. 9 below. The final system results on the test dataset are shown in table 6.

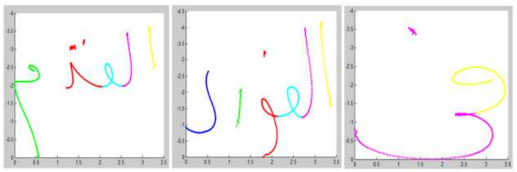


Fig. 9 Severe location shifts of secondary strokes leading to wrong assignment

TABLE VI
FINAL SYSTEM RESULTS ON TEST DATASET

	WSR	WUSR	WOSR	WBSR	CSR
HMM Output	37.05	21.61	21.61	19.73	72.46
After PSP validation	51.54	23.09	17.05	8.19	80.68
After dot restoration	36.64	18.66	28.72	15.84	71.35

IV. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a two stage word segmentation system for Arabic online handwriting based on HMM. Initially the handwritten word undergoes pre-processing and secondary strokes removal. The segmentation system first stage is HMM classifier trained using novel and common features. The HMM design is unique and has passed by several stages of improvement.

The system has been designed and tested using a self collected dataset (OHASD). Parameter optimization is done using validation dataset. The HMM segmentation-recognition procedure proposes segmentation points on the word strokes. These PSP are smarter than those that heuristic rules could propose.

HMM segmentation errors are mostly over-segmentation. Under- and bad- segmentation errors are fewer but most of them are incurable. The PSPs are validated by a rules-based post stage for segmentation enhancement and error reduction. Most errors have been cured and very promising results are obtained on the validation dataset.

Secondary strokes restoration has been done based on spatial information only which caused results deterioration due to severe location shifts between word characters and their corresponding secondary strokes.

As a future work we need to investigate further enhancement of the HMM classifier design through 3D optimization of states number, mixtures number and mixtures location in addition to grading the mixtures number (gradual increase or decrease of states' mixtures number along the HMM). Addition of contextual information like language model also may contribute remarkable segmentation-recognition result enhancement that can further be used for secondary strokes assignment on context base.

REFERENCES

[1] R. Elanwar, M. Rashwan, and S. Mashali, "OHASD: The first online Arabic sentence database handwritten on tablet PC", International Conference on Signal and Image Processing ICSIP 2010, Singapore, Proceedings of World Academy of Science, Engineering and Technology (WASET), vol. 72, pp.710-715, 2010.

[2] R. Plamondon, S. Srihari, "Online and Off-Line Handwriting Recognition, A Comprehensive Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 22, No.1, pp. 63-84, 2000.

[3] H. Bunke, "Recognition of Cursive Roman Handwriting - Past, Present and Future", Proceedings of the 7th International Conference on Document Analysis and Recognition, ICDAR'03, vol. 1, pp. 448-455, 2003.

[4] E. Kavallieratou, E. Stamatatos, N. Fakotakis, G. Kokkinakis, "Handwritten Character Segmentation Using Transformation-Based Learning", Proceedings of the 15th International Conference on Pattern Recognition ICPR, 2000, pp.634-637.

[5] C. De Stefano, M. Garruto, A. Marcelli, "A Multiresolution Approach to On-line Handwriting Segmentation and Feature Extraction", Proceedings of the 17th International Conference on Pattern Recognition, ICPR'04, vol. 2, pp. 614-617, 2004.

[6] C. De Stefano, A. Marcelli, "From Ligatures to Characters: A Shape-based Algorithm for Handwriting Segmentation", Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'02), pp. 473-478, 2002.

[7] S. Abdulla, A. Al-Nassiri, R. Abdul Salam, "Offline Arabic Handwritten Word Segmentation using rotational invariant segments features", International Arab Journal of Information Technology, vol. 5, no. 2, pp. 200-208, 2008.

[8] M. Kherallah, L. Haddad, A. Alimi, "A new Approach for Online Arabic Handwriting Recognition", Proceedings of the Second International Conference on Arabic Language Resources and Tools, pp.22-23, 2009.

[9] F. Kurniawan, M. Rahim, N. Sholihah, A. Rakhmadi, D. Mohamad, "Characters Segmentation of Cursive Handwritten Words based on Contour Analysis and Neural Network Validation", ITB J. ICT, vol. 5, no. 1, pp. 1-16, 2011.

[10] A. Rehman Khan, D. Muhammad, "A Simple Segmentation Approach for Unconstrained Cursive Handwritten Words in Conjunction with the Neural Network", International Journal of Image Processing, vol 2, no. 3, pp. 29-35, 2008.

[11] P. Cavalin, A. de Souza Britto, F. Bortolozzi, R. Sabourin, L. Oliveira, "An Implicit Segmentation-based Method for Recognition of Handwritten Strings of Characters", ACM Symposium on Applied Computing - SAC , pp. 836-840, 2006.

[12] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, S. Janet, "Unipen Project of On-Line Data Exchange and Recognizer Benchmarks", Proceedings of 12th International Conference on Pattern Recognition, pp. 29-33, 1994.

[13] M. Liwicki, H. Bunke, "IAM-OnDB - an online English sentence database acquired from handwritten text on a whiteboard", In the Proceedings of 8th International Conference on Document Analysis and Recognition, vol. 2, pp. 956–961, 2005.

[14] H. El-Abed, M. Kherallah, V. Märgner, A. Alimi, "On-line Arabic handwriting recognition competition - ADAB database and participating systems", International Journal on Document Analysis and Recognition, 2010.

[15] J. Hull, "A database for handwritten text recognition research", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 5, pp. 550–554, 1994.
[16] R. Wilkinson, J. Geist, S. Janet, In the first census optical character recognition systems Conference #NISTIR 4912, The U.S. Bureau of Census and the National Institute of Standards and Technology, Gaithersburg, MD, 1992.
[17] M. Pechwitz, S. Maddouri, V. Maergner, N. Ellouze, H. Amiri, "IFN/ENIT: Database of Handwritten Arabic Words", in Proceedings of the CIFED 2002, Tunisia, pp. 129-136, 2002.
[18] http://www.iam.unibe.ch/~fki/iamondb/
[19] S. Abdelazeem, H. Eraqi, "On-line Arabic Handwritten Personal Names Recognition System based on HMM", Proceedings of the 11th international conference on document analysis and recognition ICDAR2011, Beijing, China, pp. 1304-1308, 2011.