A Neuro-Heuristic Approach for Segmenting Handwritten Arabic Text

Alaa Hamid and Ramzi Haraty
Lebanese American University
P.O. Box 13-5053 Chouran
Beirut, Lebanon 1102 2801

Abstract

The segmentation and recognition of Arabic handwritten text has been an area of great interest in the past few years. However, a small number of research papers and reports have been published in this area due to the difficult problems associated with Arabic handwritten text processing. In this work a technique is presented that segments handwritten Arabic text. A conventional algorithm is used for the initial segmentation of the text into connected blocks of characters. The algorithm then generates pre-segmentation points for these blocks. A neural network is subsequently used to verify the accuracy of these segmentation points. Two major problems were encountered: The segmentation phase proved to be successful in vertical segmentation of connected blocks of characters. However, it couldn't segment characters that were overlapping horizontally. Second, segmentation of handwritten Arabic text depends largely on contextual information, and not only on topographic features extracted from these characters.

1. Introduction

In spite of the extensive work done on the recognition of handwritten Latin and Asian languages text and the excellent results obtained in Latin text, a few research papers and reports have been published in the area of handwritten Arabic text recognition. This is because the recognition of handwritten Arabic text is considerably harder than that of Latin text due to a number of reasons:

i. Arabic is written cursively, i.e., more than one character can be written connected to each other, forming a block of characters (BC).

ii. Arabic uses many types of external objects, such as dots, '*Hamza*', '*Madda*', and diacritic objects. These make the task of line separation and segmenting text into BCs more difficult.

iii. Arabic characters can have more than one shape according to their position inside a BC: initial, middle, final, or standalone.

iv. Characters that do not touch each other but occupy a shared horizontal space increase the difficulty of BC segmentation, as illustrated in Figure 1.

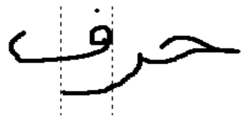

Figure 1. Two characters occupying a shared horizontal space.

v. Arabic uses many ligatures, especially in handwritten text. Ligatures, shown in Figure 2, are characters that occupy a shared horizontal space creating vertically overlapping connected or disconnected BCs.

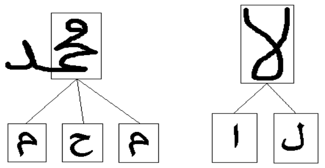

Figure 2. Arabic ligatures and their constituent characters.

This research describes a hybrid method to segment Arabic handwritten text with two main components. The first is a heuristic component, which is responsible for scanning the handwritten text, extracting connected BCs, and generating topographic features. It is also responsible for calculating pre-segmentation points, which are validated by the second component. The second component is an artificial neural network (ANN), which verifies whether pre-segmentation points are valid or invalid.

The remainder of this paper is broken down into 4 sections. Section 2 briefly describes the proposed techniques and algorithms, Section 3 provides a discussion of the experimental results, and a conclusion is presented in Section 4.

Proceedings of the ACS/IEEE International Conference on Computer systems and Applications – AICCSA-01. Beirut, Lebanon. June 26-29, 2001

2. Proposed technique

There are a number of steps that need to be taken before handwritten text can be segmented. These include scanning, binarization, and feature extraction.

2.1. Scanning

Since there is no standard benchmark database for Arabic handwritten text, samples were acquired randomly from various students and faculty members around the university. They were asked to write down their own mailing address on A4 sized paper. These addresses were then scanned at 150 pixels per inch and saved in monochrome Windows Bitmap (BMP) format. The images had different sizes ranging from 260 x 140 pixels to 1200 x 400 pixels. 360 addresses have been collected consisting of about 4000 words or 9000 BCs.

2.2. Binarization

Before any segmentation or processing could take place, it was necessary to convert the images into binary representations. A heuristic algorithm generated a matrix of ones (1's) for black pixels and zeros (0's) for white pixels.

2.3. Extracting connected BCs

The extraction of connected BCs is the first step of the segmentation phase. A recursive algorithm was implemented with 94% accuracy, scanned the whole binary matrix of the image and extracted connected BCs.

Higher accuracy couldn't be achieved because of external objects (e.g., dots and diacritics) that were too far away from their parent BCs and too near to other BCs.

2.4. Feature extraction

The feature extraction module scanned the BC binary matrix looking for topographic features to identify possible segmentation points. A complete list of the extracted features for each column of a BC is shown in Table 1. Skeletonization of the image was required in order to extract most of these features. The algorithm of [12] produced acceptable results with few enhancements and modifications.

2.5. Pre-segmentation point generation

The objective of this module was to over-segment all the connected BCs based on the features extracted for each column. The distribution of the proposed segmentation points was taken into consideration based on the average character width in a BC.

Table 1. Major features extracted for each column of BC matrix.

Feature	**Attributes**
Image width and height	
Black pixel density	Black pixel density / height
	Density minima
	Density maxima
Transitions	Number of transitions crossed
Holes	Number of holes crossed
	Total hole densities / height
Endpoints	Number of endpoints crossed
Corner points	Number of corners crossed
Fork points	Number of fork points crossed
Relative index of column in image	
Upper and lower contours	Upper and lower contour index / height
	Upper and lower contour minima or maxima
Feature relationships	Relative index of nearest left and right feature

2.6. Verification using an ANN

To train the ANN with both accurate and erroneous segmentation points, the output from the heuristic segmentation algorithm was used. It was necessary to manually separate the points generated by the algorithm into valid and invalid segmentation points and save them to a file together with the extracted set of features and desired output for each point.

To find the optimum ANN architecture to solve this problem, various networks with different types, number of hidden layers and processing elements (PEs) per each layer were tried. The ANN with the smallest number of PEs, minimum estimated generalization error, and that learned best to identify correct segmentation points was chosen.

The best ANN architecture found was a generalized feed-forward network that consisted of 52 inputs, 1 output, and 4 hidden layers, as shown in Table 2. The 52 inputs were feature attributes of a pre-segmentation point and the output was the validity of that point.

Table 2. Architecture of verification ANN

	PEs	Transfer function
Layer 1	41	Tanh
Layer 2	27	Tanh
Layer 3	20	Tanh
Layer 4	16	Tanh
Output Layer	1	Tanh

Training of the network was done using the error back-propagation technique with a training set of 48,000 exemplars. To terminate the training process efficiently, the cross-validation technique was used. Cross-validation monitors the MSE on an independent set of data and stops training when this error begins to increase. This is considered to be the point of best generalization. The cross-validation set consisted of 10,000 exemplars.

3. Experimental Results

The output range of the ANN was between –0.9 and +0.9. A positive value indicated that a point is a valid segmentation point; a negative value indicated that a point should be ignored.

A heuristic algorithm checked the results of the ANN on a test set of 10,000 exemplars. The algorithm defined the segmentation of a BC as correct when each known segmentation point was covered by an approved segmentation point by the ANN. Adequate coverage of a segmentation point is achieved when the distance from the known segmentation point to the closest approved point is less than 15% of the average character size.

Table 3 shows the results of the ANN tested on 10,000 exemplars.

Table 3. Segmentation ANN results.

	Invalid Points	Valid Points	Total
Correctly Identified	3,767	1,544	5,311
Incorrectly Identified	3,326	1,363	4,689
		Total	**10,000**

The minimum MSE achieved was 0.41. The ANN was able to identify the accuracy of 5,311 points out of the 10,000-point testing set. Of the correctly identified points, 3,767 were invalid segmentation points and 1,544 were valid segmentation points.

The ANN incorrectly identified 4,689 points. 3,326 of these points were invalid segmentation points marked as valid, and 1,363 were valid points marked as invalid.

It should be noted that the majority of incorrectly identified points were invalid segmentation points marked as valid, which implies that the ANN over-segmented the handwritten BCs. After studying the distribution of the ANN results over the range [-0.9, +0.9], it was noted that the majority of these 3,326 points were found in the [0, +0.5] range, as illustrated in Figure 3.

To decrease the number of incorrectly identified segmentation points, a threshold value was applied to the ANN output. A module was implemented which checked the ANN responses against a 0.5 threshold. Responses between 0 and +0.5 were therefore rejected. It should be noted that these rejected patterns also include 810 correctly identified valid segmentation points. However, the number of incorrectly identified points, 2,733, in the (0,0.5) range is much greater than the correctly identified points. Table 4 shows the results after rejecting these patterns.

Furthermore, rejecting patterns with responses in other ranges is not efficient because the number of correctly identified points is greater than the number of incorrectly identified points in each range.

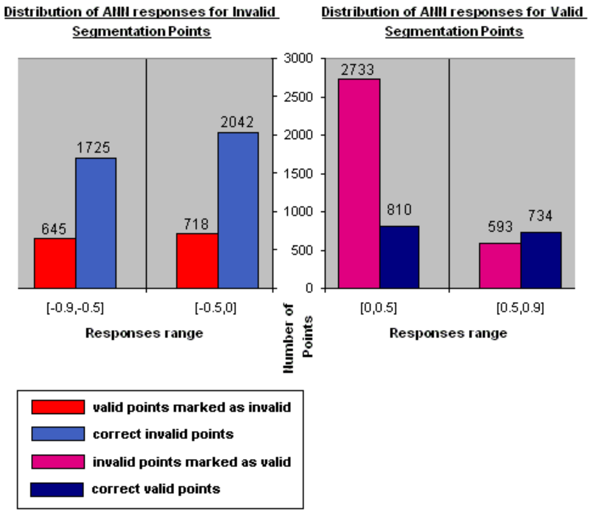

Figure 3. Distribution of ANN responses

Table 4. ANN results after rejecting patterns with responses in the (0,0.5) range.

	Invalid Points	Valid Points	Total
Correctly Identified	3,767	734	4501
Incorrectly Identified	593	1,363	1956
Rejected	2733	810	3543
		Total	**10,000**

4. Discussion

The implemented system achieved a segmentation accuracy of 53.11%, and after rejecting 35.43% of the points achieved 69.72% accuracy.

These results were attributed to objects that were impossible to segment in handwritten Arabic text. Every 100 BCs of the collected data, contained 10.16 un-segmentable ligatures and 13.02 characters with miss-located external objects. In addition, 9.24 س and ش characters occur in every 100 BCs, which are almost always un-segmentable. Other miscellaneous un-segmentable BCs include characters like the letter ض, which is always segmented into the letters ع or م, and ن.

This implies that horizontal segmentation is required in 10.16% of connected BCs. The other problems are attributed to the use of topographic features for the collected images. 22.26% of the connected BCs can be segmented differently depending on whether you look at them separately, in a word, or even in a sentence. In other words, character segmentation, especially handwritten Arabic characters, depends largely on contextual information, and not only on the topographic features.

Table 5 summarizes the results obtained by various researchers.

Table 5. Comparison of segmentation results in the literature.

Author	Accuracy	Data set used	Method used
Blumenstein and Verma [2]	81.21%	Griffith University Latin handwriting database	Neuro-conventional method
Eastwood et al. [4]	75.9%	Cursive Latin handwriting from CEDAR database.	ANN-based method
Han and Sethi [7]	85.7%	Latin handwritten words on 50 real mail envelopes	Heuristic algorithm
Lee et al. [10]	90%	Printed Latin alphanumeric characters.	ANN-based method
Srihari et al. [11]	83%	Handwritten zip codes	ANN-based method

5. Conclusion and Future Work

A heuristic segmentation technique used in conjunction with a generalized feed-forward multi-layer neural network has been presented in this paper. It was used to segment difficult handwritten Arabic text, producing promising results. With some modifications more testing shall be conducted to allow the technique to be used as part of a larger system.

The segmentation program over-segmented the BCs it was presented with. This allowed the segmentation ANN to discard improper segmentation points and leave accurate ones. Overall the whole process was very successful, however some limitations still exist.

In future work, the segmentation technique will be improved in a number of ways. First, the heuristic component of the segmentation system will need to be enhanced further. Looking for more features or possibly enhancing the current feature extraction methods can improve the accuracy results. In addition, horizontal segmentation of ligatures shall also be investigated further.

In the ANN component, more test patterns shall also be used in training and testing, and finally the technique shall be integrated into a complete handwritten recognition system with the use of contextual information.

References

[1] M. Blumenstein and B.Verma, "A Neural Based Segmentation and Recognition Technique for Handwritten Words", Griffith University, Australia.

[2] M. Blumenstein and B. Verma, "A Segmentation Algorithm used in Conjunction with Artificial Neural Networks for the Recognition of Real-World Postal Addresses", *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'97)*, Gold Coast, Australia, 1997, pp. 155-160.

[3] M. Fehri and M. Ahmed, "A Hybrid RBF/HMM Approach for Recognizing Multifont Arabic Text", Laboratoire RIADI-ENSI, Tunisia.

[4] B. Eastwood, A. Jennings, and A. Harvey, "A Feature Based Neural Network Segmenter for Handwritten Words", *Proceedings of the International Conference on Computational Intelligence and Multimedia Applications (ICCIMA '97)*, Gold Coast, Australia, 1997, pp. 286-290.

[5] B. Al-Badr and R. Haralick, "Segmentation-Free Word Recognition with Application to Arabic", University of Washington, USA.

[6] R. Srihari, "Use of Lexical and Syntactic Techniques in Recognizing Handwritten Text", Center for Document Analysis and Recognition, USA.

[7] K. Han, I. K. Sethi, "Off-line Cursive Handwriting Segmentation", *ICDAR '95*, Montreal, Canada, 1995, pp. 894-897.

[8] L. Almeida, "Multilayer Perceptrons", *Handbook of Neural Computation,* IOP Publishing Ltd and Oxford University Press, 1997, pp. C1.2: 1-C1.2: 30.

[9] T. Breuel, "Handwritten Character Recognition Using Neural Networks", *Handbook of Neural Computation,* IOP Publishing Ltd and Oxford University Press, 1997, pp. C1.2: 1-C1.2: 30, G1.3: 1-G1.3: 16.

[10] S-W. Lee, D-J. Lee, H-S. Park, "A New Methodology for Gray-Scale Character Segmentation and Recognition", *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 1996, pp. 1045-1051.

[11] S. N. Srihari, "Recognition of Handwritten and Machine-printed Text for Postal Address Interpretation", *Pattern Recognition Letters,* 1993, pp. 291-302.

[12] A. Rosenfeld, "A Simple Parallel Algorithm for Skeletonization", http://www.cs.mcgill.ca/~laleh/rosen_alg.html. 2002.