Online Segmentation of Arabic Sub-Words

George Kour

Faculty of Engineering

Tel Aviv University, Tel-Aviv Jaffa, Israel

kourgeorge@gmail.com

Raid Saabne

Faculty of Engineering

Tel Aviv University, Tel-Aviv Jaffa, Israel

Triangle R&D Center, Kafr Qarea, Israel

saabni@cs.bgu.ac.il

*Abstract*—the scripted nature of Arabic written language poses some high challenges for automatic character segmentation and recognition. Correct and efficient segmentation of Arabic text into characters is considered to be a fundamental problem. Character segmentation is a necessary preprocessing step for character recognition in many OCR systems. It is an important step because incorrectly segmented characters are unlikely to be recognized correctly. The Arabic script is composed of strokes which may contain a single or multiple connected letters. In this paper, strokes segmentation and recognition algorithm (SSRA) of Arabic scripts is presented. The system consists of three stages. The first stage is rules based engine to determine candidate segmentation points using typographical features. In the second stage a scoring is given to each possible subsequence by a recognition system to evaluate the perceptual resemblance of each subsequence to some Arabic letter. The third stage is an algorithm that determines the best subset of segmentation points. The first stage and second stages requires the most calculation effort thus are performed whilst the stroke is being written, the third stage is done at the end of the scribing process. The system has been designed and tested using the ADAB Database. Very promising results are obtained regarding the unconstrained Arabic handwriting difficulty and not using context help.

Keywords—Arabic Handwriting Recognition; Arabic Script Segmentation; Arabic Strokes Segmentation; Online Text Recognition

# Introduction

Handwriting remains the most used mean of communication and recording of information in the day-to-day life. Therefore, a growing interest in the online character recognition field has taken place in the recent years.

The Arabic language is the fifth most used languages as a first language after Chinese, Hindi, Spanish and English. It is spoken as a first language by nearly 350 million people around the globe, mainly in the Arab countries, which is about 5.5% of the world population (CIA, 2005). Nearly 25 languages have adopted the Arabic alphabet with some changes. Automated script recognition of Latin, Chinese and Kanji has been a focus of study in the last decade and impressive recognition rates were achieved. Despite the fact that Arabic alphabets are used in many languages, Arabic text recognition is at an early stage. The reason for this is lack of funds and other utilities such as text database, dictionaries, etc. [1]

Handwriting recognition (HWR) is a task of transforming a language represented in its spatial form of graphical marks into its symbolic representation [13]. Online HWR refers to the situation where the recognition is performed concurrently to the writing process.

Arabic text, both handwritten and printed is cursive. An Arabic word consists of one or more word-parts (WP). Each WP consists of one or more characters. Thus, most of the online Arabic recognition systems perform the recognition soon after the WP scribing is completed using the holistic or analytic approach.

Text recognition can be classified into two main fields: online and offline recognition. Online handwriting recognition refers to the situation where the recognition is performed concurrently to the writing process. However, in the offline script recognition field, a digital image containing text is fed to the computers and the system attempt to recognize the written text [3]. The main existing approaches for script recognition are the holistic approach and the analytic approach. The holistic approach considers the global properties of the written text while the analytic approach involves segmentation and classification of each part of the text. In the holistic approach, the recognition system needs to be trained over all words in the dictionary, while it is possible for small vocabulary of words, this is not feasible for large vocabularies (20,000 words or more). Since each words is constructed from a subset of the character alphabet, it is much more efficient to classify words using the analytic approach. [4]

Character segmentation is a critical part of the text recognition process. Correct segmentation of a word into letter is likely to result in a correct recognition. On one hand, a good segmentation algorithm is needed for a good recognition system. On the hand, a good recognition system is needed by a god segmentation algorithm. Several segmentation techniques have been proposed in the literature for Arabic OCR. However correct and efficient segmentation of Arabic text is still considered a challenging and a fundamental problem even for offline printed text. Performing such task in an online manner for handwritten texts when the segmentation is being done while the word is being written is even more challenging. Over segmentation and under-segmentation are the main problems such algorithms encounter. Later we will show how this problem is handled by our method.

The under-segmentation problem is usually as a result of two letters combinations that doesn’t contain a horizontal handler between them. For example, the following word-part لمـ (Lam i.e.) which is a combination of the letters ل (L) and م (M) or the stroke لحـ which is a combination of the letters ل (L) and ح (H). Our approach overcomes this problem by broaden the set of letters classes to include such combinations. We will refer to such combinations as hyper-letters. By doing so, even if the segmentation process did not identify a demarcation point between these two letters, it will be recognized in the later process as a single letter.

Many segmentation techniques were proposed in the literature, mostly for segmenting English cursive handwriting. These methods can be categorized to 2 main approaches:

* Dissection
* Recognition Based Segmentation

Dissection techniques learn the characteristic of the segmentation point and try to find these features in a candidate point. For example, in English cursive script segmentation a common feature is that segmentation point has local minima in the upper or lower contour of the word. Another feature is the slope of the segmentation point is low. Some techniques don’t try to segment a word to its letters but to graphemes, which are a combination of 2 or3 letters or is a part of a letter. The recognition-based techniques operation is quite different. In principle no feature-based dissection algorithm is employed. Rather, the image is divided systematically into many overlapping pieces without regard to content. These are classified as part of an attempt to find a coherent segmentation / recognition result. The main interest of this category of methods is that they bypass the segmentation problem: no complex "dissection" algorithm has to be built and recognition errors are basically due to failures in classification. In recognition-based techniques, recognition can be performed by following either a serial or a parallel optimization scheme. In the first case recognition is done iteratively in a left-to-right scan of words, searching for a "satisfactory" recognition result. The parallel method proceeds in a more global way. It generates a lattice of all (or many) possible feature-to-letter combinations. The final decision is found by choosing an optimal path through the lattice. [2]

In many cases Arabic word parts, that are connected when printed format, are written in different strokes in handwritten script. A stroke may contain a single or multiple connected letters. A stroke may also represent a delayed stroke we mentioned earlier. In this paper we propose a novel approach which performs segmentation and recognition in the strokes level. We combine both holistic and analytic techniques for recognizing open dictionary Arabic online script.

In section 2 we mention related Work done in the field of online Arabic recognition. In section 3, we describe the details of our approach. Results are displayed in the section 4. We discuss and conclude the work in section 5.

# Related Work

Mention: [1] R. I. Elanwar, M. Rashwan, and S. Mashali, “Unconstrained Arabic Online Handwritten Words Segmentation using New HMM State Design,” pp. 1046–1054, 2012.

Mention [1] M. A. I. Al-ammar, “Online Handwriting Recognition for the Arabic Letter Set,” pp. 42–49.

# Our Approach

Our approach employs both rules-based dissection and recognition-base segmentation techniques. The approach consists of several stages.

**Stage 1:** a rules-based week classifier engine for segmentation points nomination and building resemblance scoring of sub-strokes.

**Stage 2:** elimination of redundant CPs and scoring correction.

**Stage 3:** a segmentation-recognition unit, which selects the best subset of segmentation point based on the scoring given to each subsequence in stage 1.

More details on every stage are provided in the subsections below.

## First Stage: Segmentation points nomination and substrokes scoring

A stroke is a subcomponent of a WP. It contains a single or multiple letters. We assume that each letter is contained entirely in a stroke, i.e. no letter span over multiple strokes. This assumption is valid for the majority of Arabic writing styles.

A stroke is represented by a sequence of points on the 2-dimensional plane, i.e.

|  |  |
| --- | --- |
|  | (1) |

Since most Arabic segmentation points are contained in horizontal fragments (HF) of the stroke. In this stage such fragments are identified in the written text. Once an HF is detected, its medial point is set as a candidate point (CP).

It is worth mentioning that in order to get a real perceptual slop of a point, and to avoid digitizer noise, we performed a preprocessing operations on the stroke (while it is being written) that include, simplification using Douglas simplification algorithm, normalizations and resampling using splines. A candidate point  is represented by, and it holds the serial number of the point in the stroke.A sub-stroke is a sub-sequence of the stroke  that starts at candidate point and ends at candidate point, i.e.

|  |  |
| --- | --- |
|  | (2) |

The sub-stroke  may represent a letter or a portion of a letter. In some cases it also may represent a multiple connected letters; we will handle this case in later phase. The scores matrix contains the resemblance scoring for all sub-strokes.

|  |  |
| --- | --- |
|  | (3) |

is the classification scoring given by the letter classifier to the sub-stroke. It is s clear that is an upper triangular matrix. For the sake of good performance as well as segmentation correctness we add a locality constraint to improve performance and to avoid marginal segmentation. That is, we calculate a narrow band of the matrix above the diagonal is no larger than, a window parameter.

|  |  |
| --- | --- |
| if  or | (4) |

The matrixis calculated while the stroke is being scribed. Once a new candidate point is identified, a new row and column is added to the matrix with the corresponding for each added cell in the band of the lower triangular matrix. It can be easily noted that the number of cells calculates in the limited band iswhere  is the number of candidate points in the stroke.

A letters level classifier that is described in depth in [5] is used to score the sub-strokes. For the sake of completeness, here we will outline the main idea of the classification system. The classifier uses Earth movers Distance embedding technique descried in [7] to project samples of the Arabic letters to a low dimensional space. Since the embedding is into the L1 space, given a sequence, the k-NN are found using k-dtree. The exact resemblance score is determined by a minimum score that is given by a predefined linear combination of the L1 distance in the embedding space and Sokoe-Chuba DTW.

The classifier receives a sequence and a position, and it contains 4 databases; each database contains letter samples in a certain position *(Ini, Mid, Fin or Iso)* regardless of the letter label. As mentioned before a stroke can contain a single or several letters. Here we enumerate all the letters positions options that a stroke can contain.

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

Where represent more than one occurrence and  represents zero or more occurrences. The  matrix below demonstrates the databases in which the classifier will look into when trying to find the closest candidates. In Equation (3), the Greek letter represents the set of position databases that the sub-stroke may belong to:

|  |  |
| --- | --- |
|  | (5) |

## 

|  |  |
| --- | --- |
|  | (6) |

## Second Stage: Candidate points sievingand scoring correction

In this stage we eliminate redundant segmentation points. The elimination if based on the following rules:

**Rule 1**: Segmentation point lies close to the baseline.

**Rule 2**: Segmentation points do not reside in loops.

**Rule 3**: A sub-stroke length/area is proportional to the stroke/length of the containing stroke.

Rule 1 is based on the fact that segmentation point lies on the baseline (see []), thus points that are majorly far from the baseline are eliminated. The baseline is proved to be an important piece of information in both segmentation and recognition domains of both online and offline handwriting recognition. In words recognition, it plays a role in determining if an additional stroke to differentiate between diacritic dots according to their position from the baseline (above or under).

Rule 2 is clear since in most writing styles, segmentation point do not reside inside a loop.

The third rule’s goal is to avoid high scoring resulted for discordant scaling of the letters, a scoring correction should be employed. It aims to reduce the effect of scaling problem. To illustrate the discordant scaling problem, see figure below. The suffix of the letter “d” is very similar to the letter “a” (Fin). The only way to visually discriminate between them is by comparing the scaling of this suffix to the whole stroke dimensions. Thus this phase refines the recognition results according the subsequence scale in accordance to the sequence scale. We have penalized sub-strokes that are disproportional to the stroke size by multiplying the score of each sub-stroke with a Gaussian normalized by the stroke area/Length.

## Third Stage: Segmentation Selection

The goal of this phase is to select the best segmentation points set among the candidate segmentation points. It is done by finding the best segmentation path in. Three algorithms are proposed in this work. The first we name as “Forward Segmentation Selection” (FSS) and the other is named the “Backward Segmentation Selection” (BSS). The final segmentation is the segmentation that has the minimalbetween the FSS and the BSS normalized by the number of Segmentation points.

1. Forward Segmentation Selection (FSS)
2. 
3. 
4. while 
   1. 
   2. 
   3. 
   4. 
5. End while
6. Forward Segmentation Selection (FSS)
7. 
8. 
9. while 
   1. 
   2. 
   3. 
   4. 
10. End while

# Data Collection

## Samples – The ADAB database

The ADAB database is considred to be a standard in the online Arabic handwriting recognition area. It is freely available and consists of more than 20k Arabic handwritten words scribed by more than 170 different writers. The words are taken from the 937 Tunisian town/village names. The database contains the trajectory information and a plot image of the word trajectory. The ADAB-database is divided to 6 sets. Details about the number of files, words, characters, and writers are detailed in [6].

## Our Database

The information in the ADAB database provides for each word its label and the strokes that were written by the writer, thus there is no way to match strokes to letters. To test our segmentation system as well as providing letter samples to the classifier we have manually segmented the words in the database. The result of this process is an xml for each word sample that contain letter level information, as well as strokes and WPs level details. In addition, we have filtered out additional strokes. We have manually segmented ~8k samples which consisted about ~20k strokes.

Our system reads an xml files, extracts the strokes and information we provided our training system with letters and also extracted strokes to test our system.

# Results

A correct segmentation of a stroke means that all the letters in the stroke are correctly segmented. In table 1 below, you can see the length distribution of the strokes in the test set. By length we mean the number of letters the stroke contains.

The total number of tested words is 2000. The total number of strokes is 6238. The Segmentation rate is 84%. Over Segmentation rate is 9.2% and under segmentation percentage is 6.8%. The percentage of strokes that could be exactly recognized is 72%.

1. Distribution of the Strokes according to Their length

| Stroke Length | Number of samples | Percentage of samples |
| --- | --- | --- |
| 1 | 3117 | 50% |
| 2 | 1796 | 28.8% |
| 3 | 986 | 15.8% |
| 4 and more | 339 | 5.4% |

[Performance]

##### References

1. Zeki, Ahmed M., Mohamad S. Zakaria and Choong-Yeun Liong. "Segmentation of Arabic Characters: A Comprehensive Survey." Technology Diffusion and Adoption: Global Complexity, Global Innovation. IGI Global, 2013. 251-288. Web. 16 Jun. 2013. doi:10.4018/978-1-4666-2791-8.ch016
2. R. G. Casey and E. Lecolinet, “A survey of methods and strategies in character segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 18, no. 7, pp. 690–706, Jul. 1996.
3. M. Al-Ammar, R. Al-Majed and H. Aboalsamh, "Online Handwriting Recognition for the Arabic Letter Set," Recent Researches in Communications and IT.
4. S. D. Connell, “Online Handwriting Recognition Using Multiple Pattern Class Models,” *PH.D Thesis*.
5. My other paper
6. M. Kherallah, N. Tagougui, A. M. Alimi, H. El Abed, and V. Margner, “Online Arabic Handwriting Recognition Competition,” *2011 International Conference on Document Analysis and Recognition*, pp. 1454–1458, Sep. 2011.
7. S. Shirdhonkar and D. W. Jacobs, “Approximate earth mover’s distance in linear time,” *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, Jun. 2008.