Recognition of Off-Line Handwritten Arabic Words Using Hidden Markov Model Approach

Somaya Alma’adeed, Colin Higgens, and Dave Elliman
University of Nottingham
sxa, cah, dge@cs.nott.ac.uk

Abstract

Hidden Markov Models (HMM) have been used with some success in recognizing printed Arabic words. In this paper, a complete scheme for totally unconstrained Arabic handwritten word recognition based on a Model discriminant HMM is presented. A complete system able to classify Arabic-Handwritten words of one hundred different writers is proposed and discussed. The system first attempts to remove some of variation in the images that do not affect the identity of the handwritten word. Next, the system codes the skeleton and edge of the word so that feature information about the lines in the skeleton is extracted. Then a classification process based on the HMM approach is used. The output is a word in the dictionary. A detailed experiment is carried out and successful recognition results are reported.

1. Introduction

Off-line handwriting character recognition is the automatic transcription by computer, where only the image of that handwriting is available. Much work has been done on the recognition of Latin characters, both of separated and in cursive script. That of recognizing Arab characters is still limited. Recognizing Arabic characters is also important for non-speaking Arabic countries such as Farsi, Curds, Persians, and Urdu-speakers who use the Arabic characters in writing although the pronunciation is different. Among those are the following. Abuhaiba et al., dealt with some problems in the processing of binary images of handwritten text document [1]. Almuallim and Yamaguchi proposed a structural recognition technique for Arabic handwritten words [2]. A look-up table is used for the recognition of isolated hand-written Arabic characters [3]. Amin and others proposed a technique for the recognition of hand-printed Arabic characters using a neural network [4]. Obaid introduced Arabic handwritten character recognition by neural networks [5]. Saleh et al. describes an efficient algorithm for coding handwritten Arabic characters [6]. It is noted that most of these works are done assuming that the Arabic handwritten word is already segmented into separated characters before recognition. In this paper, we deal with the segmentation step, as well as the recognition step of offline handwritten Arabic words, since off-line Arabic character recognition operations have many steps that cannot be separated from each other.

In this paper, the Arabic handwritten word recognition problem is modelled within the framework of a hidden Markov model (HMM). The states of the HMM are identified with the letters of the alphabet. Once the model is established the Viterbi algorithm is used to recognize the sequence of letters composing the word.

1.1. Hidden Markov Models in Word Recognition

During the last decade, HMMs have become the predominant approach to automatic speech recognition. The success of HMMs in speech recognition has led many researchers to apply them to handwriting recognition by representing each word image as a sequence of observations [7]. The success of these attempts, however, was limited to constrained experiments: single writer, small and fixed vocabulary, small test samples, etc. The application of HMM’s to the more general problem of handwriting recognition involving large dictionaries, off-line data, unconstrained style, etc., was introduced in [7-11]. Besides handwriting recognition, HMMs have also been used to analyse document images. Vlontzos and Kung have proposed a multilevel structure of HMMs for the recognition of machine (or hand-) printed text [12]. Kuo and Agazzi have successfully spotted key words in a poorly printed document image using the pseudo 2-D HMM [13]. A complete scheme for encoding the printed document using HMM’s, from the pixel level to characters, words, and whole documents, is proposed by Kopec and Chou [14].

There are also some implementations of HMMs with Arabic OCR. The following trials do not include the implementation of HMMs on handwritten Arabic words.

Bazzi et al. [15] present an omni font, OCR system for printed English and Arabic text. The system is based on Hidden Markov Models (HMM). In this paper they focus on how to perform OCR for omni font and multi-
style data, such as plain and italic, without the need to have a separate model for each style. Atic et al [16] developed a heuristic method for segmentation, feature extraction and recognition of the printed Arabic script. Chain code transformation is applied to the main strokes of the characters, which are classified by an HMM in the recognition stage.

The application of HMMs to Arabic OCR was first attempted by Amin and Mari [17]. They used HMMs in the Post-processing stage to improve the recognition accuracy. Also in [18] they used a system that depends on the estimation of character models, a lexicon, and grammar from the training samples. Subsequently Khorsheed and Clocksin [19] present a technique for the off-line recognition of cursive printed Arabic script based on an HMM. The HMM was trained using words written in one typeface and one size, and test samples were written in two different typefaces in three sizes. Recognition rates range from 68% to 73% depending on the task. However, this does not suit Arabic handwritten words, since dots (which are important in handwritten Arabic character) are not written exactly below or above each character or edge feature as described in the paper. Also, some of the geometrical features don't suit handwritten Arabic words. In [20] a holistic system for the recognition of handwritten Farsi words using HMMs and Kohonen self-organizing vector quantization was presented. They divided the image into fixed-width frames. Each frame is divided horizontally into five zones. Each zone has four features depending on the contour direction. In this way, each frame is represented as a 20-dimensional feature vector. According to the special property of Handwritten Arabic writing we believe that their features are not enough to get a reasonable recognition rate. The recognition rate was 32% without smoothing. Except [20], the above experiments [15-19] using an HMM approach are tested on printed text not handwritten words.

1.2. Difficulties with Handwritten Arabic Characters and the Differences from Latin

Arabic handwritten characters suffer not only from scale, location and orientation variation, but also person-dependent deformations. These variations are neither predictable nor mathematically formulated. In the system build in section 3 some trials for solving the problems of Arabic handwriting recognition are implemented in pre-processing steps. The main characteristics of Arabic writing can be summarized in [4]. An important difference of Arabic handwritten characters from that of Latin is the existence of more dots. Other important differences can be seen in [21]. This research deals with the recognition of Off-Line Handwritten Arabic Characters. The problem arises from many factors which can be summarised as follows. Firstly, the system deals with cursive handwritten Arabic characters, which differ from machine print. It also deals with Arabic writing, which differs from English writing in many facets. The basic problems of handwriting recognition are common to all languages, but the special features, constraints, etc. for each language need to be considered also. Finally, it deals with off-line recognition, which differs from on-line recognition systems [21].

2. System Overview

This paper describes the operation of the complete classification process for a handwriting recognition system for a single Arabic word, from the handwriting Arabic word on the database to the output of recognized word. Any word recognition system can be divided into sections: pre-processing, recognition and post-processing. The handwritten word is normalized to be presented in a more informative manner by the stage of pre-processing. Then, recognition is carried out to identify the word; this is done by firstly estimating the data likelihoods for each frame of data in the representation, using a vector quantization (VQ) method. The proposed system has three main advantages. First, it deals with non-segmented words. Secondly, it takes advantage of the position of features in the character or sub character. Thirdly, more than 29 features are calculated that are used by the VQ and HMM classifier. The following sections, describe the operation of pre-processing. It also discusses the features used in such a system. In section four, the HMM classifier is discussed, which classifies the features captured from the word image. Finally, experimental results are included in section five.

3. Pre-processing

The main advantage in pre-processing the handwritten word image is to organize the information to make the subsequent task of recognition simpler. The main part of the pre-processing stage is normalization,
which attempts to remove some of those variations in the images, which do not affect the identity of the word [22]. This system incorporates normalization for each of the following factors: stroke width, slope, and height of the letters (see figure 1.). The normalization task will reduce each word image to one consisting of vertical letters of uniform height on a horizontal base line made of one-pixel-wide strokes. In this system, the word image is loaded and cropped. Then the slant and slope of the word is corrected and thinned. Features are calculated to represent the useful information contained in the image of the word [23]. Finally the word will be segmented into frames, so the previous features in these frames could be distributed.

4. The HMM Classifier

A Hidden Markov Model (HMM) is a doubly stochastic process with an underlying Markov process that is not observable (the states are hidden), but can only be observed through another set of stochastic process, which are produced by the Markov process (the observations are probabilistic functions of the states). Let us assume that a sequence of observation \( O = (o_1, \ldots, o_T) \) is produced by the state sequence \( Q = (q_1, \ldots, q_T) \) where each observation \( o_t \) is from the set of \( M \) observation symbols \( V = \{ v_k ; 1 \leq k \leq M \} \) and each state \( q_t \) is from the set of \( N \) states \( S = \{ s_i ; 1 \leq i \leq N \} \). Thus, an HMM can be characterized by \( \Pi = \{ \pi_i \} \), where \( \pi_i = P(q_1 = s_i) \) is the initial state probability; \( \Lambda = \{ a_{ij} \} \), where \( a_{ij} = P(q_{t+1} = s_j | q_t = s_i) \) is the state transition probability; \( \beta = \{ b_j(k) \} \), where \( b_j(k) = P(o_t = v_k | q_t = j) \) is the symbol probability; And they satisfy the probability constraints:

\[
\sum_{i=1}^{N} \pi_i = 1; \quad \sum_{i=1}^{N} a_{ij} = 1 \quad \forall i; \quad \sum_{j=1}^{M} b_j(k) = 1 \quad \forall j .
\]

We will denote the HMM by a compact notation \( \lambda = (\Pi, \Lambda, \beta) \). Instead of having 120 classes (Arabic characters) we will have only 55 classes, since we will implement the recognition system on the lexicon of words used on Arabic cheques. The 55 letters or sub-characters of the alphabet are defined as the states of our HMM, the initial \( \pi_i \), and transition \( a_{ij} \) probabilities are computed as:

\[
a_{ij} = \frac{\text{number of trans. from } l(q_i) \text{ to } l(q_j)}{\text{number of transition s from } l(q_i)}
\]

\[
\pi_i = \frac{\text{number of words beginning with } l(q_i)}{\text{total number of words in the dictionary}}
\]

Where the \( l \) function transforms a state to its representative member of the alphabet. Our System will have two phases training and testing as illustrated in figure 2.

4.1. Training Problem

Given the training sequence \( O = o_1, \ldots, o_T \), we want to adjust the model parameters \( \lambda = (\Pi, \Lambda, \beta) \) such that \( P(O|\lambda) \) is maximized. The Baum-Welch Algorithm has been used as the optimization criterion for finding the Maximum Likelihood (ML). In general, HMM’s can be trained by the Baum-Welch algorithm with satisfactory performance [7].

4.2. Recognition phase

In recognition phase a Modified Viterbi Algorithm (MVA) that can solve the recognition problem has been used [7].

5. Experimental Results

The system described in this paper has been applied to a database of handwritten words produced by 100 writers. It was created especially for this application since there is no previous publicly available database of Arabic handwritten characters available to use as a standard test bed, as the case in English writing. Samples of about 10000 Arabic words for the lexicon used in cheque filling were gathered and stored in separated files. Feature classification must be optimal in order to extract the distinct primitives and correctly identify the characters. However, the hand-printed characters tended to have “hairs” that created problems in generalizing the primitives. Another problem encountered with feature extraction was that of fixing the direction of curves. The database was segmented into training and testing data. In the training phase the words was segmented into characters or sub-characters. Feature extraction then transformed the segment images into quantity feature vectors, which were then partitioned into several groups.
by the clustering algorithm. The cluster centroid (or codebook) from this part of training is kept for further use. For each word, the (quantized) observation sequences are then used to train an HMM for this word, using the Forward-Backward re-estimation algorithm. In the testing phase a frame based analysis (pre-processing, feature extraction, and segmentation) is performed to give observation vectors, which are then vector quantized into the possible codebook vectors. The code book size was chosen empirically after testing to be 70. The resulting observation sequence and the probability measures for all the HMMs, given by $\lambda=\{\Pi, A, B\}$, are then used to calculate the log-likelihoods for the HMMs. The word associated with the HMM of highest log-likelihood is declared to be the recognized word, and the index is returned. Two thirds of the data were used for training and the rest for testing. By training and testing the system on the database of handwritten Arabic words, the system has obtained a near 45% recognition rate (without some post-processing). The prototype systems described here are promising, however, there remains a lot of room for improvement in terms of early use of the dictionary. Comparison of results, obtained in this research, with other researchers is difficult because of differences in experimental details, the actual handwriting used, the method of data collection, and dealing with real Arabic off-line handwritten words. If this work is compared to an Arabic thesis [1-6, 17-20], on recognition of Arabic off-line handwritten words, it is the first one that uses an HMM to recognize Arabic handwritten words. This also means that it uses different sets of features and segmentation techniques.

References


