Alkhateeb, Jawad H. and Ren, Jinchang and Jiang, Jianmin and Al-Muhtaseb, Husni (2011) *Offline handwritten Arabic cursive text recognition using hidden markov models and re-ranking.* Pattern Recognition Letters, 32 (8). pp. 1081-1088. ISSN 0167-8655

Offline Handwritten Arabic Cursive Text Recognition using Hidden Markov Models and Re-ranking

Jawad H AlKhateeb, Jinchang Ren, and Jianmin Jiang
School of Informatics
University of Bradford
Bradford BD7 1DP, United Kingdom
j.h.y.alkhateeb@bradford.ac.uk, j.ren@bradford.ac.uk, j.jiang1@bradford.ac.uk

Offline Handwritten Arabic Cursive Text Recognition using Hidden Markov Models and Re-ranking

Jawad H AlKhateeb, Jinchang Ren,and Jianmin Jiang
School of Informatics
University of Bradford
Bradford BD7 1DP, United Kingdom
j.h.y.alkhateeb@bradford.ac.uk, j.ren@bradford.ac.uk, j.jiang1@bradford.ac.uk

Abstract: Recognition of handwritten Arabic cursive texts is a complex task due to the similarities between letters under different writing styles. In this paper, a word-based off-line recognition system is proposed, using Hidden Markov Models (HMMs). The method employed involves three stages, namely preprocessing, feature extraction and classification. First, words from input scripts are segmented and normalized. Then, a set of intensity features are extracted from each of the segmented words, which is based on a sliding window moving across each mirrored word image. Meanwhile, structure-like features are also extracted including number of subwords and diacritical marks. Finally, these features are applied in a combined scheme for classification. Intensity features are used to train a HMM classifier, whose results are re-ranked using structure-like features for improved recognition rate. In order to validate the proposed techniques, extensive experiments were carried out using the IFN/ENIT database which contains 32492 handwritten Arabic words. The proposed algorithm yields superior results of improved accuracy in comparison with several typical methods.

Keywords: Off-line Arabic handwritten recognition; Hidden Markov Models (HMM); Re-ranking; IFN/ENIT database; Machine learning.

1. **Introduction**

Handwriting recognition (HWR) is a mechanism for transforming the written text into a symbolic representation, which plays an essential role in many human computer interaction applications including cheque verification, mail sorting, office automation, as well as natural human-computer interaction (Alma'adeed et al. 2004, Al-Hajj et al 2009, Ball 2007, Kessentini et al 2008). HWR for Latin and Chinese languages has been conducted and significant achievements have been made. However, there has been less work in Arabic handwriting recognition. This is due to the complexity of the Arabic language and lack of public Arabic handwriting databases.

In general, HWR can be categorized into two distinct types: online and off-line based systems. Online recognition is relatively easier as it can make use of additional information not available to the off-line systems such as the strength and sequential order of the writing (Amin 1998). On the contrary, off-line recognition is more difficult as it is based solely on images of written texts. However, online recognition is impossible in many applications hence off-line recognition is focused in this paper.

The recognition of handwritten Arabic scripts can be divided into segmentation based or segmentation free approaches. The former segments words into characters or letters for recognition and can be regarded as an analytical approach. The latter, which can be regarded as a global approach, takes the whole word image for recognition and therefore needs no segmentation. Although the global approach makes the recognition process simpler, it requires a larger input vocabulary than analytical approach (Khorsheed 2002).

Typical classifiers used for HWR include k-Nearest Neighbor, Neural Networks, Support Vector Machines (SVM), Hidden Markov Model (HMM), Bayesian Classification, and decision trees (Abdulkadr 2006, Alkhateeb et al. 2009a, Alkhateeb et al. 2009c, Amin et al 1996, Graves and Schmidhuber 2008, Khorsheed 2002, Lorigo and Govindaraju 2006). Unlike dealing with printed documents, recognition of handwritten Arabic texts is more difficult due to the difference in writing styles and the variations of writing in terms of stroke length, stroke regularity and stroke location, etc. As a result, an ideal classifier needs to cope with such variations in modeling the problem. In addition, combination of multiple classifiers for improved recognition becomes a new trend, though it inevitably leads to high complexity of the overall system (Al-Hajj et al 2009, Zavorin et al 2008, Menasri et al 2007).

It is our intension to design a recognition system to deal with unconstrained Arabic handwriting words written by multiple writers. Using the IFN/ENIT Arabic standard database as test set, the results of our proposed system are among the best in comparison with quite a few state-of-art approaches. The main contributions which cover several key techniques proposed in our system can be highlighted as follows.

1) To design efficient pre-processing algorithms for baseline detection and word segmentation, where statistical analysis and knowledge-assisted decision making are employed.
2) With detected baseline, several structural features are extracted such as subwords, single/double/triple dots below the baseline and single/double dots above the baseline. From segmented words, a group of intensity features are extracted using a sliding window applied on mirrored word image;
3) To apply the above features for word recognition using a combined scheme, where intensity features are used to train a HMM classifier, whose results are further improved via re-ranking using extracted structural features. The re-ranking scheme is found to generate much improved results yet avoids multiple classifiers and additional non-visual features as used in other systems.
4) A quantitative measurement is defined to indicate how biased a test set is distributed, and this is further used for error analysis when results from different test sets are available.

The remainder of this paper is structured as follows. Section 2 describes the Arabic language while Section 3 presents literature review. Details of our proposed method in terms of pre-processing and feature extraction as well as classification are discussed in Section 4.

Comprehensive results are presented in Section 5, and the paper ends with conclusions and suggestions for further work in Section 6.

2. The Arabic Language

Written by more than 250 million people, Arabic is one of the major worldwide document sources (Amin 1998). With few similarities and many differences to written English, by its nature, Arabic text is cursive, which makes its recognition more difficult than that of printed Latin text. On the other hand, Arabic writing, in a similar way to English, uses letters. The Arabic alphabet consists of 28 letters, and text is written from right to left in a cursive way. Each Arabic letter has either two or four shapes depending on its possible position in the text, including *start*, *middle*, *end*, or *alone* (Amin et al. 1996, Abdulkadr 2006). For example, the letter Ayn (ع) has the following shapes: start عـ middle ـعـ, end ـع, and alone ع. The details of the letter shapes are illustrated in Table 1, and obviously this has brought more difficulty for automatic recognition of Arabic texts.

In addition, the Arabic language uses diacritical marking such as *fattha*, *dumma*, *kasra*, *hamza*(zigzag), *shadda*, or *madda*. Using dots makes some Arabic letters special as follows (Amin 1998, Lorigo and Govindaraju 2006):

- Ten Arabic letters have one dot (ب،ج،خ،ذ،ز،ض،ظ،غ،ف،ن)
- Three Arabic letters have two dots (ت،ق،ي)
- Two Arabic letters have three dots (ث،ش)
- Several Arabic letters include loop (ص،ض،ط،ظ،ـعـ،ـغـ،ف،ق،م،ـمـ،و،ة)

The presence or absence of vowel diacritical indicates different meanings (Amin, 1998). For example: كلية refers to college or kidney, and حب denotes love or seeds, and diacritical marking are essential to differentiate between possible meanings. However, the diacritical marking may be ignored in handwritten unless the words are isolated, and this introduces additional difficulty in our recognition task. As removal of any of these dots will lead to a misinterpretation of the character, efficient pre-processing techniques have to be used in order to deal with these dots without removing them and changing the identity of the character.

There are six letters which are not connected from the left resulting in the separation of the word into sub-words or pieces of Arabic words (PAW) (Amin, 2000, Lorigo and Govindaraju, 2006, Amin, 1998). Generally, the handwritten text is written on a page divided into lines which are further divided into words. There are spaces between the lines and the words. The spaces between the words define the word boundaries. Usually, the space between the sub-words is one third of the space between the words. This is done consistently in printed text, however, it varies in handwritten text and leads to inconsistency in segmentation of words and subwords (Amin, 2000).

3. Literature Review

In (Khorsheed and Clocksin, 1999), words were recognised as a single unit depending on a predefined lexicon. In Stentiford's algorithm (Parker, 1997), skeleton of words were used for word recognition, where structural features were extracted for recognition in three consecutive steps, including segment extraction, loop extraction and segment transformation. Using vector quantization (VQ), each feature vector was mapped to the nearest symbol in the codebook resulting in a sequence of observation to be fed into HMM. The technique was tested with a lexicon of 294 words acquired from different text sources, and a recognition rate of up to 97% was achieved.

Khorsheed (Khorsheed 2003) presented another holistic recognition system for recognizing Arabic handwritten words, where structural features for the handwritten script were extracted after decomposing the word skeleton into a sequence of links with an order similar to the writing order. Using the line approximation (Parker, 1997), each line was broken into small line segments, which were transferred into a sequence of discrete symbols using VQ. Then an HMM recognizer was applied with image skeletonization to the recognition of an old Arabic manuscript (Khorsheed 2000). The HMM was performed using 296 states on 32 character models, and each model was left to right HMM with no restriction jump margin. The system was tested on 12960 recognition tests associated with 405 character samples of a single font extracted from single manuscript. The recognition rates achieved was 87% with spelling check and 72% if not.

Pechwitz and Maergner (Pechwitz and Maergner, 2003) presented an off-line system for the recognition of isolated Arabic handwritten words, where the IFN/ENIT dataset version v1.0p2 (Pechwitz et al., 2002), containing four sets (a-d), was used to valid their system. Pixel values from sliding windows were used as main features, along with Karhunen Loeve Transformation (KLT) for feature dimension reduction. The first three sets (a-c) were used for training and the remaining one for testing their Semi Continuous HMMs (SCHMM) classifier, and a recognition rate of 89% was achieved.

SCHMM was also used in Benouareth et al (Benouareth et al 2006, Benouareth et al. 2008) for off-line unconstrained handwritten Arabic word recognition. Statistical and structural features were utilized on the basis of the adopted segmentation in which implicit word segmentation was used to divide images into vertical frames of constant and variable width for feature extraction. Based on maxima and minima analysis of the vertical projection histogram, morphological complexity of handwritten characters is further considered. Using the same dataset and under same experimental conditions, the recognition rate achieved with uniform segmentation was 81.02% for top 1 and 91.74% for top 10. For non-uniform segmentation, the recognition rate was 83.79% for top 1 and 92.12% for top 10, respectively.

Similar strategy was also employed in another HMM-based system (El Abed and Maregner, 2007), where statistical features were length of skeleton in four directions extracted from five horizontal zones of equal height. Using the same training and testing conditions with the IFN/ENIT v1.0p2, the recognition rate achieved was 89.1% for top 1 and 96.4% for top 10 candidates. In (El-Hajj et al., 2005), HMM was also applied to word recognition, using 24 statistical features like foreground pixel density and concavity extracted from divided word image along with 15 baseline independent features. Through modeling each character with a left to right topology, their HMM classifier had four states for each character model resulting 159 character models in total. Again using the IFN/ENIT database v1.0p2 for training and set d for testing, their system had a recognition rate of 75.41%.

Al-Hajj et al. (Al-Hajj et al., 2007) presented a two stage system for recognizing handwritten Arabic words. In the first stage, three HMM classifiers were applied with pixel-based features to determine the best ten candidates (Top 10) using likelihood. In the second stage, results from these classifiers were fused for a combined decision via three schemes, including the sum rule, the majority vote rule, and neural network based fusion. Using the IFN/ENIT benchmark database, the recognition rate achieved was 90.96%. These three schemes were also used in (Al-Hajj Mohamad et al., 2009) to combine three homogeneous HMM classifiers for improved performance. The recognition rate achieved on IFN/ENIT v1.0p2 was 90.26% for top 1, 94.71 for top 2, and 95.68% for top 3.

4. Proposed Techniques and System Implementation

In this paper, we proposed an off-line recognition system for the handwritten Arabic cursive using HMM and re-ranking. The whole system contains three stages in terms of preprocessing, feature extraction, and classification in the following sections. The block diagram of the proposed handwritten Arabic cursive text recognition system is shown in Figure 1. As shown in Fig. 1, once a sample image is acquired, pre-processing is required to standardize the signal for better performance in the following stages. Afterwards, features are extracted and fed to a HMM classifier for classification. The results of the HMM is further refined by using a re-ranking scheme for improved accuracy. Relevant techniques are discussed in details as follows.

4.1 Preprocessing

The main aim of preprocessing is to enhance the inputted signal and to represent it in a way which can be measured consistently for robust recognition. Here preprocessing stage involves scanning the paper document, removing noise, image enhancement, and segmentation, which are strongly dependent on the quality of the paper document. As a result, pre-processing includes many relevant techniques such as thresholding, skew/slant correction, noise removal, thinning and baseline estimation as well as segmentation of words, subwords and even characters.

Although separate words have been manually segmented and binarized in the IFN/ENIT database (Pechwitz et al., 2002), we have investigated how to generally detect the baseline and also segment words from scanned handwritten texts using knowledge-based statistical models. Firstly, we project a given image to the vertical axis, and calculate the sum of pixels accordingly. The baseline is then determined as the one of the peak value in the projected signal. Since the baseline is located below the middle line, only the peak value in the bottom half part of the projected signal is used for its detection.

After detection of the baseline, word and subword are segmented as follows. Firstly, the input image is projected to the horizontal axis to form a vertical histogram. Then, distances between each pair of non-zero bins in the histogram are extracted. If this distance is no less than a threshold d_w, it refers to boundary of two words. Otherwise, if the distance is less than d_w but larger than another smaller threshold d_s, it is detected as boundary of subwords. The two thresholds d_w and d_s are optimally determined using Bayesian minimum classification error criteria, and further details can be found in (AlKhateeb et al. 2008, AlKhateeb et al. 2009b).

In an ideal handwriting model, the word has to be written in a horizontal way with both ascenders and descenders aligned along the vertical direction. However, these conditions are rarely satisfied in real data. Therefore, normalization is essential to remove the variation in handwritten images for consistent analysis and measurement. Among many algorithms proposed for this purpose, the skeletonization technique is one of the most popular and likewise the normalization algorithm in (Pechwitz and Maergner, 2003) has been employed in this research. A sample image in binary format is shown in Figure 2(a), along with its normalized counterpart in Figure 2(b).

4.2 Feature Extraction

Feature extraction is to remove the redundancy from the data and gain a more effective representation of the word image by a set of numerical characteristics, i.e. extracting most essential information from raw images. According to (Madhvanath and Govindaraju, 2001), features used in off-line recognition are classified into high level features which are extracted from the whole word image, medium level features which are extracted from the letters, and low level features which are extracted from sub-letters. Moreover, features can also be classified into structural and statistical ones. Structural features describe the topological and geometrical characteristics of a pattern, which include strokes, endpoints, loops, dots and their position related to the baseline. While statistical features are derived from statistical distribution of pixels and describing the characteristic measurements of a pattern, which include zoning, density distribution of pixels that counts the ones and zeros, moments (Lorigo and Govindaraju, 2006) etc.

To cope with the characteristics that how Arabic texts are written, sliding windows/frames technique is widely used from right to left to extract features for off-line recognition (Husni et al., 2008). In this paper, the sliding window technique used in speech recognition (Husni et al., 2008) has been adopted, yet applied to mirrored word image (MWI) after normalization in size to speed both training and testing process. For other features like discrete cosine transform (DCT) coefficients and moment invariants, please refer to our previous work in (Alkhateeb et al 2009c, Alkhateeb et al 2009d).

Starting from the first pixel of the word, a sliding window is applied to the MWI to calculate the number of non-background pixels. The horizontal sliding window has the same height of the word image, three pixels in width with one overlapped pixel. When the sliding window is moving from left to right, as shown in Figure 3, each MWI is divided into fifteen uniform strips/frames horizontally. From these window strips, in total 30 features are extracted as follows.

Firstly, the first fifteen features ($F_1 - F_{15}$) are determined as average intensity of the pixels in each strip, i.e.

$$F_i = (\text{Average pixel intensity in the i}^{\text{th}} \text{ vertical area}) \mid \text{i} \in [1,15] \qquad (1)$$

Then, average of these 15 features is used as the sixteenth feature F_{16}, which denotes overall mean intensity of the whole word image.

$$F_{16} = \sum_{i=1}^{15} F_i / 15 \tag{2}$$

Afterwards, the mean intensity of each consecutive pair of strips is extracted as fourteen additional features (F_{17}- F_{30}) as follows.

$$F_{i+16} = (F_i + F_{i+1})/2, \quad \mathrm{i} \in [1,14] \tag{3}$$

In addition, several structure-like features are also extracted including number of connected regions n_r, number of connected regions (dots) below the baseline n_b, and number of connected regions above the baseline n_a. These are called structure-like features as to some degree they represent topological structure of the image. How to use these features to refine recognized results in a combined scheme are described in details below.

4.3 Combined scheme for classification using HMM and re-ranking

Using the extracted features above, a combined scheme is proposed for recognition, using HMM as basic classifier followed by structure feature based re-ranking. HMM has great potential for handwritten recognition (Gunter and Bunke 2004), especially in modeling connected nature of Arabic cursive script (Khorsheed 2003, El-Hajj et al 2005, Pechwitz & Maergner 2003). Basically, HMM is a finite set of states (N), each of which is associated with a probability distribution (Rabiner 1989). Transitions among the states are governed by a set of probabilities called transition probabilities. To design such a HMM classifier, several procedures need to be followed including i) deciding number of states and observations, ii) choosing HMM topology, iii) model training using selected samples, and iv) testing and evaluation.

In this paper, we implement our HMM classifier using the HMM Toolkit (HTK), a public available platform for HMM development which was first used for speech recognition (Young et al., 2001). The simplest but most widely used Bakis topology is employed in our HMM. An example of such topology with seven states is illustrated in Fig. 4, allowing state transitions to the same state, the next state, and the following states only. Such constraints on state transition are consistent with feature-based observations, as the later are sequentially extracted from overlapped windows. As a result, allowing transition to the next two states is useful to incorporate with potential mis-alignment in segmenting word.

In the training phase, the model is optimized using the training data through an iterative process. The Baum-Welch algorithm, a variant of the Expectation Maximization (EM) algorithm, is utilized to maximize the observation sequence probability $P(O|\lambda)$ of the chosen model $\lambda = (\pi, A, B)$ for optimization, where parameters A, B and π respectively denote *matrix of transition probabilities*, *matrix of emission probabilities*, and *initial states probabilities*. For a training dataset of L observation sequences $V = V_1V_2...V_L$, the optimization aims to adjust model parameters and maximize the term $P(V \mid \lambda)$.

In the testing phase, a modified Viterbi algorithm is used for recognition. Given a optimized HMM $\lambda = (A, B, \pi)$ and an observation sequence $O = o_1o_2...o_N$, the observation (feature vector) is modeled with a mixture of Gaussian. Then, the Viterbi algorithm is used which searches for the highest model probability of a word given the input feature vector $P(O|\lambda)$ as

$$Q = \arg\max P(O|\lambda). \quad (4)$$

In our implemented HMM, the first K candidates of highest probability are attained and denoted as $Q = \{q_1, q_2, ..., q_K\}$. Meanwhile, their associated probability values are denoted as $\{p_1, p_2, ..., p_K\}$ where $p_1 \geq p_2 \geq ... \geq p_K$. Instead of taking q_1 as the best recognized result, all candidates in Q are re-ranked and re-ordered according to their refined probability values $\{p'_1, p'_2, ..., p'_K\}$. As a result, the best recognized result(s) will be the one(s) of maximum refined probability values.

$$q_m = \arg\max_m P'_m. \quad (5)$$

Structure-like features are used in our re-ranking scheme as follows. For an observation O, denote its structure features as $\{n_a, n_b, n_r\}$. For one candidate class c in Q, its associated probability p_c is refined as

$$p'_c = p_c \prod_{t=a,b,r} R_t(n_t, c) . \quad (6)$$

$$R_t(n_t, c) = \exp\left(-(\frac{n_t - \bar{n}_{t,c}}{\sigma_{t,c}})^2 \right). \quad (7)$$

where R_t is a Gaussian-like function for re-ranking and t is the index of structure features; parameters $\bar{n}_{t,c}$ and $\sigma_{t,c}$ respectively denote the mean and standard deviation of R_t for class c, which are determined during the training stage using all samples that belong to c. As seen, R_t achieves its maximum value of 1 when we have $n_t = \bar{n}_{t,c}$. Otherwise, the value decreases as a penalty to p_c.

It is worth noting that the above re-ranking scheme is different from several existing ones such as (AL-HAJJ et al 2009), (Prasad et al 2010), and (Saleem et al 2009). Actually, in the first two systems above, re-ranking is achieved via fusion of multiple classifiers, such as three HMMs in the first one and both a SVM classifier and a HMM classifier used in the second one. In addition, non-visual information like language model and even acoustic scores are employed for re-ranking in the last two systems. Our re-ranking scheme, on the contrary, relies on neither multiple classifiers nor additional non-visual features yet it produces much improved results as reported in Section 5. For some important parameters of the HMM, such as number of states and codebook size, they are empirically determined and relevant results are also reported and compared in the next section.

5. Experimental Results

In this section, the performance of our system is evaluated, using the well-known IFN/ENIT database. Several experiments are conducted and compared with numerous typical systems from others, under the same settings. Relevant results are presented in details below.

5.1 IFN/ENIT database

Although some work was conducted in Arabic handwritten words since three decades ago, generally they had small databases of their own or the presented results on databases which were unavailable to the public. In addition, real data from banking or postal mails are either confidential or inaccessible to common user groups. For performance evaluation of different approaches, a large and public available dataset is very essential. It is not until 2002 that such a dataset, the IFN/ENIT database (www.ifnenit.com), became available free for non commercial research (Pechwitz et al., 2002).

The IFN/ENIT database contains 946 handwritten Tunisian town/village names and their corresponding postcodes. In version v1.0p2, the database consists of 26459 Arabic names handwritten by 411 different people. These names consist of 115000 pieces of Arabic words (PAW, or subwords) and about 212000 characters. In a newer version v2.0p1e, one additional set e containing 6033 names handwritten by 87 writers was added, which makes the whole set to have 32492 name samples.

All the handwritten forms were scanned with 300dpi and converted to binary images. Each handwritten name comes with a binary image with relevant ground truth. Each ground truth entry contains the following information: i) text for the image, ii) postcode, iii) character shape sequence, iv) locations of up to two baselines, v) baseline quality, vi) quantity of words, vii) quantity of PAWs, viii) quantity of characters, and ix) writing quality.

For training and testing purposes, the whole IFN/ENIT dataset is partitioned into four subsets (a-d) in v1.0p2 and five subsets (a-e) in v2.0p1e, respectively, where normally the test set is unknown, not used for training, when a system is tested for evaluation. This enables cross validation to be applied for performance evaluation. Unlike some systems such as (Kessentini et al 2008) and (El-Hajj et al. 2005) in which only a small part of the database is used, we apply our approach to the whole dataset for evaluations. Relevant experiments and results are presented in the next two subsections.

5.2 Experiments on IFN/ENIT database v2.0p1e

In this group of experiments, four subsets (a-d) are used for training the HMM and (e) for testing. To determine an optimal codebook size for HMM, we compare the recognition rate under various codebook sizes and the results are summarized in Table 2. Possible codebook sizes are specified as 8, 16, 32, 64, and 128, respectively. As seen in Table 2, better recognition rate is yielded by increased codebook size, yet it takes longer time for training and testing the HMM classifier. In addition, it is found that the system reaches its saturation while the codebook size becomes 64 and more. As a result, an optimal codebook size is set as 64 to achieve a good tradeoff between high recognition rate and low time factor. Furthermore, it is worth noting that our re-ranking scheme helps to improve the recognition rate. In fact, it contributes 0.43%-1.34% to top 1 recognition rate and about 0.84%-3.23% for top 10. This validates the effectiveness of such re-ranking scheme for our task.

Similarly, an optimal number of states used in HMM is also determined empirically. Using possible numbers varying equally from 10 to 30, the recognition rates obtained are listed in Table 3 for comparisons. It has been noted that the recognition rate improves as the number of states increases till the HTK reaches the maximum possible state for specific feature set. This makes the training data is independent of the testing data, and hence avoid over-fitting the classifier to test the data. In our case, as seen in Table 3, the optimal number of states is found as 25. Again, we can see obvious improvements in terms of recognition rate when re-ranking scheme is used.

Furthermore, the performance of our system is compared with six others in ICDAR 2005 Arabic handwriting competition (Margner et al 2005). Using the same datasets for training and testing, relevant results are compared in Table 4. Please note that the test set e is unknown to participants during the competition, and testing results are produced using systems submitted to the organizer. Also note that the results from #5 system is incomplete, as it only tested on a subset due to data failure. Details about the competition and techniques used in each participated team can be found in (Margner et al 2005). As seen from Table 4, the top 1 recognition rate of our proposed approach is 83.55% if re-ranking is used, or 82.32% if not. On the contrary, the best result from others has a top-1 recognition rate of 75.93%. This shows that our system outperforms others over 7.6% (or 6.4% without re-ranking) in terms of top 1 recognition rate.

5.3 Experiments on IFN/ENIT database v1.0p2

As discussed in Section 3, four-subset version of IFN/ENIT database has also been widely adopted in many systems. To enable consistent performance evaluation, we apply our system on this version of datasets and compare the results in Table 5. As seen in Table 5, in total 20 groups of results from 9 systems are listed for comparisons. From Table 5, several observations can be made and summarized as follows.

When a single HMM classifier is used, the best top 1 recognition rate is achieved at 89.74% by (Pechwitz & Maergner, 2003) when baseline information from the ground truth is used. The recognition rate is degraded to 83.56% or 81.84% when baseline is estimated using skeleton or projection based techniques, respectively. Our system with re-ranking produces the second best top-1 recognition rate at 89.24%, though this reduces to 86.73% if such re-ranking is absent. The work in (ElAbed and Magner, 2007) generates almost the same good results as ours with a top-1 recognition rate of 89.10%. However, its top 10 recognition rate at 96.4% is the highest among all others. In contrast, the top 10 recognition rates from our system and (Pechwitz & Maergner, 2003) are 95.15% and 94.98%, respectively.

For multi-classifier cases, the work in (Dreuw et al 2008) is the best with a top-1 recognition rate of 92.86%. This is due to two main techniques namely character model length adaptation (MLA) and support of additional virtual training samples (MVT) on the base of their interesting white-space models, where HMMs of different topologies are applied in character and white-space models. Using a hybrid HMM/NN classifier, HMM is used to represent each letter-body, whilst NN is employed to compute the observations probability distribution (Menasri et al 2007). When three different letter models are used, the best recognition rate achieved is 87.4% for top 1 and 96.9% for top 10. Although (Al-Hajj et al. 2009) yields slightly worse recognition rates using single HMM, 87.60% for top 1 and 93.76% for top 10, improved results are produced using their combined approach through fusion of three HMMs. Under three combination strategies including sum, majority vote and multi-layer perception (MLP), the top 1 recognition rate achieved are 90.61%, 90.26% and

90.96%, respectively. Accordingly, the top 10 recognition rates are 95.87%, 95.68% and 94.44%. This on one hand shows that combined classifier indeed produces much improved top 1 classification rate. On the other hand, it seems that such combination does not necessarily ensure a high top 10 rate. One possible reason is that top 1 rate is the first priority when a combined strategy is designed. In addition, the best results from (Dreuw et al 2008) suggest that modeling of characters has great potential in correctly recognizing words.

Furthermore, it is worth noting that the results from our approach with re-ranking are among the best in Table 5, although ground truth information like baseline location and fusion of multiple classifiers are not used. Thanks to the re-ranking scheme, it has successfully improved the recognition rate yet avoided bringing much additional complexity to the algorithm.

5.4 Error analysis

Like all other systems, the proposed approach also has a certain level of error rate. Actually, our system with re-ranking has an error rate of 16.45% for tests using version v2.0p1e of the database, and this reduces to 10.76% if version v1.0p2 of the database is used. In fact, the main reasons for these errors can be summarized as follows.

The first is inconsistency within the captured handwritten samples, which includes not only variations in shape and size, but also presence or absence of diacritical marks. As discussed in Section 2, diacritical marks are essential in distinguishing ambiguity between words, yet they can be skipped or put in various forms in handwritten. If one word contains samples in various writing styles/forms or different words share one similar shape, it inevitably leads to misclassification. Consequently, spelling check might be useful to solve this problem for improved accuracy (Khorsheed, 2003).

The second is unbalanced occurrence of samples in the database, as this number varies from 3 to 381 (EL-HAJJ et al 2005). When one word has very limited samples, dividing them into different subsets affects its correct recognition, especially when the sample in test set appears differently from the one (or even absent) in the training sets. Taking the database of version v2.0p1e for example, Fig. 5 plots frequency vs. number of PAWs from both the training and test sets. As seen, there is apparent inconsistency between training and testing sets, which may lead to inaccurate modelling and low recognition rate. In addition, insufficient samples also lead to unreliable estimate of the re-ranking function, as both the mean and standard deviation for re-ranking cannot be accurately determined. Basically, more biased the samples are distributed in the test set against the whole database, more likely a higher error rate is generated. As shown in Table 6, the number of words for testing contained in test set (d) in database version v1.0p2 and test (e) in v2.0p1e are quite similar, i.e. 6735 vs. 6033. However, the degree of their biased distributions, as defined in (8), is different.

$$\eta = u_w / u_t . \tag{8}$$

where η is the biased degree, u_w and u_t respectively refer to number of writers in the whole set and the test set. In our cases, the biased degrees for database versions v1.0p2 and v2.0p1e are determined as 3.95 and 11.49. Obviously, the distribution of test set in v2.0p1e is more biased. This further explains why tests using database of version v1.0p2 yield higher recognition rate than those using version v2.0p1e.

The third is potential errors in pre-processing in terms baseline detection and word segmentation, as such errors will be propagated and lead to inexact feature extraction due to wrong word boundary and/or inaccurate extraction of topological features. Certainly using some information provided by the ground truth, such as baseline location, can improve the overall performance (Pechwitz & Maergner, 2003). However, in our system such information is not employed, as we aim to develop a generic system where ground truth is unavailable.

6. Conclusions

We have proposed a combined scheme for Arabic handwritten word recognition, using a HMM classifier followed by re-ranking. Basically, intensity features are used to train the HMM, and topological features are used for re-ranking for improved accuracy. Using the IFN/ENIT database, the performance of our proposed method is compared with quite a few state-of-art techniques, including those in ICDAR 2005 competition and several recently published ones. Although the best results are generated by using fusion of multiple HMMs, the results of our proposed approach are among the best when a single HMM classifier is used. However, ground truth information like baseline location is not employed in our system, which enables it to be applied for more generic applications. In addition, it is worth noting that with slightly adaptation the proposed techniques can be applied to other pattern recognition tasks. Further investigations include more accurate pro-processing such as subword segmentation and dots detection for more effective re-ranking as well as to apply other classifiers like dynamic Bayesian networks (DBN).

7. References

Abdulkadr, A., 2006. Two-tier approach for Arabic offline handwriting recognition. *Proc. 10th Int. Workshop on Frontiers in Handwriting Recognition* (IWFHR), pp. 161-166.

Al-Hajj, M. R., Likforman-Sulem, L. & Mokbel, C., 2009. Combining slanted-frame classifiers for improved HMM-based Arabic handwriting recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 31**,** pp. 1165-1177.

Al-Hajj, M. R., Mokbel, C., & Likforman-Sulem, L., 2007. Combination of HMM-based classifiers for the recognition of Arabic handwritten words. *Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR).*

Alkhateeb, J. H., Jiang, J., Ren, J., Khelifi, F., and Ipson, S. S., 2009a. Multiclass classification of unconstrained handwritten Arabic words using machine learning approaches. *The Open Signal Processing Journal*, 2(1), pp. 21-28.

Alkhateeb, J. H., Ren, J., Ipson, S. S. & Jiang, J., 2008. Knowledge-based baseline detection and optimal thresholding for words segmentation in efficient pre-processing of handwritten Arabic text. *Proc. 5th Int. Conf. Information Technology: New Generations (ITNG)*

Alkhateeb, J. H., Ren, J., Ipson, S. S. & Jiang, J., 2009b. Component-based segmentation of words from handwritten Arabic text. *Int. J. Computer Systems Science and Engineering,* 5(1).

Alkhateeb, J. H., Ren, J., Jiang, J., and Ipson, S. S., 2009c. A machine learning approach for offline handwritten Arabic words. *Proc. Cyber Worlds*.

Alkhateeb, J. H., Ren, J., Jiang, J., and Ipson, S. S., 2009d. Unconstrained Arabic handwritten word feature extraction: a comparative study. *Proc. 6th Int. Conf. Information Technology: New Generations (ITNG)*

Alma'Adeed, S., Higgins, C., and Elliman, D., 2004. Off-line recognition of handwritten Arabic words using multiple hidden Markov models. *Knowledge-Based Systems,* 17**,** pp. 75-79.

Amin, A., 1998. Off-line Arabic character recognition: the state of the art. *Pattern Recognition*, 31**,** pp. 517-530.

Amin, A., Al-Sadoun, H., and Fischer, S., 1996. Hand-printed Arabic character recognition system using an artificial network. *Pattern Recognition*, 29**,** pp. 663-675.

Ball, G. R., 2007. Arabic Handwriting Recognition using Machine Learning Approaches. *Ph.D. thesis*, The Faculty of Graduate School of State University of New York at Buffalo.

Benouareth, A., Ennaji, A., and Sellami, M., 2006. HMMs with explicit state duration applied to handwritten Arabic word recognition. IN ENNAJI, A. (Ed.) *18th Int. Conf. Pattern Recognition (ICPR).*

Benouareth, A., Ennaji, A. and Sellami, M., 2008. Semi-continuous HMMs with explicit state duration for unconstrained Arabic word modeling and recognition. *Pattern Recognition Letters,* 29**,** pp. 1742-1752.

Dreuw, P., Jonas, S., and Ney, H., 2008. White-space models for offline Arabic handwriting recognition. *Proc. 19th Int. Conf. Pattern Recognition* (ICPR).

El-Hajj, R., Likforman-Sulem, L., and Mokbel, C., 2005. Arabic handwriting recognition using baseline dependant features and hidden Markov modeling. *Proc. 8th Int. Conf. Document Analysis and Recognition (ICDAR).*

El-Abed, H., and Margner, V., 2007. Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten Arabic words. *Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR'07).*

Graves, A. and Schmidhuber, J., 2008. Offline handwriting recognition with multidimensional recurrent neural networks. *Proc. 22nd Conf. Neural Information Processing Systems* (NIPS).

Gunter, S., and Bunke, H., 2004. HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components. *Pattern Recognition,* 37**,** pp. 2069-2079.

Husni, A. A.-M., Sabri, A. M., and Rami, S. Q., 2008. Recognition of off-line printed Arabic text using Hidden Markov Models. *Signal Process.*, 88**,** pp. 2902-2912.

Kessentini, Y., Paquet, T., and Benhamadou, A. M., 2008. Multi-script handwriting recognition with n-streams low level features. *Proc. 19th Int. Conf. Pattern Recognition* (ICPR)

Khorsheed, M. S., 2000. Automatic Recognition of Words in Arabic Manuscripts. *Computer Laboratory.* University of Cambridge.

Khorsheed, M. S., 2002. Off-Line Arabic Character Recognition – A Review. *Pattern Analysis & Applications,* 5**,** pp. 31-45.

Khorsheed, M. S., 2003. Recognising handwritten Arabic manuscripts using a single hidden Markov model. *Pattern Recognition Letters,* 24**,** pp. 2235-2242.

Khorsheed, M. S., and Clocksin, W. F., 1999. Structural features of cursive Arabic script. *Proc. 10th British Machine Vision Conf.* The University of Nottingham, UK.

Lorigo, L. M., and Govindaraju, V., 2006. Offline Arabic handwriting recognition: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 28**,** pp. 712-724.

Madhvanath, S., and Govindaraju, V., 2001. The role of holistic paradigms in handwritten word recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 23**,** pp. 149-164.

Margner, V., Pechwitz, M., and Abed, H. E., 2005. ICDAR 2005 Arabic handwriting recognition competition. In Pechwitz, M. (Ed.) *Proc. 8th Int. Conf. Document Analysis and Recognition (ICDAR).*

Menasri, F., Vincent, N., Augustin, E., and Cheriet, M., 2007. Shape-based alphabet for off-line Arabic handwriting recognition. *Proc. 9th Int. Conf. Document Analysis and Recognition* (ICDAR), 2, pp. 969-973.

Parker, J. R., 1997. *Algorithms for Image Processing and Computer Vision* John Wiley and Sons, Inc

Pechwitz, M., Maddouri, S. S., Maergner, V., Ellouze, N., and Amiri, H., 2002. IFN/ENIT - database of Arabic handwritten words. *Colloque International Franco-phone sur l'Ecrit et le Document (CIFED).*

Pechwitz, M., and Maergner, V., 2003. HMM based approach for handwritten Arabic word recognition using the IFN/ENIT - database. IN MAERGNER, V. (Ed.) *Proceedings Seventh International Conference on Document Analysis and Recognition.*

Prasad, R., Bhardwaj, A., Subramanian, K., Cao, H., and Natarajan, P., 2010. Stochastic segment model adaptation for offline handwriting recognition. *Proc. Int. Conf. Pattern Recognition,* pp. 1993-1996.

Rabiner, L. R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE,* 77, 257-286.

Saleem, S., Cao, H., Subramanian, K., Kamali, M., Prasad, R., Natarajan, P., 2009. Improvements in BBN's HMM-based offline Arabic handwriting recognition system. *Proc. 10th Int. Conf. on Document Analysis and Recognition* (ICDAR).

Young, S., Evermann, G., Kershaw, D., Moore, G., Odeli, J., Ollason, D., Valtchev, V., and Woodland, P., 2001. *The HTK Book*, Cambridge University Engineering Department.

Zavorin, I., Borovikov, E., Davis, E., Borovikov, A., and Summers, K., 2008. Combining different classification approaches to improve off-line Arabic handwritten word recognition. *Proc. of SPIE,* 6815(681504).