Window repositioning for Printed Arabic Recognition

Ihab Khoury, Adrià Giménez, Alfons Juan and Jesús Andrés-Ferrer
Universitat Politècnica de València, Camí de Vera s/n, 46022 València, Spain

Abstract

Bernoulli HMMs are conventional HMMs in which the emission probabilities are modeled with Bernoulli mixtures. They have recently been applied, with good results, in off-line text recognition in many languages, in particular, Arabic. A key idea that has proven to be very effective in this application of Bernoulli HMMs is the use of a sliding window of adequate width for feature extraction. This idea has allowed us to obtain very competitive results in the recognition of both Arabic handwriting and printed text. Indeed, a system based on it ranked first at the ICDAR 2011 Arabic recognition competition on the Arabic Printed Text Image (APTI) database. More recently, this idea has been refined by using *repositioning* techniques for extracted windows, leading to further improvements in Arabic handwriting recognition. In the case of printed text, this refinement led to an improved system which ranked second at the ICDAR 2013 second competition on APTI, only at a marginal distance from the best system. In this work, we describe the development of this improved system. Following evaluation protocols similar to those of the competitions on APTI, exhaustive experiments are detailed from which

Email address: {ialkhoury,agimenez,ajuan,jandres}@dsic.upv.es (Ihab Khoury, Adrià Giménez, Alfons Juan and Jesús Andrés-Ferrer)

state-of-the-art results are obtained.
Keywords: Bernoulli HMMs, Printed Arabic Recognition, Sliding Window, Repositioning

1. Introduction

Hidden Markov Models (HMMs) are now widely used for off-line text recognition in many languages, in particular, languages with Arabic script (Dehghan et al., 2001; Günter and Bunke, 2004; Märgner and El Abed, 2007, 2009; Grosicki and El Abed, 2009). Following the conventional approach in speech recognition (Rabiner and Juang, 1993), HMMs at global (line or word) level are built from shared, *embedded*, HMMs at character (subword) level, which are usually simple in terms of number of states and topology. In the common case of real-valued feature vectors, state-conditional probability (density) functions are modeled as Gaussian mixtures since, as with finite mixture models in general, their complexity can be easily adjusted to the available training data by simply varying the number of components.

After decades of research in speech recognition, the use of certain real-valued speech features and embedded Gaussian (mixture) HMMs is a de-facto standard (Rabiner and Juang, 1993). However, in the case of text recognition there is no such standard. In fact, very different sets of features are in use today. In (Giménez and Juan, 2009) we proposed to by-pass feature extraction and directly feed columns of raw, binary pixels into *embedded Bernoulli (mixture) HMMs (BHMMs),* that is, embedded HMMs in which the emission probabilities are modeled with Bernoulli mixtures. The basic idea is to ensure that no discriminative information is filtered out during feature

extraction, which in some sense is integrated into the recognition model. In (Giménez et al., 2010), we improved our basic approach by using a sliding window of adequate width to better capture image context at each horizontal position of the text image. This improvement, to which we refer as *windowed BHMMs,* achieved very competitive results on the well-known IfN/ENIT database of Arabic town names (Märgner and El Abed, 2010). More recently, very good results on the Arabic Printed Text Image (APTI) database were also achieved using the same approach, which ranked first in the ICDAR 2011 Arabic recognition competition for printed Arabic text (Slimane et al., 2011).

Although windowed BHMMs achieved good results on IfN/ENIT and APTI, it was clear to us that text distortions are more difficult to model with wide windows than with narrow (e.g. one-column) windows. In order to circumvent this difficulty, we have considered new, adaptive window sampling techniques, as opposed to the conventional, direct strategy in which the sampling window center is applied at a constant height of the text image and moved horizontally one pixel at a time. More precisely, these adaptive techniques can be seen as an application of the direct strategy followed by a *repositioning* step by which the sampling window is repositioned to align its center to the center of gravity of the sampled image. This repositioning step can be done horizontally, vertically or in both directions. Although vertical repositioning is expected to have more influence on recognition results than horizontal repositioning, we have studied both separately and in conjunction, so as to confirm this expectation.

In (Giménez et al., 2014b), the repositioning techniques described above

are introduced and extensively tested on different databases for off-line handwriting recognition. As expected, vertical repositioning provides excellent results, not only on IfN/ENIT, but also on other well-known databases such as IAM words and RIMES. In the case of printed text, the use of repositioning techniques has allowed us to significantly improve our system at the ICDAR 2011 first competition on APTI. Indeed, our improved system obtained much better results at the ICDAR 2013 second competition on APTI, in which it ranked second at a marginal distance from the first (Slimane et al., 2013). In this work, we describe the development of this improved system. Following evaluation protocols similar to those of the competitions on APTI, exhaustive experiments are described from which state-of-the-art results are obtained.

In what follows, we first review BHMMs (Sec. 2). Then, we describe the approach through which we are achieving the best results: windowed BHMMs with repositioning (Sec. 3) and its use for printed Arabic recognition by application of the Bayes decision rule (Sec. 4). In Sec. 5, we provide the results of a complete series of experiments on APTI as well as a comparison with results from other authors on this database. Finally, concluding remarks are given in Sec. 6.

2. Bernoulli HMMs

Let $O = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ be a sequence of feature vectors. An HMM is a probability (density) function of the form:

$$P(O \mid \mathbf{\Theta}) = \sum_{q_1,\ldots,q_T} \prod_{t=0}^{T} a_{q_t q_{t+1}} \prod_{t=1}^{T} b_{q_t}(\mathbf{o}_t)\,, \tag{1}$$

where the sum is over all possible *paths* (state sequences) $q_0, \ldots, q_{T+1}$, such that $q_0 = I$ (special *initial* or *start* state), $q_{T+1} = F$ (special *final* or *stop* state), and $q_1, \ldots, q_T \in \{1, \ldots, M\}$, being M the number of regular (non-special) states of the HMM. On the other hand, for any regular states i and j, a_{ij} denotes the *transition* probability from i to j, while b_j is the *observation* probability (density) function at j.

A Bernoulli (mixture) HMM (BHMM) is an HMM in which the probability of observing a binary feature vector $\mathbf{o}_t$, when $q_t = j$, follows a Bernoulli mixture distribution for the state j

$$b_j(\mathbf{o}_t) = \sum_{k=1}^{K} \pi_{jk} \prod_{d=1}^{D} p_{jkd}^{o_{td}} (1 - p_{jkd})^{1-o_{td}} , \tag{2}$$

where o_{td} is the d-th bit of $\mathbf{o}_t$, π_{jk} is the prior of the k-th mixture component in state j, and p_{jkd} is the probability that this component assigns to o_{td} to be 1.

As discussed in the introduction, BHMMs at global (line or word) level are built from shared, embedded BHMMs at character level. More precisely, let C be the number of different characters (symbols) from which global BHMMs are built, and assume that each character c is modeled with a different BHMM of parameter vector $\mathbf{\Theta}_c$. Let $\mathbf{\Theta} = \{\mathbf{\Theta}_1, \ldots, \mathbf{\Theta}_C\}$, and let $O = (\mathbf{o}_1, \ldots, \mathbf{o}_T)$ be a sequence of feature vectors generated from a sequence of symbols $S = (s_1, \ldots, s_L)$, with $L \leq T$. The probability of O can be calculated, using embedded HMMs for its symbols, as:

$$P(O \mid S, \mathbf{\Theta}) = \sum_{i_1, \ldots, i_{L+1}} \prod_{l=1}^{L} P(\mathbf{o}_{i_l}, \ldots, \mathbf{o}_{i_{l+1}-1} \mid \mathbf{\Theta}_{s_l}) , \tag{3}$$

where the sum is carried out over all possible segmentations of O into L

segments, that is, all sequences of indices $i_1, \ldots, i_{L+1}$ such that

$$1 = i_1 < \cdots < i_L < i_{L+1} = T + 1;$$

and $P(\mathbf{o}_{i_l}, \ldots, \mathbf{o}_{i_{l+1}-1} \mid \mathbf{\Theta}_{s_l})$ refers to the probability (density) of the l-th segment, as given by (1) using the HMM associated with symbol s_l.

Maximum likelihood estimation (MLE) of BHMM parameters does not differ significantly from the conventional Gaussian case, and it can be efficiently performed using the well-known EM (Baum-Welch) re-estimation formulae (Rabiner and Juang, 1993; Young et al., 1995). Please see (Giménez et al., 2014b) for more details. Also as in the conventional Gaussian case, BHMM parameters can be estimated by discriminative training (Giménez et al., 2014a).

3. Windowed BHMMs with repositioning

Given a binary image normalized in height to H pixels, we may think of a feature vector $\mathbf{o}_t$ as its column at position t or, more generally, as a concatenation of columns in a window of W columns in width, centered at position t. This generalization has no effect neither on the definition of BHMM nor on its MLE, although it might be very helpful to better capture the image context at each horizontal position of the image. As an example, the first row in Fig. 1 shows a binary image of 4 columns and 5 rows, which is transformed into a sequence of four 15-dimensional feature vectors by application of a sliding window of width 3. For clarity, feature vectors are depicted as 3×5 subimages instead of 15-dimensional column vectors. Note that feature vectors at positions 2 and 4 would be indistinguishable if, as in our previous approach, they were extracted with no context ($W = 1$).

Although one-dimensional, "horizontal" HMMs for image modeling can properly capture non-linear horizontal image distortions, they are somewhat limited when dealing with vertical image distortions, and this limitation might be particularly strong in the case of feature vectors extracted with significant context. To overcome this limitation, we have considered three methods of window *repositioning* after window extraction: *vertical, horizontal,* and *both.* The basic idea is to first compute the center of mass of the extracted window, which is then repositioned (translated) to align its center to the center of mass. This is done in accordance with the chosen method, that is, horizontally, vertically, or in both directions. Obviously, the feature vector actually extracted is that obtained after repositioning. An example of feature extraction is shown in Fig. 1 in which the standard method (no repositioning) is compared with the three methods repositioning methods considered.

It is helpful to observe the effect of repositioning with real data. Fig. 2 shows the sequence of feature vectors extracted from a real sample of the APTI database, with and without (both) repositioning. As expected, (vertical or both) repositioning has the effect of normalizing vertical image distortions, especially translations.

4. Bernoulli HMMs for printed Arabic recognition

Given an observation O of unknown class, we use the Bayes decision rule to classify O into the class to which it belongs with highest *(posterior)* probability or, equivalently:

$$c^*(O) \;=\; \underset{c=1,\ldots,C}{\operatorname{argmax}} \quad \log P(c) + \log P(O \mid c) \tag{4}$$

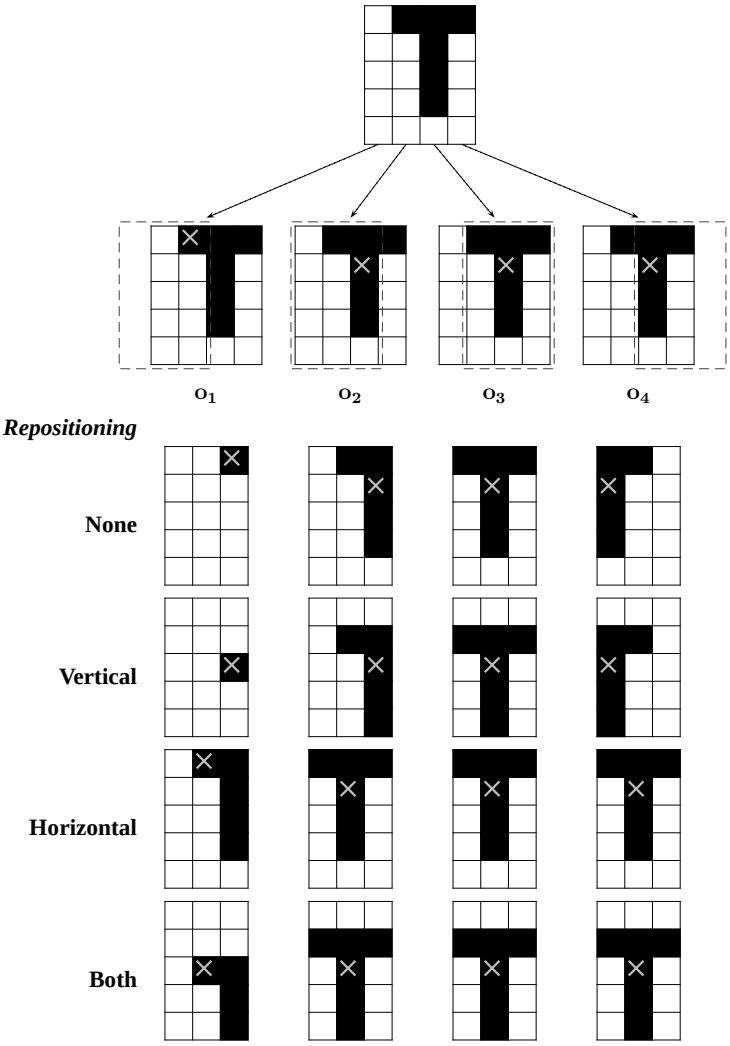

Figure 1: Example of transformation of a 4×5 binary image (top) into a sequence of 4 15-dimensional binary feature vectors $O = (\mathbf{o}_1, \mathbf{o}_2, \mathbf{o}_3, \mathbf{o}_4)$ using a window of width 3. After window extraction (illustrated under the original image), the standard method (no repositioning) is compared with the three repositioning methods considered: vertical, horizontal, and both directions. Mass centers of extracted windows are also indicated.

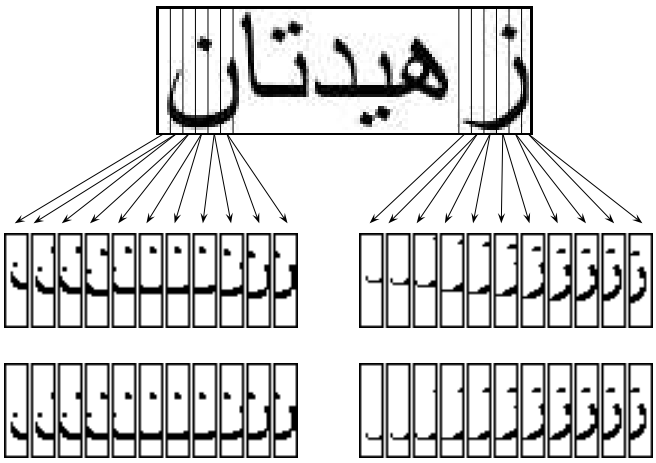

Figure 2: Original sample *Image_18_ArabicTransparent_5111* from set1 from APTI database (top) and its sequence of feature vectors produced with and without (both) repositioning (center and bottom, respectively).

where C is the total number of classes and, for each class $c = 1, \ldots, C$, $P(c)$ is its *prior* probability and $P(O \mid c)$ is the class-conditional probability (density) for O to come from class c.

Class priors and class-conditional probability (density) functions are usually estimated from a set of *training* observations. The conventional approach to estimate class priors is simply to compute their relative frequencies from the training set. However, the estimation of class-conditional probability (density) functions is more involved and depends on the type of representation space for the observations. Usually, each class-conditional probability (density) function is modeled by an appropriate parametric function whose parameters are estimated by MLE from the training data. As an example, consider the problem of classifying images of *isolated printed Arabic characters.* The number of classes is modest and it is not difficult to collect many

training examples for each class. Therefore, class priors can be accurately estimated by the conventional method. Also, if images are represented as sequences of feature vectors, each class-conditional probability function can be modeled by an independent BHMM (Eqs. (1) and (2)) with parameters estimated by MLE from training observations of its class (Giménez et al., 2014b).

The above approach for the estimation of class priors and class-conditional probability (density) functions is no longer applicable to classification problems with large number of classes due to the lack of training data for each class. Consider, as we do in this work, the problem of classifying images of *printed Arabic words*. Collecting a number of training observations for each word will be really difficult if we are interested in recognizing a large number of different words. Indeed, it will be impossible if we are interested in building an *open-vocabulary* recognizer, that is, one even able to recognize words not "seen" (with no observations) in the training data. As with Arabic handwriting recognition in general, the usual approach in this case consists in using global (word) models defined in terms of local (subword) models. This is the approach followed in this work. Formally, given an observation O of an unknown word, we use Eq. (4) to decide to which word corresponds:

$$w^*(O) \;=\; \underset{w}{\operatorname{argmax}} \quad \log P(S_w) + \log P(O \mid S_w, \boldsymbol{\Theta}) \tag{5}$$

where, for each word w, S_w is its sequence of symbols (characters), $P(S_w)$ is its prior probability and $P(O \mid S_w, \boldsymbol{\Theta})$ is the probability for O to be generated from a BHMM for w (Eq. 3). Word priors are modeled with n-gram language models at character level (Jelinek, 1997). Word-conditional probability functions are modeled by BHMMs built from shared, embedded

BHMMs at character level (Eq. (3)) with parameters trained by MLE.

Clearly, the direct way to measure the error of a word recognizer is to count the (relative) number of misclassified observations in a collection of *test* observations (i.e. samples held out during training). In what follows, this is referred to as the Word Error Rate (WER). Apart from the WER, we also use the Character Error Rate (CER), that is, the (relative) number of misclassified characters. In practice, the CER can be considered equivalent to the WER for comparison purposes.

5. Experiments

As indicated in the introduction, in this Section we provide the results of a complete series of experiments on APTI as well as a comparison with results from other authors on this database. APTI is briefly described in Section 5.1 together with its basic preprocessing for the experiments below. Then, two experimental protocols are defined in Section 5.2, UPVPC1 and UPVPC2, whose results are reported separately in Sections 5.3 and 5.4 respectively. Finally, the idea of vertical repositioning is also tried on recent state-of-the-art techniques based on neural networks in Section 5.5.

5.1. APTI database and preprocessing

The Arabic Printed Text Image (APTI) database is a collection of images of Arabic Printed words. It was recently published by (Slimane et al., 2009) for large-scale benchmarking of open-vocabulary, multi-font, multi-size and multi-style text recognition systems in Arabic. It consists of 113284 different single words, each one available in 10 different fonts, 10 different font sizes, and also 4 different styles.

APTI is divided into six equilibrated sets ($set1$, $set2$, ..., $set6$) to allow for flexibility in the design of experimental protocols. Each set has different words, but characters are equally distributed. The five first sets are available for the scientific community. The sixth set is kept by the authors for future evaluation of systems in blind mode.

For the experiments reported below, APTI was preprocessed by scaling all images in the first five sets to a height of D pixels (for 10 different values of D from 30 to 50) while keeping the aspect ratio. Scaled images were then binarized by application of the Otsu's method (Otsu, 1979).

5.2. Experimental protocols: UPVPC1 and UPVPC2

APTI was used first in the Arabic Recognition Competition of ICDAR 2011 (Slimane et al., 2011). Two experimental protocols were defined which differ in the number of fonts used: APTIPC1 and APTIPC2. In APTIPC1, only the Arabic Transparent font was used. In APTIPC2, however, five different fonts were used: Arabic Transparent (Trans), Andalus (Anda), Diwani Letter (Diw), Simplified Arabic (Simp), and Traditional Arabic (Trad). In both protocols, only the *Plain* font style was used, with sizes of 6, 8, 10, 12, 18 and 24 pixels. As indicated above, the first five sets were available to participants for system training, while the sixth set was held-out by the organizers for system comparison in blind mode.

In this paper, we could not use the training-test partition used at the ICDAR 2011 competition because the sixth set is not publicly available. Instead, we used the first four sets for training and the fifth set for testing. More precisely, we defined two new protocols: UPVPC1 and UPVPC2. In UPVPC1, 13000 images from the first four sets were randomly drawn (10000

for training and 3000 for testing). In UPVPC2, we used the whole first four sets for training and the whole fifth set for testing. In particular, we used 2266500 images for training, and 566040 for testing.

5.3. Results using the UPVPC1 protocol

For (computational) simplicity, the UPVPC1 protocol was used in a first series of experiments to study the effect on the CER of various key parameters. We began with experiments for font size 6, which were then extended to other font sizes. In particular, for each dimension D in $\{30, 32, \ldots, 50\}$, each sliding window width W in $\{1, 3, \ldots, 11\}$, each number of states Q in $\{4, 5, 6, 7, 8\}$ and each number of mixture components K in $\{1, 2, 4, ..., 32\}$, a BHMM-based word recognizer was trained from the training data of font size 6 in the UPVPC1 protocol. For $K = 1$, BHMMs were initialized by first segmenting training data with a "neutral" model, and then using the resulting segments to perform a Viterbi initialization. Initialized BHMMs were then trained with 4 EM iterations. For $K > 1$, BHMMs were initialized by splitting the mixture components of the models trained with $K/2$ mixture components per state. Again, after initialization, BHMMs were trained with 4 EM iterations. On the other hand, word priors were modeled with 5-gram language models at character level.

The above training procedure led to a different recognizer for each combination of key parameter values (apart from the font size itself). Each of them was of the form given by Eq. (5) though, as usual in (Arabic) text recognition, a *Grammar Scale Factor (GSF)* was used to adjust the importance of class priors with respect to word-conditional observation probabilities (i.e. the GSF is a constant multiplier for log-priors). For each combination of parameter

values and each value of $GSF \in \{20, 30, 40, 50\}$, the corresponding recognizer was assessed in terms of CER from the test data of font size 6 in the UPVPC1 protocol.

Figure 3 shows the CER as a function of D (top left), K (top right), Q (bottom left) and GSF (bottom right); for $W = 1$, 3, 7 and 11 (the curves for $W = 5$ and 9 are similar and have been omitted for clarity). Each plotted point shows the best CER obtained over all values tried for the parameters not given. The best CER obtained is 3.4% for $D = 38$, $W = 7$, $Q = 7$, $K = 32$ and $GSF = 50$. In the plot at the top left, it is shown for $D = 38$ and $W = 7$, as the minimum CER obtained for all values tried for Q, K and GSF.

From the results in Fig. 3, it is clear that the use of windowed BHMMs is of crucial importance. Indeed, the best CER obtained with no windows ($W = 1$) is 6.6%; i.e. it nearly doubles the best CER with windows. Note also that, as W, the number of mixtures components (K) has a strong effect on the CER. The best error rates were obtained with the maximum value of K tried (32). Therefore, this and larger values of K need to be tried in further experiments with more training data. The dimension (D), number of states (Q) and GSF are also key parameters to be adjusted, though Fig. 3 does not show wide fluctuations in CER for the ranges of values considered.

As discussed in (Dreuw et al., 2009), letters in Arabic script differ significantly in length, and thus it might not be appropriate to model all of them using BHMMs of fixed number of states. With this idea in mind, an experiment similar to that described above was carried out for $D = 38$, $W = 7$, $K = 32$, $GSF = 50$ and variable number of states. To decide the number of

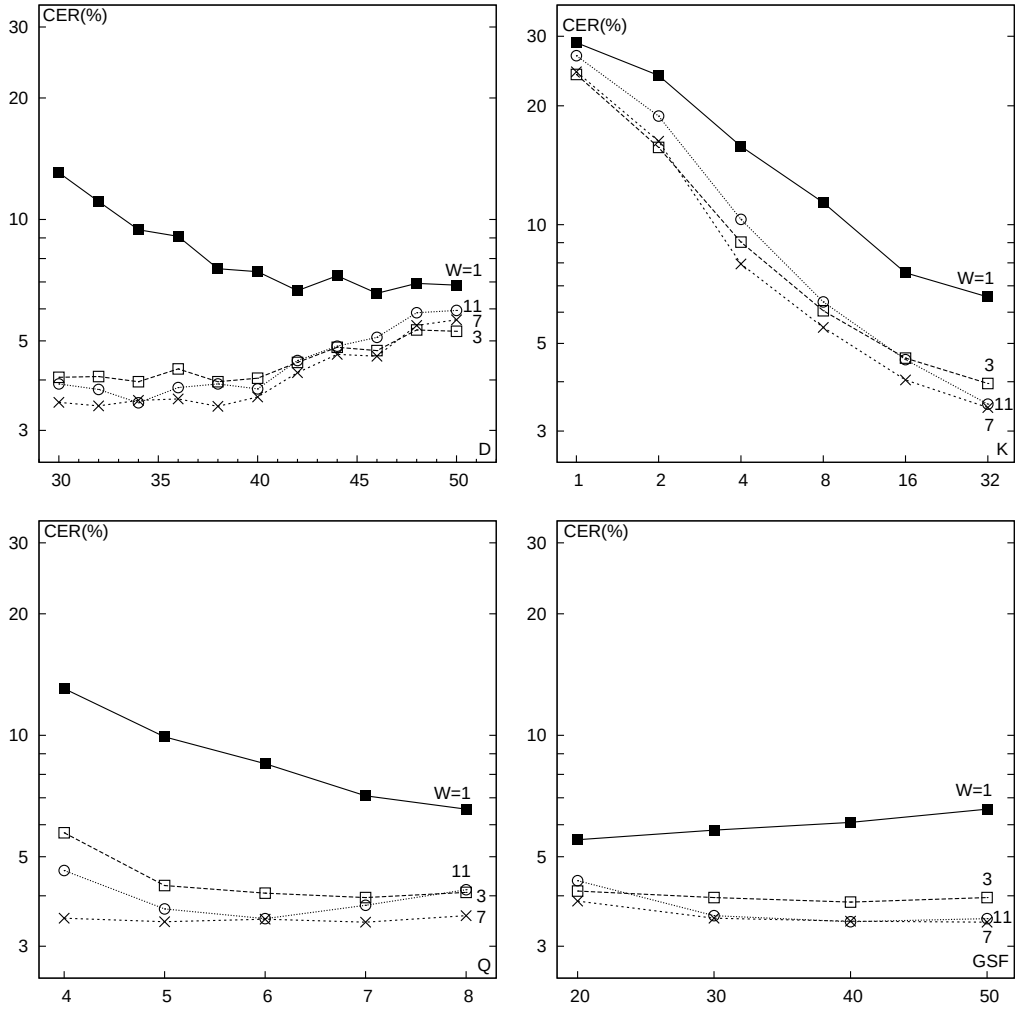

Figure 3: CER(%) as a function of the dimension D (top left), number of mixture components K (top right), number of states Q (bottom left) and GSF value (bottom right); for sliding window widths of $W = 1$, 3, 7 and 11.

states for each character, we first Viterbi-segmented all training data using BHMMs of 7 states, and then computed the average length of the segments associated with each character. Given an average segment length for character c, $\bar{T}_c$, its number of states was set to $F \cdot \bar{T}_c$, where F is a *factor* measuring the average number of states that are required to emit a feature vector. Thus, its inverse, $\frac{1}{F}$, can be interpreted as a *state load,* that is, the average number of feature vectors that are emitted in each state. For instance, $F = 0.2$ means that only a fraction of 0.2 states is required to emit a feature vector or, alternatively, that $\frac{1}{0.2} = 5$ feature vectors are emitted on average in each state. We tried all values of F in $\{0.1, 0.2, \ldots, 0.9\}$. The best result achieved is a CER of 3.2%, using $F = 0.5$, which is significantly better than the best result obtained above with fixed number of states (3.4%).

To complete our experiments with font size 6 data in the UPVPC1 protocol, the best recognizer found above was also tested with the four repositioning methods described in Sec. 3. As expected, the best CER, 1.1%, was obtained with *vertical* repositioning alone. Also as expected, it was similar to the CER achieved with repositioning in *both* directions (1.2%), and significantly better than those obtained with *horizontal* and *no* repositioning (3.2% for both).

The experiments described above in this Section were extended to all font sizes. More precisely, for each font size $S \in \{8, 10, 12, 18, 24\}$, each $D \in \{30, 32, \ldots, 50\}$, $W \in \{1, 3, \ldots, 11\}$, $Q \in \{5, 6, 7\}$ and $K \in \{1, 2, 4, ..., 32\}$, a BHMM-based word recognizer was trained and tested, for each value of $GSF \in \{30, 40, 50\}$, as described above. Also as above, the best recognizer for each size was then tested with variable number of states ($F \in \{0.3,$

296 ..., 0.7}) and different repositioning techniques ($R = \{N, V, H, B\}$; where
297 N=None, V=Vertical, H=Horizontal and B=Both vertical and horizontal).
298 The results obtained were similar to those reported in Fig. 3 for font size
299 6. More precisely, the best error rates were obtained with windows of width
300 $W \in \{7, 9, 11\}$, $K = 32$ components, $GSF = \{40, 50\}$, variable number of
301 states with $F \in \{0.4, 0.5, 0.6\}$, and vertical repositioning. For brevity, these
302 error rates are not reported here in detail, as those in Fig. 3 for font size 6.
303 Instead, only a summary of best error rates is reported in Table 1 (including
304 font size 6 for completeness). Note that the best recognizer (combination of
305 parameter values) for each font size is trained within the parameter ranges
306 indicated above. Indeed, all recognizers trained within these ranges provide
307 nearly identical error rates.

Table 1: Best recognizer (combination of parameter values) and its CER(%) for each size.

Size	D	W	R	F	K	GSF	CER(%)
6	38	7	V	0.5	32	50	1.1
8	40	7	V	0.6	32	40	0.6
10	44	9	V	0.5	32	40	0.6
12	40	9	V	0.5	32	40	0.4
18	40	9	V	0.5	32	40	0.5
24	42	11	V	0.4	32	40	0.8

308 To get some insight into the behavior of our windowed BHMMs, a real
309 model for the character is (partially) shown in Figure 4 (bottom) together
310 with its Viterbi alignment with a real image of the character , extracted from
311 sample *Image_24_ArabicTransparent_562, set1* (top). Bernoulli prototypes

are represented as gray images where the gray level of each pixel measures the probability of its corresponding pixel to be black (white = 0 and black = 1). From these prototypes, it can be seen that the model works as expected, i.e. each state from right to left accounts for a different local part of ﻉ, as if the sliding window was moving smoothly from right to left.

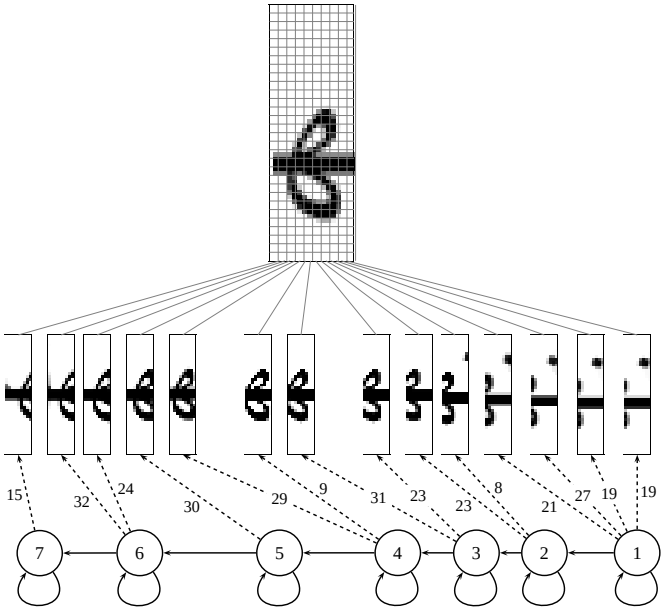

Figure 4: Real BHMM example for character ﻉ and its Viterbi alignment with a real image of the character ﻉ, extracted from sample *Image_24_ArabicTransparent_562* (top).

5.4. Results using the UPVPC2 protocol

The UPVPC1 protocol was used to study the effect on the CER of various key parameters, variable number of states, and repositioning. Taking into account the best results obtained with it, the UPVPC2 protocol was used in a new series of experiments to obtain results in conditions

similar to those used in the ICDAR 2011 Arabic Recognition Competition (see Sec. 5.2). In particular, for each of the five font types considered in UPVPC2, $T \in \{Trans, Anda, Diw, Simp, Trad\}$, and each font size $S \in \{6, 8, 10, 12, 18, 24\}$, a BHMM-based word recognizer was trained and tested from the data in UPVPC2 of font type T and size S. We used $D = 40$, $W = 9$, $R = V$, $F = 0.5$ (on a Viterbi segmentation produced by a recognizer trained with $Q = 7$, $K = 128$ and $GSF = 40$), $K = 128$ and $GSF = 40$. Except for the K, these parameter values are within the parameter ranges leading to the best error rates with the UPVPC1 protocol. However, in the case of K, we used 128 instead of 32. As discussed in Sec. 5.3, values of K larger than 32 had to be tried, especially with more training data as with the UPVPC2 protocol. Actually, we tried each $K \in \{1, 2, 4, ..., 128\}$, though $K = 128$ provided the best error rates in all cases.

Table 2 shows CER results for each font type and size. The error rates labeled as 2013a in the Year column were obtained as described above. That is, each test sample was accompanied by its font type and size so as to select its appropriate recognizer. However, the error rates labeled as 2013b were obtained in a slightly different way, by only providing the font size of each test sample. In this case, given a test sample of size S, all the five font-dependent recognizers for size S were run in parallel and that producing the highest classification score (see Eq. (5)) was chosen to decide the recognized word. The error rates labeled as 2011 are the best results of the ICDAR 2011 competition, which were also obtained by only providing the font size of each test sample.

A first conclusion that can be drawn from Table 2 is that the figures

Table 2: CER results for each font type and size (2013a="font type and size given"; 2013b="only font size given"; 2011="best results from the ICDAR 2011 competition").

Font/Size	Year	6	8	10	12	18	24	Mean
Andalus	2013a	0.9	0.2	0.1	0.1	0.0	0.0	0.2
	2013b	0.9	0.2	0.1	0.1	0.0	0.0	0.2
	2011	1.1	5.2	3.9	3.3	3.3	3.0	3.3
Arabic Transparent	2013a	0.6	0.1	0.1	0.0	0.0	0.1	0.2
	2013b	0.6	0.1	0.1	0.0	0.0	0.0	0.1
	2011	1.0	3.5	3.4	3.9	3.8	3.9	3.3
Simplified Arabic	2013a	0.5	0.1	0.1	0.0	0.0	0.0	0.1
	2013b	0.4	0.1	0.1	0.0	0.0	0.0	0.1
	2011	0.8	3.9	3.3	3.1	3.0	2.6	2.8
Traditional Arabic	2013a	6.4	1.3	0.5	0.3	0.2	0.2	1.5
	2013b	6.5	1.3	0.5	0.3	0.2	0.2	1.5
	2011	10.7	18.1	14.1	11.5	12.5	11.7	13.1
Diwani Letter	2013a	10.0	7.2	6.7	6.2	6.1	5.9	7.0
	2013b	10.0	7.2	6.7	6.2	6.1	5.9	7.0
	2011	9.1	24.2	16.6	10.9	5.1	7.4	12.2

labeled as 2013a and 2013b are virtually identical. Therefore, when font size is known but font type is not, the procedure described above to obtain the 2013b results seems absolutely reliable. Another important conclusion from Table 2 is that the results of this work outperform by a large extent those from the competition. Note that, on average, recognition of Andalus, Arabic Transparent and Simplified Arabic is nearly perfect in terms of CER. On the other hand, recognition of Traditional Arabic and Diwani Letter is fairly good and comparatively much better than that of the ICDAR 2011 competition.

Apart from the above multi-font and mono-size recognition results, the ICDAR 2011 competition also included mono-size results on only the Arabic Transparent font. For this particular font, results were published for both, competition participants (IPSAR and UPV) and organizers (DIVA-REGIM). Also, more recent results have been published by (Awaida and Khorsheed, 2012), and by (Dershowitz and Rosenberg, 2013). The most recent results come from the ICDAR 2013 second competition on APTI, which included three more participants than in its first edition: SID, THOCR and Siemens (Slimane et al., 2013). All these results are shown in Table 3 in terms of CER and WER. UPV-REC1, UPV-BHMM and UPV-2013 refer to our system at, respectively, ICDAR 2011, ICDAR 2013 and this work. Note that the results of UPV-BHMM and UPV-2013 are nearly identical and thus, as expected, the UPVPC2 protocol provides a good approximation to the experimental conditions of the ICDAR competitions on APTI. These results are much better than those of UPV-REC1 and only at a marginal distance from the best system at the ICDAR 2013 second competition on APTI. They

are also much better than those reported in (Khoury et al., 2013), where an initial, preliminary part of the experiments and results described here can also be found.

Table 3: CER and WER results for the Arabic Transparent font in each size.

System	Year		6	8	10	12	18	24	Mean
IPSAR	2011	WER	94.3	26.7	25.0	16.9	22.9	22.5	34.7
		CER	40.6	5.8	4.9	3.1	4.3	3.2	10.3
UPV-REC1	2011	WER	5.5	2.6	3.3	7.5	15.4	15.6	8.3
		CER	1.0	0.4	0.6	1.3	3.1	4.0	1.7
DIVA-REGIM	2011	WER	13.1	4.1	4.3	6.1	2.1	1.1	5.1
		CER	2.0	0.8	0.7	1.2	0.3	0.3	0.9
Awaida et al.	2012	CER	-	-	-	-	-	-	3.4
Dershowitz et al.	2013	WER	72.4	21.1	10.2	6.0	1.0	1.5	18.7
		CER	31.8	5.6	2.5	2.4	0.2	0.4	7.2
UPV-BHMM	2013	WER	2.8	0.3	0.2	0.1	0.1	0.1	0.6
		CER	0.5	0.1	0.1	0.0	0.0	0.0	0.1
SID	2013	WER	5.7	3.8	1.8	1.2	3.4	2.6	3.1
		CER	0.3	0.0	0.0	0.1	0.0	0.0	0.1
THOCR	2013	WER	10.5	4.2	5.2	7.5	5.4	5.0	6.3
		CER	1.7	0.5	0.8	0.9	0.9	0.8	0.9
Siemens	2013	WER	0.1	0.1	0.0	0.1	0.0	0.0	0.1
		CER	0.0	0.0	0.0	0.0	0.0	0.0	0.0
UPV-2013	2013	WER	3.0	0.4	0.3	0.2	0.2	0.2	0.7
		CER	0.6	0.1	0.1	0.0	0.0	0.1	0.2

5.5. New results using a DNN hybrid HMM system and vertical repositioning

Previous experiments have shown that the results obtained by using BHMMs are improved by applying the vertical repositioning technique. In recent work on handwritten recognition, vertical repositioning has also shown a significant improvement when used with other models than Bernoulli HMMs. In particular, in (Doetsch et al., 2012), a notable improvement was reported by using a Long Short Term Memory recurrent neural network (LSTM-RNN) tandem HMM and vertical repositioning on Arabic and French handwriting. This improvement is also observed in (Hamdani et al., 2014) where the window repositioning is used as a preprocessing step.

In order to asses that the vertical repositioning is useful for printed Arabic recognition with the current state-of-the-art techniques based on neural networks, such as LSTM-RNN, we have carried out a new series of experiments using the UPVPC2 protocol and a Deep Neural Network (DNN) hybrid HMM system (Dahl et al., 2012). This technique is similar to the Long Short Term Memory (LSTM) technique applied in (Rashid et al., 2013). It has been implemented in a recently released, open-source toolkit for automatic speech recognition called TLK toolkit (The transLectures-UPV Team, 2013). On the basis of our experience on the application of TLK to speech recognition tasks within the transLectures project, we decided to use it also for the additional experiments discussed in this Section. The results of these experiments, with and without vertical repositioning, are shown in Table 4.

As with the winner of ICDAR 2013 (Table 3), the results in Table 4 are nearly perfect. Even though the error is nearly zero, vertical repositioning still obtains slight improvements. In particular, for the more challenging

Table 4: CER and WER results for the Arabic Transparent font in each size.

System	Year		6	8	10	12	18	24	Mean
Vertical Rep.	2014	WER	0.16	0.13	0.12	0.12	0.13	0.15	0.14
		CER	0.03	0.03	0.02	0.02	0.03	0.03	0.03
Without Rep.	2014	WER	0.22	0.20	0.12	0.13	0.13	0.16	0.16
		CER	0.04	0.04	0.02	0.02	0.03	0.03	0.03

font sizes (6 and 8), a modest improvement is achieved when applying repositioning. Specifically, for font size 6 results were 0.16% with repositioning and 0.22% without repositioning. (Note that, as we were using 19000 test samples approximately for each font size, a difference of 0.06% accounts for about 11 classification errors.) In a similar way, for font size 8, results were 0.13 and 0.20 for repositioning and non-repositioning respectively.

6. Concluding remarks

Windowed Bernoulli HMMs with repositioning have been described and extensively tested for printed Arabic recognition on the Arabic Printed Text Image (APTI) database. A system based on these models, though with no repositioning, ranked first at the ICDAR 2011 Arabic recognition competition for printed Arabic text, also based on the APTI database. Following evaluation protocols similar to those of the competition, this system has been largely improved by the use of repositioning and an exhaustive experimentation to adjust various key parameters and model topology (variable number of states). Results comparatively much better than those of the competition have been reported on multi-font and mono-size recognition, with nearly perfect performance for most fonts in terms of Character Error Rate. Indeed, a

second edition of the competition on APTI was recently held at the ICDAR 2013 and our improved system obtained results nearly identical to those reported here. This second edition was harder than the first and our system ranked second, though only at a marginal distance from the best.

For future work, we would be interested in carrying out a deep analysis to compare repositioning with other, more complex, techniques for baseline detection and correction.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no 287755. Also supported by the Spanish Government (Plan E, iTrans2 TIN2009-14511 and AECID 2011/2012 grant).

References

Awaida, S., Khorsheed, M., 2012. Developing discrete density Hidden Markov Models for Arabic printed text recognition, in: Computational Intelligence and Cybernetics (CyberneticsCom), 2012 IEEE International Conference on, pp. 35–39.

Dahl, G.E., Member, S., Yu, D., Member, S., Deng, L., Acero, A., 2012. Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition, in: IEEE Transactions on Audio, Speech, and Language Processing.

Dehghan, M., Faez, K., Ahmadi, M., Shridhar, M., 2001. Handwritten Farsi (Arabic) word recognition: a holistic approach using discrete HMM. Pattern Recognition 34, 1057–1065.

Dershowitz, N., Rosenberg, A., 2013. Language, Culture, Computation: Studies in Honor of Yaacov Choueka. Springer-Verlag. volume 8000 of *Lecture Notes in Computer Science*. chapter Arabic Character Recognition.

Doetsch, P., Hamdani, M., Ney, H., Gimenez, A., Andres-Ferrer, J., Juan, A., 2012. Comparison of Bernoulli and Gaussian HMMs using a vertical repositioning technique for off-line handwriting recognition, in: International Conference on Frontiers in Handwriting Recognition, Bari, Italy. pp. 3–7.

Dreuw, P., Heigold, G., Ney, H., 2009. Confidence-Based Discriminative Training for Model Adaptation in Offline Arabic Handwriting Recognition, in: ICDAR '09, Barcelona (Spain). pp. 596–600.

Giménez, A., Andrés-Ferrer, J., Juan, A., 2014a. Discriminative Bernoulli HMMs for isolated handwritten word recognition. Pattern Recognition Letters 35, 157 – 168. Frontiers in Handwriting Processing.

Giménez, A., Juan, A., 2009. Embedded Bernoulli Mixture HMMs for Handwritten Word Recognition, in: ICDAR '09, Barcelona (Spain). pp. 896–900.

Giménez, A., Khoury, I., Andrés-Ferrer, J., Juan, A., 2014b. Handwriting word recognition using windowed Bernoulli HMMs. Pattern Recognition Letters 35, 149 – 156. Frontiers in Handwriting Processing.

Giménez, A., Khoury, I., Juan, A., 2010. Windowed Bernoulli Mixture HMMs for Arabic Handwritten Word Recognition, in: ICFHR' 10, Kolkata (India). pp. 533–538.

Grosicki, E., El Abed, H., 2009. ICDAR 2009 Handwriting Recognition Competition, in: ICDAR '09, Barcelona (Spain). pp. 1398 – 1402.

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

Hamdani, M., Doetsch, P., Kozielski, M., El-Desoky Mousa, A., Ney, H., 2014. The RWTH Large Vocabulary Arabic Handwriting Recognition System, in: International Workshop on Document Analysis Systems, France.

Jelinek, F., 1997. Statistical Methods for Speech Recognition. MIT Press.

Khoury, I., Gimnez, A., Juan, A., Andrs-Ferrer, J., 2013. Arabic Printed Word Recognition Using Windowed Bernoulli HMMs, in: Petrosino, A. (Ed.), Proc. of the 17th Int. Conf. on Image Analysis and Processing ICIAP 2013. Springer Berlin Heidelberg. volume 8156, pp. 330–339.

Märgner, V., El Abed, H., 2007. ICDAR 2007 - Arabic Handwriting Recognition Competition, in: ICDAR '07, Curitiba (Brazil). pp. 1274–1278.

Märgner, V., El Abed, H., 2009. ICDAR 2009 Arabic Handwriting Recognition Competition, in: ICDAR '09, Barcelona (Spain). pp. 1383–1387.

Märgner, V., El Abed, H., 2010. ICFHR 2010 - Arabic Handwriting Recognition Competition, in: ICFHR '10, Kolkata (India). pp. 709–714.

Otsu, N., 1979. A Threshold Selection Method from Gray-Level Histograms. IEEE Trans. on Systems, Man and Cybernetics 9, 62–66.

Rabiner, L., Juang, B.H., 1993. Fundamentals of speech recognition. Prentice-Hall.

Rashid, S.F., Schambach, M.P., Rottland, J., von der Nüll, S., 2013. Low Resolution Arabic Recognition with Multidimensional Recurrent Neural Networks, in: Proceedings of the 4th International Workshop on Multilingual OCR, ACM, New York, NY, USA. pp. 6:1–6:5.

Slimane, F., Ingold, R., Kanoun, S., Alimi, A.M., Hennebert, J., 2009. A New Arabic Printed Text Image Database and Evaluation Protocols, IEEE. pp. 946–950.

Slimane, F., Kanoun, S., Abed, H.E., Alimi, A.M., Ingold, R., Hennebert, J., 2011. ICDAR 2011 - Arabic Recognition Competition: Multi-font Multi-size Digitally Represented Text, IEEE. pp. 1449–1453.

Slimane, F., Kanoun, S., El Abed, H., Alimi, A.M., Ingold, R., Hennebert, J., 2013. ICDAR 2013 Competition on Multi-font and Multi-size Digitally Represented Arabic Text, CPS. pp. 1465–1469.

The transLectures-UPV Team, 2013. The translectures-upv toolkit (tlk). http://translectures.eu/tlk. http://www.translectures.eu/tlk/citing-tlk/.

Young, S., et al., 1995. The HTK Book. Cambridge University Engineering Department.