Variable length and context-dependent HMM letter form models for Arabic handwritten word recognition

Anne-Laure Bianne-Bernard[a b], Fares Menasri[a], Laurence Likforman-Sulem[b], Chafic Mokbel[c], Christopher Kermorvant[a]

[a]A2iA SA, 40 bis rue Fabert, Paris, France
[b]Telecom ParisTech, dept. TSI, CNRS-UMR 5141, Paris, France
[c]University of Balamand, Lebanon

ABSTRACT

We present in this paper an HMM-based recognizer for the recognition of unconstrained Arabic handwritten words. The recognizer is a context-dependent HMM which considers variable topology and contextual information for a better modeling of writing units. We propose an algorithm to adapt the topology of each HMM to the character to be modeled. For modeling the contextual units, a state-tying process based on decision tree clustering is introduced which significantly reduces the number of parameters. Decision trees are built according to a set of expert-based questions on how characters are written. Questions are divided into global questions yielding larger clusters and precise questions yielding smaller ones. We apply this modeling to the recognition of Arabic handwritten words. Experiments conducted on the OpenHaRT2010 database show that variable length topology and contextual information significantly improves the recognition rate.

Keywords: Arabic handwriting recognition ; HMM-based system ; state-based clustering.

1. INTRODUCTION

The recognition of Arabic writing has many applications such as mail sorting, bank checks reading and the recognition of modern and historical handwritten documents. Arabic writing is very challenging for off-line recognition systems.[1] The handwriting is highly cursive which makes it difficult to deslant. It includes various small-size marks which modify the meaning of letters (the diacritics). Last, when dealing with texts, dictionary sizes may be very large due to the formation of Arabic words with prefixes and suffixes from word roots.

Different approaches have been proposed for recognizing isolated words and printed text lines.[2,3] For word recognition, the analytical strategy is very popular : a word model is built from the concatenation of character models. Moreover segmenting words into characters is avoided and characters models are built from word images and their transcription. The analytical strategy is convenient for enlarging a vocabulary with new words since new vocabulary words can be described through their compound letters, without providing their images.

HMMs are effective for modeling unconstrained words since they can cope with non-linear distortions. The analytical strategy can be implemented in HMM systems through the so-called sliding window approach.[4,5] Such systems can be easily applied for both Latin and Arabic and achieve state-of-the-art performance.[6,7]

In the following, an HMM-based classifier is used for the recognition of handwritten Arabic words. This system takes into account the context of a character within a writing unit. For a given character, we have considered the influence of neighboring characters on its shape, as shown on Figure 1. We have used our knowledge of the shapes of neighboring characters to assist the modeling of individual Arabic letters. Such knowledge has been embedded in knowledge-based rules and decision trees used by our state clustering process and this results in more accurate character models.

For this purpose, different character models can be built according to different contexts. This approach is known as the context-dependent approach in the domain of speech recognition and has been applied to printed character recognition. To our knowledge, only a few works deal with contextual modeling in handwriting recognition.[8–10]

Contextual approaches lead to an excessive growth in the number of models, since one model is needed for each pair of adjacent characters. It is thus desirable to reduce the number of models and model parameters while

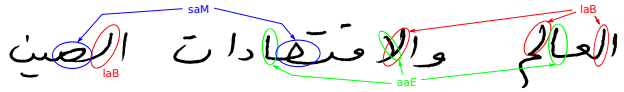

FIGURE 1. Illustration of the influence of context for handwriting. The three words العالم والاقتصادات الصين have been written by the same writer. However, characters laB , aaE and saM yield different shapes.

preserving model refinement. Hence, model sharing and parameter tying are necessary to reduce the number of parameters. Schussler and Niemann[8] describe a context-dependent system using HMMs, where all sub-word units (from monographs to the whole word) are modeled within a word hierarchy. Models with not enough training samples are eliminated. The state-based tying proposed by Natarajan *et al.*[3] uses a mixture of 128 Gaussians associated to each state position of contextual models (trigraphs) corresponding to the same base character. The total number of models can also be reduced by clustering all trigraphs according to contexts described not as characters but as ascending or descending strokes.[10] Fink *et al.*[9] also proposed for Latin handwriting a system based on contexts, these contexts being defined as broad categories. A data-driven clustering of the 1500 Gaussian densities of the mixture is performed at each state position and for each category. It is worthy to note that clustering and tying the contextual models may offer, in addition to reducing the number of parameters, the possibility to automatically capture common contextual effects.

The context-dependent system described below has some common characteristics with the context-based systems described in Ref. 3 and Ref. 9. In both, trigraphs are modeled and parameters are shared at each state position. Our system contrasts with these previous approaches as our state clustering is knowledge-driven : we cluster models using decision trees where questions at each node are expert-based and specific to the way handwritten characters are drawn. In addition, we include context from the neighboring windows by adding dynamic features (derivative) in each feature vector. We have applied such approach to Latin handwriting.[11] In the present paper we apply it to Arabic handwriting and we present specific rules for this script.

The paper is organized as follows : Section 2 details our sliding window system, the method to train HMM with variable topology and context-dependent models, Section 3 reports our experiments on the OpenHaRT2010 database and the conclusion and future extensions are given in Section 4.

2. VARIABLE TOPOLOGY AND CONTEXT DEPENDENT HMM MODELS

In this section, we describe our baseline model and our two contributions : variable HMM topology training and context-dependent character form modeling.

2.1 Baseline HMM model

Our baseline model is based on a sliding window feature extraction and an HMM letter form modeling. A sequence of feature vectors is extracted from right to left through overlapping windows applied on binary deslanted word images. Within each window a set of 34 features is extracted :

– 26 features are inspired of the geometric features proposed by El-Hajj *et al.*4 : 13 features related to pixel densities, 12 features related to pixel configurations (in order to capture stroke concavities), and a derivative feature. Some of these 26 features are baseline dependent.

– 8 features are local gradient histogram features, derived from the work of Rodriguez and Perronnin.[12] For each window, the histogram of orientations is computed using all its pixels.

For our experiments, the following parameters are used : window width of 9 pixels, overlap between two sliding windows equal to 3 pixels. The number of cells per window for the geometric features is 20. These parameters happened to be the best for our images.

We introduce the context at the feature extraction level through derivative features. The derivation is computed with a regression. In the speech recognition domain, the first order regression is known as delta coefficients. K is the chosen depth of the regression, giving the number of surrounding feature vectors (2 . K) used for computing the dynamic features. The final feature vector is thus the concatenation of the original feature vector and its first order regression vector.

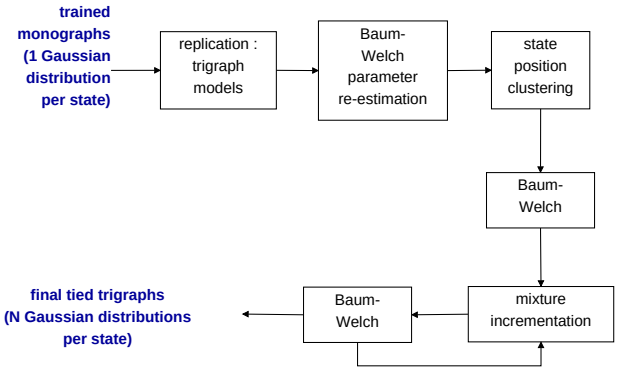

FIGURE 2. Training process uses state tying and Gaussian mixture incrementation.

A word is modeled by the concatenation of its compound character models. For the baseline model, all character models share the same HMM Bakis topology : 12 emitting states, one self transition, and left-right transitions allowed to the next two states. We consider HMMs with continuous observation densities so that the observation probability density for each state is a mixture of N_G Gaussian distributions.

2.2 Variable length monograph training

In Arabic, as in most of alphabetic languages, not all the letters have the same length. Therefore, the adaptation of the HMM topology to the character length has been shown to be important for handwriting recognition. Several methods have been proposed, either based on the length of the extracted feature vector[13] or on an iterative selective process based on the likelihood of the different topologies.[14] We propose here a method of the latter kind, but less computationally intensive.

First, all the letter HMM are initialized with the same topology (Bakis topology, 12 states per model, 1 Gaussian distribution) and several iteration of the Baum-Welch algorithm are used to estimate the parameters. During the last iteration, the state occupation statistics are computed :

$$\Gamma(s) = \sum_{t=1}^{T} \gamma_s(\boldsymbol{o}_t). \tag{1}$$

where $\gamma_s(\boldsymbol{o_t})$ is the posterior of the observation $\boldsymbol{o_t}$ in state s. The number of states of the character C is then :

$$\lfloor L_s(C) \rfloor = \frac{\sum_{s \in S_C} \Gamma(s)}{|C|} \tag{2}$$

where S_C is the set of states of the HMM corresponding to the character C and $|C|$ the frequency of the character C in the training set. One can consider that $\sum_{s \in S_C} \Gamma(s)$ computes the total number of times an observation corresponds to any state of the character C (in terms of probability). Hence, dividing this number by $|C|$ gives an estimation of the number of states the HMM of C should have. When $\lfloor L_s(C) \rfloor$ is computed for all the characters, the occupation statistics are re-estimated with the new topologies. For each character HMM, the process is iterated until the variation of its topology (in number of states) is below a predefined threshold. When no more HMM topology is to be modified, the process stops.

2.3 Context-dependent trigraph models training

Once the HMM topologies have been chosen, each monograph is replicated to create its different context-dependent variants. Since the number of parameters of the context-dependent models is very large, the models must share parameters in order to be well estimated. Shared parameters are obtained with a state-based tying algorithm described in the next section.

State tying determines which states can share the same Gaussian distributions. The state position-based principle is that for a given central letter, all states corresponding to the same position in an HMM model are

FIGURE 3. Illustration of state clustering for the trigraphs centered on character *seB* (ـسـ).

subject to agglomerative clustering. Our approach consists of building expert-based rules and decision trees to perform this state clustering.

The merging or splitting of state clusters is driven by a binary tree whose nodes correspond to questions on the characteristics of the models. Such decision trees have been designed for speech recognition at the phone level by experts.[15] To our knowledge, no work using such trees for handwriting recognition exists.

In our case, decision trees are based on a set of questions on the behavior of left and right contexts, and are applied to states. Based on the same initial set of questions, one tree is built for every state position of all trigraphs with the same central letter. Starting at the root node, all the states corresponding to the same position and the same central letter are gathered in a single cluster. Then, the binary question which maximizes the likelihood of the two children clusters it would create is chosen, and the split is made, creating two new nodes. This splitting continues until the increase in likelihood falls below a threshold or no questions are available to create nodes with a sufficient state occupancy count.

Let us consider a node containing the set of states $\boldsymbol{S}$ to be split in a given tree. The set $\boldsymbol{S}$ corresponds to the set of training frames $\{o_f\}_{f \in \boldsymbol{F}}$. As all states in $\boldsymbol{S}$ are tied in the node, they all share the same mean $\mu(\boldsymbol{S})$ and covariance matrix $\Sigma(\boldsymbol{S})$. The likelihood of $\boldsymbol{S}$ generating the set of frames is hence given by :

$$L(\boldsymbol{S}) = \sum_{f \in \boldsymbol{F}} \sum_{s \in \boldsymbol{S}} log(Pr(\boldsymbol{o}_f; \mu(\boldsymbol{S}), \Sigma(\boldsymbol{S})))\gamma_s(\boldsymbol{o}_f) \tag{3}$$

where $\gamma_s(\boldsymbol{o}_f)$ is the a posteriori probability of frame $\boldsymbol{o}_f$ being generated by state $\boldsymbol{s}$. Based on the work of Young[16] and assuming that we work with Gaussian probability density functions, $L(\boldsymbol{S})$ can be rewritten :

$$L(\boldsymbol{S}) = -\frac{1}{2}(log[(2\pi)^n|\Sigma(\boldsymbol{S})|] + n)\Gamma(\boldsymbol{S}) \tag{4}$$

$\Gamma(\boldsymbol{S})$ is the accumulated state occupancy of the node, $\Gamma(\boldsymbol{S}) = \sum_{f \in \boldsymbol{F}} \sum_{s \in \boldsymbol{S}} \gamma_s(\boldsymbol{o}_f)$, and n is the dimension of the feature vectors.

We introduce then ΔL_q :

$$\Delta L_q = L(\boldsymbol{S}_{q+}) + L(\boldsymbol{S}_{q-}) - L(\boldsymbol{S}) \tag{5}$$

The split of the state set into two subsets $\boldsymbol{S}_{q+}$ (answer to q is *yes*) and $\boldsymbol{S}_{q-}$ (answer to q is *no*) is made by question q^* which maximizes ΔL_q, provided that $\Gamma(\boldsymbol{S}_{q+})$ and $\Gamma(\boldsymbol{S}_{q-})$ are over the minimal state occupancy threshold, and that ΔL_q is above the threshold of minimal increase in likelihood. This condition can be reformulated[17] :

$$\begin{aligned} q^* \quad = \quad & argmin_q\{\frac{1}{2}\ [\Gamma(\boldsymbol{S}_{q+})log(|\Sigma(\boldsymbol{S}_{q+})|) + \\ & \Gamma(\boldsymbol{S}_{q-})log(|\Sigma(\boldsymbol{S}_{q-})|) - \Gamma(\boldsymbol{S})log(|\Sigma(\boldsymbol{S})|)]\} \end{aligned} \tag{6}$$

The parameters ensuring efficient sizes of state clusters, namely the minimal state occupancy threshold and the minimal increase in likelihood are tuned on the validation database. Trees reduced to their only root can be observed. They correspond to monographs with few examples which aim to tie all their corresponding trigraphs into a single model.

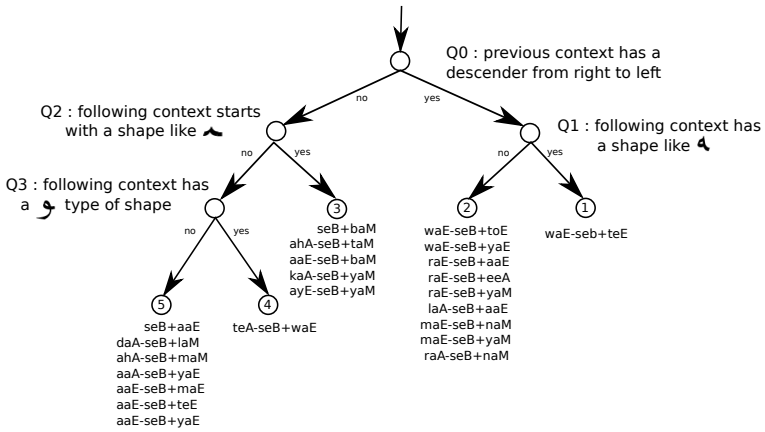

FIGURE 4. Example of a decision tree for state clustering : questions and clusters are shown for the 2nd state of all *-seB+* (سـ) trigraphs.

Decision trees have the ability of modeling unseen trigraphs with existing ones. This property is useful when test and training dictionaries differ. Each state of a new trigraph is positioned at the root node of the tree corresponding to the same state position and the same central letter. Then each state follows a designated path along its belonging tree, defined by answering questions on the trigraph contexts, until it reaches a node where a cluster is positioned. The state model representing the cluster is the model assigned to the considered state number of the new trigraph.

After state-based tying, the number of Gaussian distributions associated to one state is incremented step by step by splitting the Gaussians of the mixture with highest weight (see Ref. 11).

2.4 Knowledge-based Rules for Arabic trigraphs

As we already stated in previous work[18],[11] it is quite obvious that the way of writing a character within a word is affected by adjacent letters, therefore the justification of using context-models to improve the accuracy of modeling.

We use HTK syntax[19] to designate trigraphs, and IFN-ENIT transliteration[20] to make it writeable with ASCII characters. For example, in Arabic word فيل , the letter يـ is surrounded by letters فـ (previous-context) and لـ (following-context).Using HTK notation, previous-context is defined by '-' and following context is defined by '+', which gives the trigraph : faB-yaM+laE. The construction of our question sets is driven by the two following hypothesis :

– letters with similar ending strokes will have a tendency to affect the following central letter in a similar manner

– letters with similar beginning strokes will have a tendency to affect the previous central letter in a similar manner

According to those hypothesis, Arabic letters which share the same shapes[21] should be good candidates to build question sets (QS). Those question sets are used in our clustering decision trees. Similar endings will lead to groupings in previous-context question sets (P_QS), whereas similar beginnings will lead to groupings in following-context question-sets (F_QS).

For example, the beginning (right part) of letters { ـقـ ـفـ ق } is very similar. They will be regrouped in the same F_QS : { *+faM , *+kaM , *+faE , *+kaE }. Identically, { ـر ر } are grouped in the set { *+raE , *+zaE

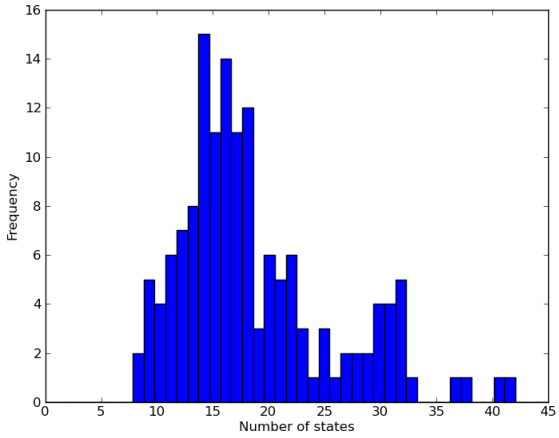

FIGURE 5. Histogram of the number of states per HMM model for the 150 models trained with variable length monograph training algorithm.

} In { ث ت ب ثـ تـ بـ يـ نـ شـ ش سـ س }, all letters do not share the same shapes, but again their beginning (right part) looks similar. This should lead to the creation of the following F_QS : { *+seM, *+shM, *+seE, *+shE, *+naM, *+baM, *+taM, *+thM, *+baE, *+taE, *+thE} The same is true for ending (left part) of letters, used to build P_QS :

{ غ ع } → { ayM-* , ghM-* }

{ ز ر و } → { waE-* , waA-* , raE-* , raA-* , zaE-* , zaA-* }

See Ref.22 for the full list of question sets.

3. EXPERIMENTS

We have tested the different models on the OpenHaRT2010 database.[23] This database is composed of a 40,000 handwritten documents in Arabic, annotated at word level (position and value). It corresponds to a very large number of word images (about 4 million). This recognition task is much harder than the other tasks proposed on Arabic, for example on the IFN-ENIT database. This difficulty is due to several factors : writing conditions (presence of lines), variety of writers, very large vocabulary, etc.

We have trained our models on the predefined set called *phase_1_train_set*, containing 739,849 isolated word images. We have trained HMM models for 117 Arabic letter forms, 10 digits and 23 punctuation symbols. We used the set called *Phase_1_dev_set* as validation set (81,948 images). The vocabulary for decoding has been selected on the training set and consisted of the 20,000 most frequent words. All the images have been rescaled to 300 dpi.

We first study the result of our training procedure for variable topology HMM. Figure 5 shows the histogram of the number of states for the 170 HMM models. If most of the models have a state number between 10 and 20, some of them have around 30 states and the maximum is reached with 42 states. The range of different state numbers per HMM is larger than what we have observed for other languages such as French or English. This could be due to a larger variability of Arabic handwriting.

We report now the recognition results of the different models, the baseline HMM, the variable topology HMM and the context-dependent HMM with variable topology. Table 1 shows the isolated word recognition results on the OpenHart2010 *Phase_1_dev_set*. We report the recognition rates on the complete set *Phase_1_dev_set* (*with out-of-vocabulary (OOV) words*) and also only on the words of the set which are present in the vocabulary (*without OOV*). The latter results show the impact of the vocabulary coverage on the recognition rate. A study of the impact of vocabulary size on the recognition rate is interesting but out of the scope of this paper.

TABLE 1. Isolated word recognition results on the OpenHart2010 *Phase_1_dev_set* for the baseline model (fixed size) and the two proposed improvements : variable size models and context-dependent models. Rates are given with or without out-of-vocabulary (OOV) words.

Model	**Recognition rate without OOV**	**Recognition rate with OOV**
Fixed size model, context independent	47.7%	39.3 %
Variable size, context independent	52.3%	43.0 %
Variable size, context dependent	56.1%	46.2%

TABLE 2. Whole page recognition results on the OpenHart2010 *Evaluation set* for the proposed model and for two other systems : a context-independent HMM and a context-dependent HMM, both with fixed topology. In all cases, language models were used. Rates are directly comparable with the results of the OpenHaRT2010 evaluation.

Model	**Recognition rate**
Fixed size HMM, context independent	45.0%
Fixed size HMM, context dependent	54.0%
Variable size HMM, context dependent	58.0%

Table 1 demonstrates that the two modeling methods proposed in this paper significantly improve the recognition results. First, the method to adjust the topology of the HMM yields a 10% increase in recognition rate, from 39.3% to 43.0%. Second, the context-dependent modeling yields a supplementary increase of 7.5%, from 43.0% to 46.2%. The same improvements are observed when the evaluation is done only on the words in the vocabulary (no OOV).

Finally, we compare the proposed system to other recognition systems submitted to the OpenHaRT2010 evaluation.[23] For this evaluation, we have used a language model. For each line of text (given by the annotation), a lattice of recognition results was built using, for each word position, the 50 most likely recognition results given by the recognizer. The language model was used to rescore the lattice and the best path was computed in the rescored lattice. The sequence of words along the best path gave for each word position the final recognition result of the systems. The language model was a 3-gram using modified Kneyser-Ney smoothing, trained on 370 million words of the Arabic Gigaword v2 corpus[24] using the SRILM toolkit[25] after a simple tokenization. The recognition rates on the OpenHaRT2010 evaluation set are given on Table 2 for the proposed model and for two of our previous models submitted to the evaluation in 2010, a context-independent HMM and a context-dependent HMM, both with fixed topology. The proposed model outperforms our previous models and yields a 7% increase in recognition rate compared to our best previous model. For this evaluation, the best result was obtained by our model based on a combination of three recognizers, with 62.3% of recognition rate.

4. CONCLUSION

We have described a system for Arabic isolated word recognition based on HMM with variable topology and context-dependent letter form modeling. The topology of each HMM is automatically adapted to each letter form to be modeled. For the context-dependent models, parameter sharing is obtained thanks to a state tying algorithm using decision trees defined with morphological questions. The improvements obtained with these two techniques are measured on the OpenHaRT2010 database. The two proposed techniques yield a significant improvement over our baseline model.

However, there are numerous ways to improve our system. First, we need to develop a better modeling of the Arabic characters, for example when a vertical ligature is used. This can be done by defining writing variants, the same way it is done in speech recognition for pronunciation variants. Second, we need to increase the size of the vocabulary in order to reduce the out-of-vocabulary rate. Expanding the vocabulary will also increase the error rate, but this effect can be counterbalanced by the use of language models. Finally, this system can be combined with other recognition systems, as shown successfully for the OpenHaRT 2010 competition.

REFERENCES

[1] Amin, A., "Off-line arabic character recognition : the state of the art," *Pattern Recognition* **31**(5), 517–530 (1998).

[2] Lorigo, L. M. and Govindaraju, V., "Offline arabic handwriting recognition : A survey," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(5), 712–724 (2006).
[3] Natarajan, P., Lu, Z., Schwartz, R. M., Bazzi, I., and Makhoul, J., "Multilingual machine printed ocr," *Int. Journ. of Pattern Recognition and Artificial Intelligence* **15**(1), 43–63 (2001).
[4] Al-Hajj, R., Likforman-Sulem, L., and Mokbel, C., "Combining slanted-frame classifiers for improved HMM-based arabic handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(7), 1165–1177 (2009).
[5] Pechwitz, M. and Märgner, V., "HMM-based approach for handwritten arabic word recognition using the IFN/ENIT database," in [*Proc. of the Int. Conf. on Document Analysis and Recognition*], 890–894 (2003).
[6] Märgner, V. and Abed, H. E., "ICFHR 2010 - arabic handwriting recognition competition," in [*Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*], 709–714 (2010).
[7] Plotz, T. and Fink, G., "Markov models for offline handwriting recognition : a survey," *Int. Journ. on Document Analysis and Recognition* **12**, 269–298 (2009).
[8] Schussler, M. and Niemann, H., "A HMM-based system for recognition of handwritten address words," in [*Proc. of the Inty. Workshop on Frontiers in Handwriting Recognition*], 505–514 (1998).
[9] Fink, G. and Plotz, T., "On the use of context-dependent modeling units for HMM-based offline handwriting recognition," in [*Proc. of the Int. Conf. on Document Analysis and Recognition*], **2**, 729–733 (2007).
[10] El-Hajj, R., Mokbel, C., and Likforman-Sulem, L., "Recognition of arabic handwritten words using contextual character models," in [*Proc. of Document Recognition and Retrieval*], (2008).
[11] Bianne, A.-L., Menasri, F., Mohamad, R. A.-H., Mokbel, C., Kermorvant, C., and Likforman-Sulem, L., "Dynamic and contextual information in HMM modeling for handwritten word recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(10), 2066–2080 (2011).
[12] Rodriguez, J. A. and Perronnin, F., "Local gradient histogram features for word spotting in unconstrained handwritten documents," in [*Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*], (2008).
[13] Zimmermann, M. and Bunke, H., "Hidden markov model length optimization for handwriting recognition systems," in [*Proc of the Int. Workshop on Frontiers in Handwriting Recognition*], 369–375 (2002).
[14] Schambach, M.-P., "Model length adaptation of an HMM based cursive word recognition system," in [*Proc. of the Int. Conf. on Document Analysis and Recognition*], **1**, 109–113 (2003).
[15] Chelba, C. and Morton, R., "Mutual information phone clustering for decision tree induction," in [*Proc. of the Int. Conf. on Spoken Language Processing*], (2002).
[16] Young, S. J., Odell, J. J., and Woodland, P. C., "Tree-based state tying for high accuracy acoustic modelling," in [*Proc. of the Workshop on Human Language Technology*], 307–312 (1994).
[17] Zen, H., Tokuda, K., and Kitamura, T., "Decision tree based simultaneous clustering of phonetic contexts, dimensions, and state positions for acoustic modeling," in [*Proc. of the Eur. Conf. on Speech Communication and Technology*], 3189–3192 (2003).
[18] Bianne, A.-L., Kermorvant, C., and Likforman-Sulem, L., "Context-dependent hmm modeling using tree-based clustering for the recognition of handwritten words," in [*Proc. of Document Recognition and Retrieval*], **7534** (2010).
[19] Young, S. and al., [*The HTK Book V3.4*], Cambridge University Press (2006).
[20] Pechwitz, M., Maddouri, S. S., Maergner, V., Ellouze, N., and Amiri, H., "IFN/ENIT database of handwritten arabic words," in [*Colloque International Francophone sur l'Ecrit et le Document*], (2002).
[21] Menasri, F., Vincent, N., Cheriet, M., and Augustin, E., "Shape-based alphabet for off-line arabic handwriting recognition," in [*Proc. of the Int. Conf. on Document Analysis and Recognition*], 969–973 (2007).
[22] http ://perso.telecom-paristech.fr/ lauli/ArabicScriptRules/.
[23] http ://www.nist.gov/itl/iad/mig/hart2010.cfm.
[24] Graff, D., "Arabic gigaword second edition," (2006).
[25] Stolcke, A., "SRILM – an extensible language modeling toolkit," in [*Proc. Intl. Conf. on Spoken Language Processing*], **2**, 901–904 (2002).