

# The A2iA-Telecom ParisTech-UOB System for the ICDAR 2009 Handwriting Recognition Competition.

C. Kermorvant<sup>1</sup>, F. Menasri<sup>1</sup>, A-L. Bianne<sup>1,4</sup>, R. Al-Hajj<sup>2</sup>, C. Mokbel<sup>3</sup>, L. Likforman-Sulem<sup>4</sup>

<sup>1</sup> A2iA SA  
Paris, France  
{ck,fm,alb}@a2ia.com

<sup>2</sup> Lebanese International University  
Beirut - Lebanon  
rami.alhajj@gmail.com

<sup>3</sup>University of Balamand,  
Tripoli, Lebanon  
chafic.mokbel@balamand.edu.lb

<sup>4</sup> Telecom ParisTech / TSI  
Paris, France  
likforman@telecom-paristech.fr

## Abstract

*This article describes the isolated word recognizer presented by the authors to the ICDAR 2009 French handwriting recognition competition. This system is a combination of three isolated word recognizers based on different features and models. A novel n-best combination method is proposed and compared to standard combination methods. New results on the ICDAR 2009 test database are reported.*

## 1. Introduction

In the domain of handwriting recognition, several different technologies are today competing for the state-of-the-art performances : hidden Markov models (HMM) based on Gaussian mixtures models, hybrid neural network-HMM, recurrent neural networks, recognizers based on grapheme segmentation or based on a sliding windows. Regarding the features used by the different models, no consensus has emerged on the best features, contrary to what has happened in speech recognition with the mel-scaled cepstral coefficients. The different technologies lead to recognizers with different behaviour, so that the combination of these recognizers allows to dramatically improve the recognition performances [4]

The ICDAR 2009 competition for French isolated

handwritten word recognition [6] allowed to compare the recognition systems from eight different research laboratories and based on a large variety of technologies. Among these systems, four were based on a single recognizer (TUM, Irisa, Itesoft, LSIS) and four were a combination of several recognizer : Telecom ParisTech - A2iA (3 recognizers), Litis ( 2 recognizers), UPV (4 recognizers), Siemens (10 recognizers). Regarding the recognition results, all the systems based on combinations performed better than the systems based on a single recognizer, with the exception of the TUM system which was a single recognizer and outperformed all the other systems.

The problem of combining classifiers or recognizers has been extensively studied and many combination schemes are possible in sequential, in parallel, in hierarchical order, with and without training [7]. In this work, we only consider combinations in parallel since we combine lists of candidates provided by classifiers considered as independent.

In [5], two forms of HMM and a recurrent neural network were combined using different combination methods and a novel method (exponential Borda count) was proposed. This combination method allowed a reduction of almost 20% of the error rate on the IAM data base. In [3], Al-Hajj et al. proposed a neural network combination method in order to combine the results of three HMM word classifiers with sliding windows of different angles. This combination scheme pro-

vided better results than standard state-of-the-art methods (like Sum-Rule, Plural Vote and Borda Count).

This article describes the recognition system submitted to the ICDAR 2009 competition for French isolated handwritten word recognition [6] by A2iA, Telecom ParisTech and the University of Balamand. This system is a combination of three recognizers based on two different and complementary techniques : Gaussian HMM based on sliding window and hybrid neural network HMM based on grapheme segmentation. We also describe the improved neural network combination method that we used for the competition.

We present recognition results on the same test database as the ICDAR 2009 competition so that a direct comparison with the other systems in competition is possible. We report recognition results with accentuated characters that complement the results presented for the competition.

In section 2, we describe the three different handwritten word recognizers that we combined. In section 3, we recall several standard combination methods and describe our method based on neural networks. In section 4, we compare 5 combination methods of our 3 recognizers on the ICDAR 2009 database. We recall the results of the other systems in competition and report supplementary results on accentuated character recognition.

## 2 Description of the word recognizers

In our experiments, we have combined three different handwritten word recognizers based on HMM : one recognizer based on grapheme segmentation and a Multi-layer perceptron classifier (MLP-HMM) and two recognizers based on sliding windows and Gaussian mixture classifier (GMM-HMM). The characteristics of the different recognizers are summarized in Table 1.

### 2.1 Feature Extraction

We have trained recognizers on two kinds of features, based on either grapheme extraction or sliding windows.

#### 2.1.1 Grapheme-based feature extraction

For this kind of feature extraction, each word is segmented into graphemes, which is either a character or a subpart of a character. The grapheme segmentation process is based on the detection of singular points (bottom part of concavities, extremities of horizontal segments) on the contour of the connected components.

For each detected grapheme, a set of 74 features is computed based on grapheme characteristics, pixel densities and profiles [8]. In order to cope with the variability of grapheme extraction, grapheme classes are defined using a K-Means algorithm on all the features extracted from the training database. The number of classes is a parameter to be found on an independent validation set.

#### 2.1.2 Sliding windows feature extraction

For this kind of feature extraction, a sequence of feature vectors is extracted from left to right through overlapping windows. In our experiments, the window width was set to 8 pixels, the overlap between two sliding windows to 4 pixels, and the number of cells per window to 20. Within each window a set of 28 geometric features is extracted based on pixel densities, background/foreground transitions, concavities patterns [3]. The feature vector is extended with its first order derivative.

## 2.2 HMM word modeling

A word HMM is defined by the concatenation of its compound character HMM models. All the character models share the same topology: number of emitting states, with left-right transitions and skips allowed. In the case of grapheme segmentation, each state of the character HMM model corresponds to a grapheme. We have defined 78 different case and accent sensitive character models, which can be letters (accentuated or not), numerals, symbols (hyphen, apostrophe, etc.) or silence.

## 2.3 Observation modeling

### 2.3.1 Gaussian mixture

For the GMM-based recognizers, the observation probability density for each state is modeled with a mixture of 20 Gaussian distributions (GMM). We have tried two training procedures for GMM :

**incremental** : start with a flat initialization on a single Gaussian HMM and then increase progressively the number of Gaussians in the mixtures until the likelihood's variation becomes small.

**direct** : start directly with a 20 Gaussian mixture and train the model with fifteen EM iterations. The EM algorithm used in this case is the iterative segmental-EM or the Viterbi algorithm (a specific EM case using  $L_\infty$  norm) [9].

### 2.3.2 Multi-layer perceptron

In the MLP-HMM recognizer, a neural network is trained to evaluate the posterior probability of each grapheme class with respect to the feature vector. The neural network is a multi-layer perceptron with as many input neurons as features (74), one hidden layer and as many output neurons as grapheme classes. The neural network was trained in a supervised way with a stochastic gradient back-propagation training algorithm. The number of neurons on the hidden layer is found by validation on an independent validation set.

The hybrid NN-HMM was trained using the following procedure :

1. Cluster the feature vectors from the train database using a  $k$ -Means algorithm (initialization). The value  $k$  is given and defines the number of grapheme classes (whose posterior probabilities are predicted by the neural network). The clustering algorithm defines an annotation of the database at feature level.
2. Train the neural network on the annotated base of feature vectors
3. Compute the sequences of observation probability with the new neural network for all words in the train set.
4. Train the HMM on the sequences of observation probability using the Baum-Welch algorithm.
5. Decode the training set with the hybrid NN-HMM recognizer in order to create an annotated base of feature vectors.
6. Go back to item 2. until no improvement is observed on the recognition rate

The convergence is usually observed after 3 to 5 iterations. Note that this procedure is similar to the Expectation-Maximization algorithm, with 2. and 4. being the maximization steps and 3. and 5. being the Expectation steps.

## 2.4 Context-dependent character modeling

For one of the GMM-HMM system, we have used context-dependent models of characters, called trigraphs, as described in [2]. The main drawback of context dependent models is the very high number of possible different trigraphs, leading to a too large number of parameters to estimate on the available training database. Hence, parameter clustering is considered :

transition matrix tying and a state-level tree-based clustering permits to divide the number of state definitions by ten, and the final number different trigraphs by three. The binary questions defining the decision trees have been specifically created for an offline cursive Latin handwriting recognition task.

The training of context-dependent character models is performed as follows :

- Monograph initialization : build character HMMs with 8 states and 1 Gaussian per state; estimate the models with the Baum-Welch algorithm.
- Trigraph initialization : for each character, all the trigraphs (listed from the training lexicon) centered on this character are initialized with the corresponding monograph HMM.
- Trigraph estimation : run the Baum-Welch algorithm with the constraint that all trigraphs centered on a same letter share the same transition matrix.
- Trigraph states clustering : cluster the states for each position of each central character using a top-down clustering algorithm based on binary decision trees
- Gaussian mixture increase : alternate gaussian splitting and re-estimation until 20 gaussian distributions per mixture is reached.

Tree-based clustering of states was chosen instead of data-driven clustering because of its usefulness for decoding : if the test and train lexicons are different, any unseen trigraph of the new lexicon can be modeled thanks to decision trees.

## 3 Combination methods

In this work, we consider the combination of the outputs of  $K$  isolated word recognizers. For every image of word to be recognized, each recognizer, denoted  $R_i$ , provides a list of  $N$  hypothesis  $h_{ij}$  along with scores  $s_{ij}$ .

### 3.1 Rank-based combination methods

#### 3.1.1 Plural Vote

In this method, the hypothesis which is the most frequent in the first  $N$  ranks for each recognizer is considered as the best hypothesis. The result of the combination is the word  $w^* \in W$  such that :

$$w^* = \arg \max_{w \in W} \left( \sum_{i=1}^K \delta(w, \bigcup_{j=1}^N h_{ij}) \right)$$

	Grapheme MLP-HMM	GMM-HMM HTK	GMM-HMM HCM
Segmentation	grapheme	sliding window	sliding window
Feature vector size	74	56	56
Number of states per character HMM	3	8	8
Observation Classifier	Multi-layer perceptron	20 gaussian mixture	20 gaussian mixture
Gaussian Training algorithm	-	incremental	direct
Context-dependant character models	no	yes	no
Library	A2iA	HTK	HCM

**Table 1. Overview of characteristics of the 3 word recognizers.**

with

$$\delta(w, X) = \begin{cases} 1 & \text{if } w \in X \\ 0 & \text{otherwise} \end{cases}$$

### 3.1.2 Borda Count

For this method, a score is computed depending on its rank in the list. If lists of  $N$  candidates are considered, the first-place candidate receives  $N$  votes, the second candidate  $N - 1$ , etc, and the candidate ranked last receives 1 vote. The result of the combination is the word  $w^* \in W$  such that :

$$w^* = \arg \max_{w \in W} \left( \sum_{i=1}^K N - r_i(w) + 1 \right)$$

and  $r_i(w)$  is the rank of word  $w$  in the list provided by  $R_i$ . If  $w$  is not in the list,  $r_i(w) = N + 1$ .

### 3.1.3 Exponential Borda Count

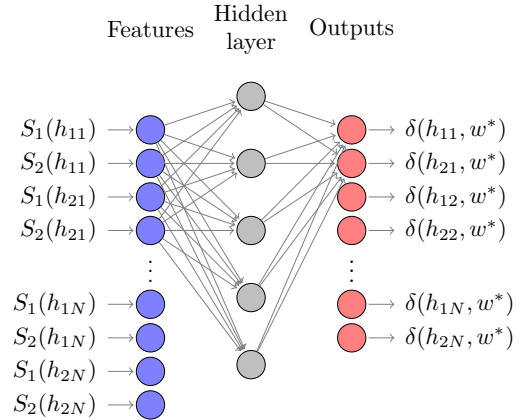
The Exponential Borda Count method was presented as an enhancement to the original Borda Count combination method [5]. The vote of candidate at rank  $r_i$  is changed from  $N - r_i + 1$  to  $(N - r_i + 1)^p$ ,  $p \geq 1$  where  $p$  is optimized on an independent validation database. The result of the combination is the word  $w^* \in W$  such that :

$$w^* = \arg \max_{w \in W} \left( \sum_{i=1}^K (N - r_i(w) + 1)^p \right)$$

## 3.2 Score-based combination methods

For score-based combination methods, each recognizer should be able to give a score to any word, even if it is not in the N-best hypothesis. In this case, the score of this word according to the recognizer is set to 0. This allows to extend the score function  $S_i$  for a given recognizer  $R_i$  to all words in the vocabulary :

$$S_i(w) = \begin{cases} s_{ij} & \text{if } \exists j \text{ such that } h_{ij} = w \\ 0 & \text{otherwise} \end{cases}$$



**Figure 1. Topology of the neural network for the combination of two recognizers (not all connections are shown).**

In our experiments, the scores  $s_{ij}$  are normalized over the  $N$ -best word candidates for each recognizer.

### 3.2.1 Sum Rule

Even if the sum-rule is one of the easiest way of combining lists of candidates, it is quite resilient to estimation errors [7]. The result of the combination is the word  $w^* \in W$  such that :

$$w^* = \arg \max_{w \in W} \left( \frac{1}{K} \sum_{i=1}^K S_i(w) \right)$$

### 3.2.2 Combination with a neural network

We propose an extension of the combination model based on a Neural Network previously proposed by Al-Hajj [3]. The previous method allowed to combine the best response of  $K$  recognizers into a single response. A drawback of this method is that if none of the recognizers outputs the right answer in first position, the combination has no chance of finding the true class (no

reordering of answers is made). One way to improve this method is to consider for each recognizer a list of candidates, and to combine the different list with the neural network.

The input vector of our combination neural network is the score given by each recognizer to each hypothesis at rank  $j$  in all the lists of candidates, taken for all  $j$  :

$$[S_k(h_{ij})] \quad \forall i, k \in \{1, \dots, K\} \quad \forall j \in \{1, \dots, N\}$$

The size of the input vector is  $K^2 \times N$ . For the training of the neural network, we define the target output vector as a binary vector of size  $K \times N$ , where for each recognizer, is coded if the hypothesis at rank  $j$  was the correct word or not:

$$[\delta(h_{ij}, w^*)] \quad \forall i \in \{1, \dots, K\} \quad \forall j \in \{1, \dots, N\}$$

where  $\delta$  is the kronecker function. The number candidates in the list  $N$  and the number of hidden cells are found empirically with a validation set (in our experiments,  $N=5$  and we used 30 hidden units). The neural network topology for combining two recognizers is given on Fig. 1. After training, the neural network provides a rescoreing of the candidates  $h_{ij} \quad \forall i \in \{1, \dots, K\} \quad \forall j \in \{1, \dots, N\}$ . Then the sum-rule is applied to merge the answers which correspond to the same word. The word with the highest score is selected as the result of the combination.

## 4 Experiments

### 4.1 The Rimes/ICDAR2009 database

We have trained and tested our combinations of handwritten word recognizers on the Rimes database [1] and use the ICDAR2009 [6] data and evaluation procedure : 44197 word images are given for train, 7542 word images for validation, and 7464 word images for test. The training lexicon includes 4500 words, the validation lexicon includes 1600 words and so does the test lexicon. It can be noted that the lexicons are case and accents sensitive and that lexicons are different from training/validation to test. All the recognizers were trained on the train database, the hyper-parameters for the recognizers or for the combinaison were optimized on the validation set and all the recognition results are given on the test set. Three different tasks were defined during the ICDAR2009 competition :

- the WR1 task where each word of the test is associated to a list of 99 words chosen randomly among test words in addition to the right transcription.

Recognizer/combination	Task	
	WR2	WR3
Grapheme MLP-HMM	81.4±0.9	75.7±1.0
GMM-HMM HTK	80.7±0.9	76.1±1.0
GMM-HMM HCM	77.6±1.0	72.0±1.0
Sum Rule(HTK,HCM)	83.6±0.8	79.5±0.9
Sum Rule(HTK,MLP-HMM)	89.1±0.7	85.4±0.8
Borda Count	85.0±0.8	82.4±0.9
Plural Vote	86.3±0.8	83.2±0.9
Exp. Borda Count	88.6±0.7	85.4±0.8
Sum Rule	89.9±0.7	86.8±0.8
Neural Network	90.3±0.7	86.9±0.8

**Table 2. Correct recognition rates in % with confidence intervals for the different combinations of recognizers for the tasks WR2 and WR3 (case insensitive and accent less).**

- the WR2 task where the given dictionary is composed of all the test words (1612 words).
- the WR3 task where the given dictionary contains also words of the training dataset (5334 words)

We report here only results on task WR2 and WR3 that we consider more difficult and more realistic than WR1.

### 4.2 Results

The recognition results of the systems obtained with the different combinaison schemes are presented on Table 2. The reported rates are computed without taking into account the case and accent (case insensitive and accent less) for the tasks WR2 and WR3. The recognition rate of each individual recognizer is also given. This table shows that the error rate can be reduced by 50% by combining several classifiers : on average, each classifier achieve 80% of recognition rate individually and more than 90% of recognition can be achieved by combining them. Moreover, the combination of classifiers based on different technologies is more efficient than the combinaison of similar classifiers, as noted by [5]. The combination of two classifiers based on sliding window HMM (HTK and HCM) yield recognition rates of 83.6% (WR2) and 79.5% (WR3) whereas combining a sliding windows HMM and a grapheme based MLP-HMM yield recognition rates of 89.1% (WR2) and 85.4% (WR3).

Regarding the combination methods, the methods based on the rank (Borda Count, PluralVote, Exponential Borda Count) are less efficient than the methods us-

System	GT		normalized GT	
	Top1	Top10	Top1	Top10
TUM	93.2	99.0	93.4	99.0
A2iA-Telecom	89.2	97.2	90.3	98.7
UPV	86.1	98.0	86.4	98.0
Siemens	81.3	96.4	81.7	96.4
IRISA	79.6	92.8	80.0	93.3
LITIS	74.1	94.9	75.6	94.9
Itesoft	59.4	76.7	68.1	86.9

**Table 3. Correct recognition rates in % for the different systems in competition at ICDAR 2009 for the task W2.**

System	GT		normalized GT	
	Top1	Top10	Top1	Top10
TUM	91.0	98.3	91.4	98.4
A2iA-Telecom	85.4	95.5	86.9	98.1
UPV	83.2	96.8	83.8	96.9
IRISA	74.7	90.3	75.7	91.5
Siemens	73.2	93.4	74.4	93.5
LITIS	66.7	91.6	67.4	93.5
LSIS	52.4	54.9	57.3	60.4
Itesoft	50.4	72.9	57.6	82.6

**Table 4. Correct recognition rates in % for the different systems in competition at ICDAR 2009 for the task W3.**

ing the recognition score (Sum rule, neural network). The best recognition rates are reached with the sum-rule and the neural network combinations, with similar recognition rates at 90.0% (WR2) and 86.9.0% (WR3).

The Tables 3 and 4 summarize the recognition results of the ICDAR 2009 competition. We have added the recognition results of the A2iA-Telecom ParisTech system with accents than were not submitted to the competition. With these new results, the A2iA-Telecom ParisTech system rank second on both W2 and W3 tasks.

## 5 Conclusion

In this paper, we have presented the recognition system submitted by A2iA, Telecom ParisTech and the University of Balamand to the ICDAR 2009 handwritten word recognition competition. This system is a combination of 3 recognizers based on different feature extractions (grapheme and sliding windows) and different HMM (based on multi-layer perceptron or gaussian

mixture models). We have have proposed a novel n-best combinaison with neural network and compared several combination scheme. We have shown that the combinations based score (sum rule and our novel method) outperform the combinations based on rank. New results on the ICDAR 2009 test database complete the result already published: with theses results, our system finishes in second position.

## References

- [1] E. Augustin, M. Carré, E. Grosicki, J.-M. Brodin, E. Geoffrois, and F. Preteux. Rimes evaluation campaign for handwritten mail processing. In *Proc. of the Workshop on Frontiers in Handwriting Recognition*, pages 231–235, 2006.
- [2] A.-L. Bianne, C. Kermorvant, and L. Likforman-Sulem. Context-dependent hmm modeling using tree-based clustering for the recognition of handwritten words. In *Proc. of the Document Recognition & Retrieval Conference*, 2010.
- [3] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel. Combining slanted-frame classifiers for improved hmm-based arabic handwriting recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 31(7):1165–1177, 2009.
- [4] H. ElAbed and V. Margner. Reject rules and combination methods to improve arabic handwritten word recognizers. In *Proc. of the Int. Conf. on Frontiers in Handwriting Recognition*, 2008.
- [5] V. Frinken, T. Peter, A. Fischer, H. Bunke, T.-M.-T. Do, and T. Artieres. Improved handwriting recognition by combining two forms of hidden markov models and a recurrent neural network. In *Proc. of the Int. Conf. on Computer Analysis of Images and Patterns*, pages 189–196, Berlin, Heidelberg, 2009. Springer-Verlag.
- [6] E. Grosicki and H. ElAbed. Icdar 2009 handwriting recognition competition. In *Proc. of the Int. Conf. on Document Analysis and Recognition*, 2009.
- [7] J. Kittler, M. Hatef, R. P. W. Duin, and J. Matas. On combining classifiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.
- [8] S. Knerr, E. Augustin, O. Baret, and D. Price. Hidden markov model based word recognition and its application to legal amount reading on french checks. *Computer Vision and Image Understanding*, 70(3):404–419, 1998.
- [9] C. Mokbel, H. Abi Akl, and H. Greige. Automatic speech recognition of arabic digits over the telephone network. In *Proc. of the Int. Conf. on Research Trends in Science and Technology*, 2002.