S. Knerr, E. Augustin

A2iA
Tour CIT, BP 59
75749 Paris Cedex 15, France
sknerr@a2ia.iway.fr

A Neural Network-Hidden Markov Model Hybrid for Cursive Word Recognition

Keywords: cursive handwriting recognition, Neural Network - Hidden Markov Model Hybrid, check recognition, legal amount recognition, optical character recognition.

Preferred conference track: Systems and Applications (Document Analysis, Character Recognition, Handwriting).

S. Knerr, E. Augustin

A2iA
Tour CIT, BP 59
75749 Paris Cedex 15, France
sknerr@a2ia.iway.fr

Abstract

We present a Neural Network - Hidden Markov Model Hybrid for the recognition of cursive words. Here, in contrast to conventional HMMs, the neural network provides the grapheme observation probabilities, and the Hidden Markov Models (HMM) provide the transition probabilities which are used to compute word probabilities. During the training of the hybrid, the HMMs provide the targets for the training of the neural network. The proposed approach models words with ergodic HMMs and is designed for small vocabularies such as the recognition of legal amounts from bank cheques. An extension of the approach to letter models which can be concatenated in order to form word models and which allow for large vocabularies is also briefly discussed. We report results obtained on a large data base of words from French cheques, showing recognition rates close to 93% for the 30 word vocabulary relevant for French legal amounts.

A Neural Network-Hidden Markov Model Hybrid for Cursive Word Recognition

1Introduction

This paper presents a Neural Network - Hidden Markov Model Hybrid for cursive word recognition and its application to the recognition of legal amounts from French cheques. Recently, Hidden Markov Models (HMMs) which benefit from a wide body of experience in the field of speech recognition have been more and more applied to handwriting recognition, on-line as well as off-line [Bengio et al., 1992, Bengio et al., 1995, Bourlard et al., 1995, Kimura et al., 1993, Knerr et al., 1997, Kundu et al., 1989, Manke et al., 1995, Seiler et al., 1997]. The basic idea here is that handwriting can be interpreted as a left-to-right sequence of ink signals, analog to the temporal sequence in speech. However, in order to also account for the 2-D character of handwriting, we segment words into smaller 2-D objects, called graphemes. Within our approach, letters can be composed of 1, 2, or 3 graphemes (see also Figure 5).

The motivation for the work on the hybrid presented here originates from a critical analysis of our earlier work on cursive word recognition using conventional HMMs [Knerr et al., 1998]. The major shortcoming of the conventional HMMs we have used is the vector quantization level which does not convey enough information about the handwriting to later stages of the recognition process. Following Bourlard et al., we have chosen to compute the observation probabilities with a neural network instead of the commonly used Gaussian mixtures [Bourlard et al., 1994]. The local observation probabilities, in our system we use graphemes as the basic observation units, are provided via a neural network and the transition probabilities are provided by the HMMs. The training procedure of the neural network maximizes posterior probabilities in contrast to the maximum likelihood estimation for the Gaussian mixtures.

We applied the NN-HMM Hybrid to the problem of recognizing cursive words from legal amounts on French cheques. This particular problem benefits from a small vocabulary of 30 words including a class for noise and for numerals. The latter models the centimes in the legal amount which are sometimes written as numerals and not as words. On the other hand, the handwriting on French cheques is of bad quality as can be seen from some examples in Figure 4, and the segmentation of the handwriting from the cheque document is difficult because of background pictures on the cheque forms and a lack of standards for French cheque forms. Figure 1 shows an example of a French bank cheque. Moreover, the reader-sorters currently used on the French market impose black and white images as the starting point for the recognition process. Many recognition problems will be alleviated as soon as gray scale images are available.

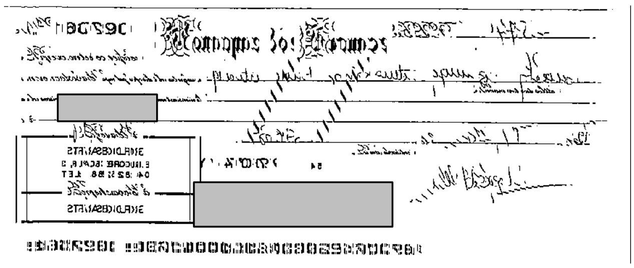

Figure 1: Example of a French bank cheque.

The work presented here is part of the A2iA INTERCHEQUE recognition system. An earlier version of the complete INTERCHEQUE system has been presented in [Knerr et al., 1997]. Besides the word recognizer discussed in this paper, the complete system also uses a global word recognizer which proceeds without segmenting words into smaller objects [Gorsky, 1994].

2NN-HMM Hybrid

In the proposed NN-HMM Hybrid, the observation probabilities $P(state_i\,(t) \mid grapheme\,(t))$ are provided via a neural network and the transition probabilities $P(state_i\,(t) \mid state_j\,(t-1))$ are provided by the HMM, as in a conventional non-hybrid HMM. Figure 2 shows an overview of the complete hybrid system. The left side of the figure shows the processing steps at recognition time: (1) segmentation of the word into graphemes, (2) feature extraction for each grapheme, (3) computation of the posterior class probability $P(state_i\,(t) \mid grapheme\,(t))$ for each grapheme using the neural network, (4) computation of likelihoods for the 30 word classes using the HMMs with their transition probabilities, and (5) computation of the posterior word class probabilities via Bayes rule. The right side shows the training procedure of the NN-HMM Hybrid involving the training of the HMMs, computation of the target values for the Neural Network, and the training of the neural network. The training procedure for the hybrid will be discussed in the next section. Note, that the only class labels necessary for the training of the hybrid are the word labels. No ground truthing at the grapheme or character level needs to be provided.

In our first experiments with NN-HMM Hybrids we use ergodic HMMs to model words. In an ergodic HMM each state may transit to all other states of the model, including itself [Rabiner et al., 1986]. Therefore, the topology of the models is fixed, and the only parameter is the number of states.

For applications with larger vocabularies it is prohibitive to build a separate model for each word. The usual approach in speech recognition is to model sub-units of words by individual HMMs, and then to concatenate these HMMs in order to build word models. We use the grapheme as the sub-unit, and we have built 3-state left-to-right HMMs which model letters. Within our system, letters are formed of 1-3 graphemes. The letter models can be concatenated at recognition time in order to form left-right models of words according to a given lexicon. We have used a set of conventional HMMs similar to this type as a bootstrap for the NN-HMM Hybrid. More details are presented at the end of the next section.

A single neural network is used for all word models. We use a Multilayer Perceptron with one hidden layer of sigmoid units and a Softmax output layer [Bridle, 1990]. One of the advantages of using a Softmax output function is that the outputs automatically sum to 1. According to Bayes rule, the estimates of the posterior probabilities P($state_i$ (t) | grapheme (t)) computed by the neural network are divided by the prior state probabilities P($state_i$) which results in scaled likelihoods p(grapheme (t) | $state_i$ (t)). These scaled likelihoods are used as observation probabilities in the HMMs. In conventional HMMs, the latter are usually computed from the features using Gaussian mixture densities [Levinson et al., 1983].

3Training the NN-HMM Hybrid

In this section, we will first discuss the training of the neural network, then the training of the HMMs, and finally the training of the complete hybrid which alternates between training the neural network and training the HMMs.

The neural network is trained in order to provide the grapheme observation probabilities in the hybrid scheme. The training of the neural network involves the minimization of the Kullback-Leibler distance between the target values d_k^i and the network outputs s_k^i, where the index i runs over all P training samples and the index k runs over all C output classes:

$$J_{KL}(W)=\sum_{i=1}^{P}\sum_{k=1}^{C} d_k^i \log \frac{d_k^i}{s_k^i}$$

Both the targets d_k^i and the network outputs s_k^i take values in the interval [0, 1]. It has been shown that if the target values represent posterior probabilities or if $d_k^i=1$ for the true class and $d_k^i=0$ for all other classes, then the outputs s_k^i of the trained neural network classifier approximate posterior class probabilities [Lippmann et al., 1991].

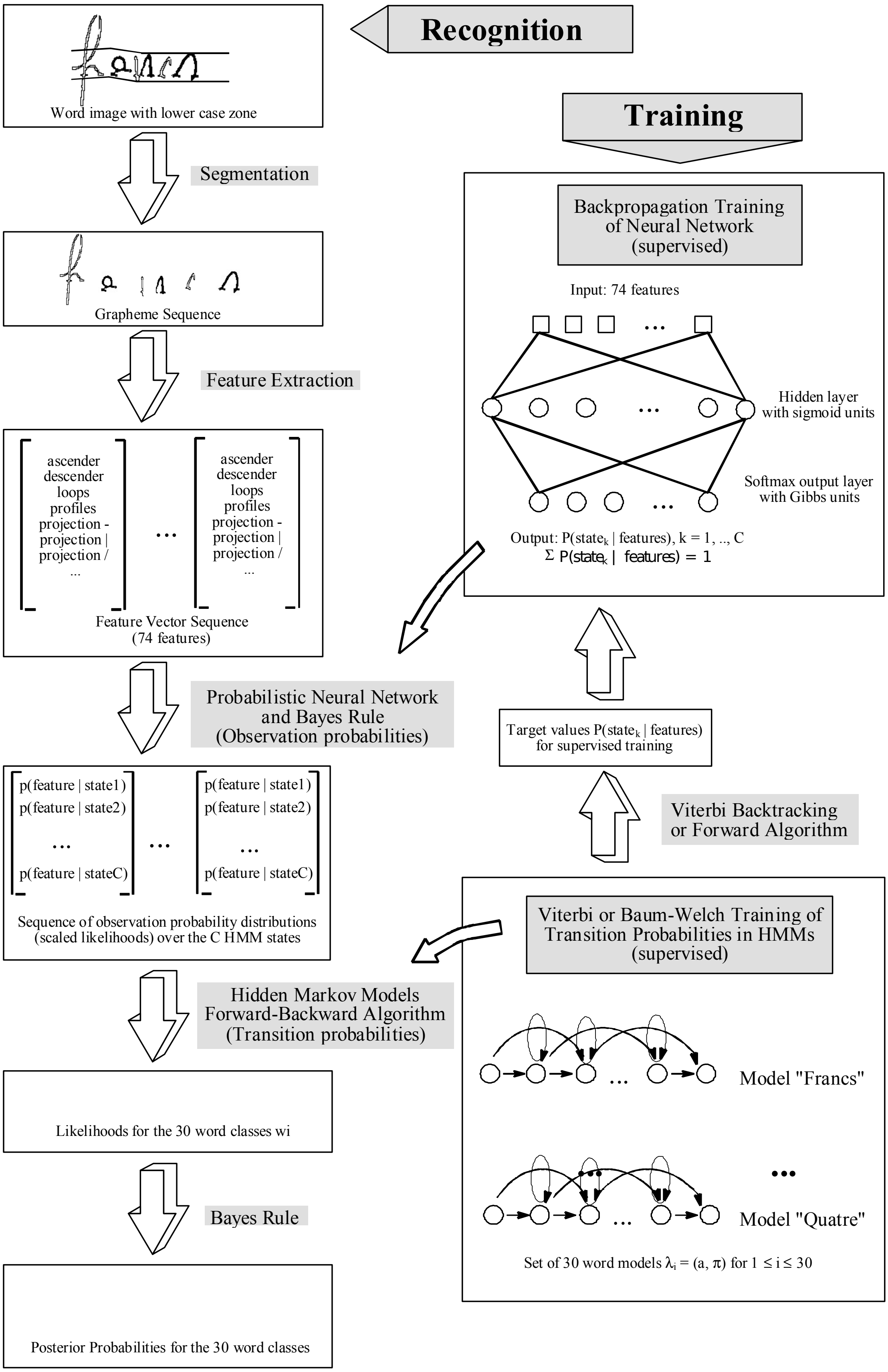

Figure 2: Overview of the Neural Network - Hidden Markov Model Hybrid. The left side shows the recognition process, the right side shows the training procedure.

The optimal number of hidden units is found empirically and depends on the number of classes C, the number of training samples P available, and the distribution of the samples in the input space. The results reported in this paper have been obtained with a neural network with 100 hidden units. However, in all our experiments, performances have shown no critical dependence on the choice of the number of hidden units, as long as the network provides the minimal capacity to solve the problem. The cost function $J_{KL}(W)$ is minimized by using the on-line backpropagation algorithm, i.e. a stochastic gradient algorithm. The step size of the gradient is decreased at each epoch during training. In order to avoid over-fitting, the performance of the classifier is continuously estimated on a validation set, and the set of weights W which gives the best performance on this validation set is kept. This typically corresponds to a few epochs, i.e. each training sample is only presented a few times (< 10) during the training of the neural network.
Note, that the training of the neural network is discriminant at the grapheme level, i.e. there is a competition between the output classes since the outputs sum to 1, and each output is optimized during training by samples of its own class as well as by samples of other classes.

The transition probabilities of the HMMs are learned by maximizing the likelihood p(obs | model) for each word model. Here, obs is the grapheme sequence representing the observation. For this purpose, either Baum-Welch training or Viterbi training can be used [Rabiner et al., 1986]. With Viterbi training, only the best path for a given word sample is used in order to update the HMM parameters. With Baum-Welch training, all paths are considered taking into account their relative importance reflected by the overall probability of the path. Baum-Welch training usually leads to better performances and is less dependent on the initialization of the HMM parameters. Therefore, we train each of the 30 word models using the Baum-Welch algorithm and the observation probabilities computed by the neural network. Note, that using a maximum likelihood cost function leads to non discriminant training at the word level, i.e. each word model is optimized based on samples of its own class only. Thereby, a model learns to respond with a high probability when presented a sample of its own class, but it does not learn to respond with a low probability when presented a sample of another class which shows a high resemblance to its own class. This problem will be partly reflected by the strong confusion of the words "six" and "dix" in the application discussed later.

Training the complete hybrid proceeds by an EM like iterative training scheme where the training of the neural network and the HMM training alternate [Bourlard et al., 1995]:

1. If we start the hybrid training procedure with a set of trained HMMs, then the first step of a training iteration consists of applying Viterbi backtracking (0/1 targets) or the forward-backward algorithm (targets in [0..1]) in order to provide the target values for the training of the neural network. All words of the training data base, i.e. their grapheme sequences, are analyzed by their corresponding HMM. Note, that each word is only analyzed using the HMM corresponding to its own word class. In the case of Viterbi backtracking, the best path through the HMM is computed, and each grapheme (t) is associated with a single state s*(t). This provides the input/output-target pairs for the training of the neural network: grapheme feature vector (t) / {d*=1, all other d_k=0}. If the forward-backward algorithm is used, each grapheme is associated with a probability distribution over the C states of the HMM. Therefore, the input/output-target pairs for the neural network training are: grapheme feature vector (t) / {d_1=P($state_1$(t)|word obs), ..., d_C=P($state_C$(t)|word obs)}.
2. All input/output-target pairs from all word classes are used to train the neural network as discussed above.

3. Then, the HMMs are re-trained using the observation probabilities P(state$_i$ | grapheme feature vector (t)) computed by the newly trained neural network.
4. Resume the training procedure at step 1.

As for the neural network training, the recognition performance of the complete hybrid is evaluated on a validation set after each iteration of the hybrid training scheme (steps 1-4). Our experiments (see section 4) have shown that the word recognition performance of the hybrid reaches a maximum after about three iterations. When the Viterbi approximation is used in order to compute the targets for the neural network training, the word recognition performance slowly descends after reaching the maximum. This overtraining effect is even less important (less than 0.1% during the next three iterations) when the full probabilities are used as the targets for the neural network training. However, after training the hybrid, we keep the set of parameters for the neural network and for the HMMs which lead to the maximum performance on the validation set.

From the above it is clear that such a hybrid system needs to be jump-started. Either a trained neural network or an initial set of already trained HMMs is necessary. The first can be obtained by training a neural network on a data base of labeled graphemes, but the labeling process is tedious and expensive. Therefore, we opted to bootstrap the hybrid system from a set of conventional left-right HMMs trained on the same 30 word classes and using the same feature vectors.

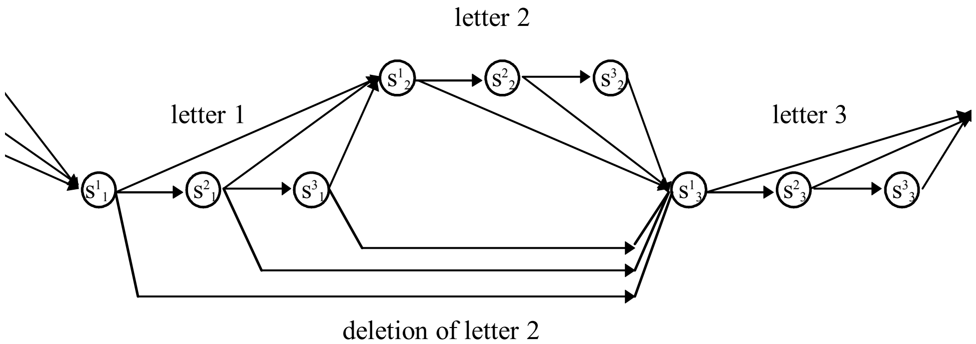

Figure 3: Part of a constraint HMM modeling 3 consecutive letters with 1, 2, or 3 grapheme states each. The second letter can be deleted.

A strongly constrained HMM has been designed for each word class: each letter in the word is modeled by 3 consecutive states as shown in Figure 3. In contrast to the letter models discussed in section 2, no parameter tying is applied to the models here, i.e. each word has its own model of a given letter. There are 23 different letter classes in the 30 word vocabulary relevant for the recognition of French legal amounts. These 23 classes include one class for *num* and one for *noise* (both will be explained in the next section). From each state of a letter it is possible to transit to the first state of the next letter or to the first state of the next but one letter. Thereby, a letter model can account for letters with 1, 2, or 3 graphemes. Viterbi backtracking on these models therefore leads to associations of graphemes with letter states (first, second, or third state of a specific letter in a given word), independently of the word class. This is important in our approach, where we use only one neural network for all word classes. In the next section, we will report results obtained with hybrids which have 23, 44, or 54 states (neural network outputs). In the first case, each state of the hybrid is associated with all three states of a letter class in the conventional left-right models. In the second case, 23 hybrid states account for the first state of the letter classes and 21 hybrid states account for the second and third states. In the third case, all statistically important letter states of the conventional left-right models have their counterpart in the hybrid system. Thereby, using Viterbi backtracking, it is possible to compute the targets for the first iteration of the neural network training. Then, HMM training and neural network training are alternated within the above hybrid training scheme.

4Application to the recognition of cursive words from French bank cheques

Reading the legal amount on bank cheques involves the recognition of cursive handwriting which is often of bad quality and badly segmented from other graphical objects on the cheque form, such as printed text or background pictures. See Figure 4 for some examples of legal amounts. On the other hand, this application benefits from a small vocabulary, typically 20 to 30 words. Therefore, it is possible to build a separate word model for each word in the vocabulary.

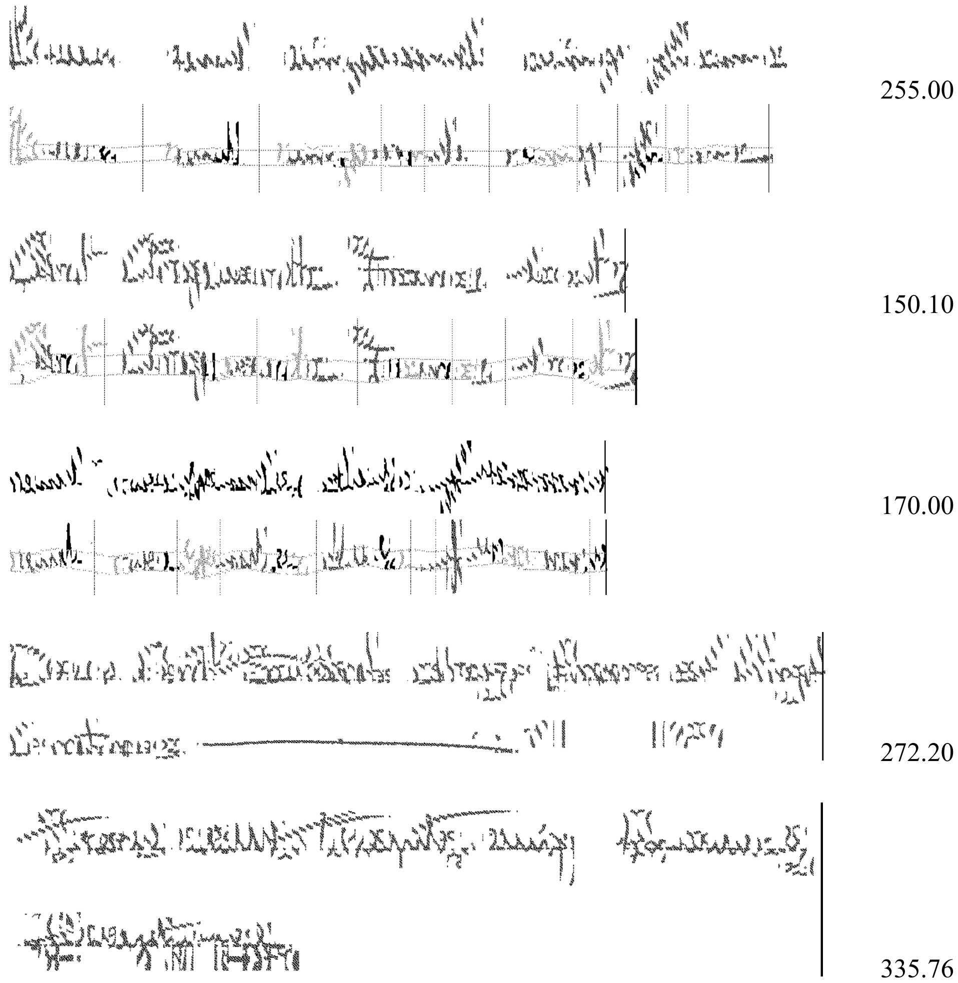

Figure 4: Some examples of French legal amounts after extraction from the cheque form and removal of printed lines and noise. For the first three examples, the word cuts, the graphemes and the effect of slant correction are also shown. The last two examples show legal amounts which extend over two text lines. All legal amounts in this Figure are recognized correctly by our system, except the numerals of the centimes in the last example. The true amounts are given on the right side of the Figure.

For the French recognition system, 30 word classes have been defined: {un, deux, trois, quatre, cinq, six, sept, huit, neuf, dix, onze, douze, treize, quatorze, quinze, seize, vingt, trente, quarante, cinquante, soixante, cent, mille, francs, frs, centimes, cts, et, *noise*, *num*}. The *noise* class accounts for objects in the text line which are not words belonging to the legal amount, such as text printed on the cheque form, lines drawn on the cheque by the writer in order to prevent any later changes to the amount, parts of background pictures, etc. The *num* class accounts for any numerals which sometimes appear in the centimes part of the legal amount. These numerals are later recognized separately by an algorithm which is similar to the courtesy amount recognition algorithm described in an earlier publication [Anisimov et al., 1995]. "frs" and "cts" are abbreviations for "francs" and "centimes" respectively. The difference between upper case and lower case letters has not been modeled explicitly, i.e. no special word classes have been defined for words which start with upper case letters or which are entirely written in upper case letters. The word "cent" which sometimes takes an "s" at the end has also been left to implicit modeling by the corresponding HMM.

A data base of about 50,000 cheques from various French banks has been used throughout the experiments. All cheque images are black and white and have a resolution of 240 dpi. As already discussed in earlier publications, many of the problems encountered in our experiments would be alleviated by starting from gray scale images [Knerr et al., 1997]. For the purpose of the work discussed in this paper, the legal amounts have been segmented into words and each word has been slant corrected and segmented into graphemes (Figure 5). Two reference lines have been computed on the complete legal amount which divide the image in an ascender zone, a descender zone, and a zone which contains the lower case letters without their ascenders and descenders. Words have been ground truthed by operators. No labeling at the grapheme or character level has been provided. The 170,000 words of the data base have been divided into a training set of about 130,000 words and a test set of about 40,000 words.

For each grapheme in a given word, a 74 dimensional feature vector is computed. The geometrical features are normalized with respect to the height of the lower case zone in order to make the features invariant with respect to the size of the handwriting. The features include: (1) height and width of the grapheme, ratio of height to width, (2) ascenders, descenders, (3) ink pixel densities in the ascender, descender, and lower case zone, (4) surface of loops, (5) left, right, upper, lower profile (distance of ink to respective side of the bounding box), (6) number of transitions between black and white pixels along horizontal, vertical, and the two diagonal directions, (7) density projections for the same four directions.

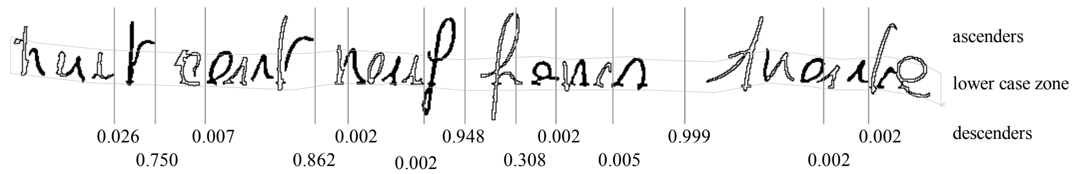

Figure 5: French legal amount (809.30 F). The amount has been slant corrected, and the ascender zone, descender zone, and lower case zone have been detected. Word cuts with probabilities are indicated by vertical bars, and the color of consecutive graphemes alternates between black and white.

We have trained the NN-HMM Hybrid on the 130,000 word training set, with a bootstrap from a set of conventional HMMs. The word recognition performance as evaluated on a validation set has reached a maximum after 3 iterations. From there on, the neural network continues to improve slightly at each iteration, but the word recognition rate at the output of the HMMs stays constant or slowly decreases.

The probabilistic neural network presented in section 2 computes posterior class probabilities P($state_i$ | features) from these feature vectors, with $1 \le i \le C$. Several values for the number of states C have been tried: C = 23, 44, 54. The value 23 corresponds to the number of distinct letter classes within the vocabulary of French legal amounts, including again a class for *num* and one for *noise*. These posterior class probabilities for the graphemes are divided by the a priori probabilities of the corresponding states in order to obtain scaled likelihoods p(features | $state_i$), i.e. the grapheme observation probabilities. Using these observation probabilities, the likelihoods for the 30 word classes can be computed from the trained HMMs (using the transition probabilities) by means of the forward algorithm. Finally, the posterior word class probabilities are computed using again Bayes rule.

	Grapheme Reco. Neural Network	Word Recognition NN - HMM Hybrid				
	REC(1)	REC(1)	REC(2)	REC(4)	REC(8)	Avg. Pos.
Convent. left-right HMM	-	82.5	91.5	96.5	98.9	1.46
Hybrid, 23 states, targets 0/1	53	89.4	95.4	98.3	99.5	1.25
Hybrid, 44 states, targets 0/1	48.4	91.4	96.5	98.6	99.6	1.19
Hybrid, 54 states, targets 0/1	47.3	91.8	96.7	98.7	99.7	1.18
Hybrid, 23 states, targets [0...1]	53.2	89.1	95.3	98.2	99.5	1.25
Hybrid, 44 states, targets [0...1]	48.4	91.8	96.5	98.8	99.7	1.18
Hybrid, 44 states, targets [0...1] + word priors	48.4	92.9	97.1	99.0	99.7	1.15

Table 1: Recognition results on the 40,000 word test data base. REC(K) gives the percentage of correct answers among the top K candidates. Avg. Pos. is the average position of the correct answer in the candidate list. Note that all results except for the last row have been obtained without using the prior word class probabilities.

Table 1 presents results on the 40,000 words test data base for 28 word classes ("centimes" = "cts" and "francs" = "frs"). Note that all results except for the last row in the table have been obtained without taking the prior word class probabilities into account. Using the word class priors, the results go up by about 1% in recognition performance. The best word recognition performance so far has been obtained with a 44 state hybrid using the full probabilities for the hybrid training procedure: 92.9% recognition rate, and an average position of the true word class in the candidate list of 1.15. 99% of all words in the test data base have been among the top 4 choices in the candidate list.

When increasing the number of states in the hybrid, the recognition performance on the word level improves, while the recognition performance at the grapheme level (output of the neural network) decreases. This seems normal considering that the neural network has to discriminate between more classes which share more and more resemblance. On the other hand, the hybrid uses the extra degrees of freedom in order to model words in more detail. Note, that when the hybrid is bootstrapped from conventional left-to-right HMMs, the 23 state hybrid models all graphemes of a letter by the same state, whereas the 44 state or 54 state hybrid uses 2 or 3 states for each letter, which is more realistic. However, the relationship between states and letters established when bootstrapping is lost to a large extend in the subsequent iterations of the hybrid training scheme.

Comparing the results obtained with a hybrid training scheme using Viterbi backtracking (neural network targets 0/1) to the results obtained using the full probabilities (neural network targets [0...1]), the difference is only small. This has two explications: first, our data base is large, and therefore the 0/1 targets approximate the full probabilities. Second, when looking at the distributions of the full target probabilities, more than 90% of the targets fall into the intervals [0..0.1] and [0.9...1]. Therefore, the difference between the two methods is small. This is not true when small training sets are used.

As can be seen from Table 1, we have gained close to 10% in recognition rate by replacing the conventional HMMs by NN-HMM Hybrids. Note that the conventional HMMs and the hybrids have been trained on the same date bases using the same feature set. For the conventional HMMs however, a simple vector quantization has been used for the computation of the observation probabilities from the feature vectors. The hybrid also outperforms a global word recognizer by more than 10% in recognition rate [Knerr et al., 1997]. An error analysis has shown that most of the recognition errors in the hybrid system are due to (i) noise or extraneous objects in the legal amount image which have not been removed before recognition or (ii) confusions between classes with a high resemblance such as "six"/"dix", "treize"/"seize", or "cts"/"et".

We have integrated the Hybrid with 44 states into the A2iA INTERCHEQUE system. The 44 states hybrid seemed to be the best trade-off between recognition performance and processing time. At the legal amount level, the current system obtains about 60% recognition. The result from the legal amount recognition is combined with the result from the courtesy amount recognition which leads to a recognition rate of 65-73% for about 1% substitution at the cheque level. The performance at the cheque level depends on (i) the percentage of cheques with centimes (centimes are often written in the second legal amount line and are difficult to recognize), on (ii) the percentage of cheques with printed amounts (amounts printed by cash-registers can be anywhere on the cheque form and are often of very bad quality), and on (iii) the distribution of amounts. The A2iA INTERCHEQUE system applies a final reject rule which depends on the amount recognized: the larger the amount, the stricter the reject rule [Knerr et al., 1997].

Comparisons to other word recognition systems in the literature and on the market are difficult due to the non-availability of common data bases. However, a word recognition rate of about 93% for a 30 word vocabulary on real life handwriting of bad quality is certainly among the best recognition performances reported. Olivier et al. have presented results of about 70% recognition rate obtained on a 27 word vocabulary using a data base of 2,400 French cheques [Olivier et al., 1995]. Their recognition system is based on graphemes and HMMs. The SRTP of La Poste has obtained about 84% recognition rate on a data base of 3,000 French postal cheques using a 30 word vocabulary [Gilloux et al., 1995]. Their system is based on a HMM/RBF-neural network hybrid using a very simple feature set. Note that amounts on French postal cheques are much easier to recognize due to the absence of background pictures on the cheque forms. Using a global word recognition approach, Guillevic et al. have obtained about 72% recognition rates on the legal amounts from US cheques [Guillevic et al., 1995]. Saon et al. have obtained about 80% recognition rate on an A2iA data base which is comparable to the data bases used in this paper [Saon et al., 1997]. They have used an HMM-MRF (Markov Random Field) approach which is based on an analysis of pixel columns and, therefore, does not need graphemes. However, this approach is computationally heavy.

5Conclusions and future work

It has been shown that the integration of grapheme and word recognition into a single system, in our case a NN-HMM Hybrid, results in superior performances and that it has the practical advantage of making obsolete the labeling at the grapheme or character level. Only the much less time consuming labeling at the word level needs to be provided. The price to pay is that the hybrid needs to be bootstrapped. However, the conventional HMMs which we used for this bootstrap also do not need any labeling at the grapheme or character level.

We have obtained word recognition rates close to 93% for the 30 word vocabulary of legal amounts from French bank cheques.

In the future, we will replace the maximum likelihood training of the HMMs by a discriminant training procedure. We will also integrate more context at the grapheme level into the recognition process by using a TDNN type neural network [Waibel et al., 1989].

6Acknowledgments

It is a pleasure to thank Hervé Bourlard for stimulating discussions, and Olivier Baret, Nikolai Gorsky and David Price for their contributions to the work presented in this paper.

7References

V. Anisimov, N. Gorsky, D. Price, O. Baret, S. Knerr, "Bank Check Reading: Recognizing the Courtesy Amount", *Proceedings of the Third International Computer Science Conference*, Hong Kong, 1995.

Y. Bengio, R. De Mori, G. Flammia, R. Kompe, "Global Optimization of a Neural Network - Hidden Markov Model Hybrid", *IEEE Transactions on Neural Networks*, Vol. 3, No. 2, pp. 252-259, 1992.

Y. Bengio, Y. LeCun, C. Nohl, C. Burges, "LeRec: A NN-HMM Hybrid for On-Line Handwriting Recognition", *Neural Computation*, Vol. 7, No. 6, pp. 1289-1303, 1995.

H. Bourlard, N. Morgan, "*Connectionist Speech Recognition: A Hybrid Approach*", Kluwer Academic Publishers, 1994.

H. Bourlard, Y. Konig, N. Morgan, "REMAP: Recursive Estimation and Maximization of a posteriori Probabilities", *ICSI Technical Report* TR-95-064, 1995.

J.S. Bridle, "Probabilistic Interpretation of Feedforward Classification Network Outputs, with Relationship to Statistical Pattern Recognition", in *Neurocomputing*, NATO ASI Series, F. Fogelman, ed., Vol. 68, pp. 227-236, 1990.

M. Gilloux, B. Lemarie, M. Leroux, "A Hybrid Radial Basis Function Network/Hidden Markov Model Handwritten Word Recognition System", *Proceedings of the International Conference on Document Analysis and Recognition 1995*, Vol. 1, pp. 394-397, 1995.

N. Gorsky, "Off-line Recognition of Bad Quality Handwritten Words Using Prototypes", *Fundamentals in Handwriting Recognition*, NATO ASI Series F, Vol. 124, S. Impedovo, ed., Springer, pp. 199-217, 1994.

D. Guillevic, C.Y. Suen, "Cursive Script Recognition: A Sentence Level Recognition Scheme", *International Workshop on Frontiers in Handwriting Recognition IV*, pp. 216-223, Taipei, Taiwan, 1994.

D. Guillevic, C.Y. Suen, "Cursive Script Recognition Applied to the Processing of Bank Cheques", *Proceedings of the International Conference on Document Analysis and Recognition 1995*, Vol. 1, pp. 11-14, 1995.

F. Kimura, M. Shridar, Z. Chen, "Improvements of a Lexicon Directed Algorithm for Recognition of Unconstrained Handwritten Words", *Proceedings of the Second International Conference on Document Analysis and Recognition*, Tsukuba, pp. 18-22, 1993.

S. Knerr, V. Anisimov, O. Baret, N. Gorsky, D. Price, J.C. Simon, "The A2iA INTERCHEQUE System: Courtesy Amount and Legal Amount Recognition for French Checks", *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 11, No. 4, pp. 505-548, 1997.

S. Knerr, E. Augustin, "Hidden Markov Model Based Word Recognition and its Application to Legal Amount Reading on French Checks", to be published in the special issue on Document Image Understanding of the *Journal for Computer Vision and Image Understanding*, 1998.

A. Kundu, Y. He, P. Bahl, "Recognition of Handwritten Words: First and Second Order Hidden Markov Model Based Approach", *Pattern Recognition*, Vol. 22, No. 3, pp. 283-297, 1989.

R.P. Lippmann, M.D. Richard, "Neural Network Classifiers Estimate Bayesian a posteriori Probabilities", *Neural Computation*, Vol. 3, pp. 461-483, 1991.

S. Manke, M. Finke, A. Waibel, "NPen++: A Writer Independent, Large Vocabulary On-Line Cursive Handwriting Recognition System", *Proceedings of the International Conference on Document Analysis and Recognition*, Montreal, 1995.

C. Olivier, T. Paquet, M. Aliva, Y. Lecourtier, "Recognition of Handwritten Words Using Stochastic Models", *Proceedings of the International Conference on Document Analysis and Recognition 1995*, Vol. 1, pp. 19-23, 1995.

L.R. Rabiner, B.H. Juang, "An Introduction to Hidden Markov Models", *IEEE ASSP Magazine*, pp. 4-16, 1986.

G. Saon, A. Belaid, "Off-Line Handwritten Word Recognition Using a Mixed HMM-MRF Approach", *Proceedings of the Fourth International Conference on Document Analysis and Recognition*, Ulm, 1997.

R. Seiler, M. Schenkel, F. Eggimann, "Cursive Handwriting: Off-Line versus On-Line Recognition", *Progress in Handwriting Recognition*, A.C. Downton, S. Impedovo, eds., World Scientific, 1997.

A. Waibel, T. Hanazawa, G. Hinton, K. Shiano, K. Lang, "Phoneme Recognition using Time-Delay-Neural Networks*", IEEE Transactions on Acoustics, Speech and Signal Processing*, 1989.