Lexicon	10	100	1000
RR(1)	99.0	96.0	87.7

Table 7. Performances obtained with the context-dependent upper-case letter model

And by resuming again our interpolation technique, we finally obtain the following results (Table 8).

Lexicon	10	100	1000
RR(1)	99.2	96.3	88.9

Table 8. Performances obtained with the definitive interpolated model

The improvements shown in Table 8 are equivalent to an error rate reduction of 11.1%, 11.9% and 5.1% for lexicon sizes of 10, 100 and 1000 respectively. Table 7 shows that the context-dependent upper-case letter model does not improve the model described in section 3.2 if we drop the interpolation technique. This is explained by the fact that upper-case letters that trained the corresponding models previously are now distributed over the two kinds of context-dependent upper-case letter models considered in this new modeling. Therefore, the improvement in the accuracy of the new model is balanced by the diminution of the samples to train the parameters of the new models. This also explains the superiority of the interpolation technique over the new model (Table 8) for which more parameters were not reliably estimated than that in section 3.2.

4. Conclusion and future work

We presented in this paper the latest developments carried out to enhance the performance of our word recognition system. In summary, the overall recognition rate improvements are 0.3%, 1.3% and 2.6%, leading to error rate reductions of 27.3%, 26.0% and 19.0% for lexicon sizes of 10, 100 and 1000 respectively. This is a significant improvement given that only the modeling aspect was investigated. To complete this study, we are currently testing other approaches to find out the limits of our system by insisting on the modeling aspect. For instance, we are developing a new letter model to characterize the dependence between the left and right segments of an oversegmented letter into two segments by allowing in this model several over-segmentation paths, thus overcoming implicitly the assumption of independency between observations in an HMM.

Acknowledgements

This research was supported by grants from NSERC, FCAR, and the French Post.

References

[1] Poritz A.B., "Hidden Markov Models: A Guided Tour," Proc. of ICASSP'88, pp.7-13, 1988.

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

[3] Bahl L., Jelinek F., Mercer R. "A maximum likelihood approach to speech recognition", IEEE PAMI 5:179-190, 1983.

[4] Lee K.F., Hon H.W, Hwang M.Y., Huang X., "Speech Recognition Using Hidden Markov Models: A CMU Perspective," Speech Communication 9, Elsevier Science Publishers B.V. (North-Holland), 1991, pp.497-508.

[5] Rabiner L.R., "A tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proceedings of the IEEE, 77 (2): 257-286, Feb. 1989.

[6] Ha J.Y., Oh S.C., Kim J.H., "Recognition of Unconstrained Handwritten English Word with Character and Ligature Modeling," IJPRAI, Vol.9, pp.535-556, 1995.

[7] Bercu S., Lorette G., "On-Line Handwritten Word Recognition: An Approach based on Hidden Markov Models," IWFHR-III, pp.385-390.

[8] Gillies A.M., "Cursive Word Recognition Using Hidden Markov Models," Proc. of the 5th USPS Advanced Technology Conf., pp.557-562, Nov 30-Dec 2, 1992.

[9] Caesar T., Gloger J.M., Kaltenmeier A., Mandler E., "Handwritten Word Recognition Using Statistics," The Institution of Electrical Engineers. Printed and published by the IEEE, Savoy Place, London WC2R 0BL, UK, 1994.

[10] Gilloux M., Leroux M., Bertille J-M., "Strategies for Cursive Script Recognition Using Hidden Markov Models," Machine Vision and Applications, Vol. 8, pp.197-205, 1995.

[11] Cho W., Lee S-W., Kim J.H., "Modeling and Recognition of Cursive Words with Hidden Markov Models," Pattern Recognition, Vol. 28, No. 12, 1995, pp.1941-1953.

[12] Chen M.Y., Kundu A., Shrihari S.N., "Variable Duration Hidden Markov Model and Morphological Segmentation for Handwritten Word Recognition," IEEE Trans. on Image Processing, Vol. 4, No. 12, Dec. 1995, pp.1675-1687.

[13] A. El-Yacoubi. "Modélisation Markovienne de l'écriture manuscrite. Application à la reconnaissance des adresses postales," PhD thesis, Université de Rennes I, France, Sept. 1996.

[14] El-Yacoubi A., Gilloux M., Sabourin R., Suen C.Y., "Objective Evaluation of the Discriminant Power of Features in an HMM-based Word Recognition System," BSDIA'97, pp.60-73, Nov. 02-05, Curitiba, Brazil.

[15] Morgan N., Bourlard H., "Continuous Speech Recognition," IEEE Signal Processing Magazine, May 1995, pp.25-42.

[16] El-Yacoubi A., Bertille J.M., Gilloux M., "Towards a more effective handwritten word recognition system," IWFHR IV, pp.378-385, Taiwan, Dec. 1994.

data, we must consider a learning database different from that used for training models M_1 and M_2. In our experiments, we used the *Baum-Welch* algorithm to train the λ parameters on a set of 4000 images. The interpolated model was then tested on the same test set as before. The results obtained are given in Table 4.

Lexicon	**10**	**100**	**1000**
RR(1)	99.0	95.4	87.6

Table 4. Performances obtained with the interpolated model

We see that the interpolated model improves the recognition rate by 0.1%, 0.4% and 1.3%, leading to an error rate reduction of 9.0%, 8.0% and 9.5% for lexicon sizes of 10, 100 and 1000 respectively. This is a significant improvement if we keep in mind that this interpolation technique mostly attends to more reliably estimate the parameters related to events that are not frequently seen and therefore the improvement only relies on a relatively small set of samples of the test set in which these events appear.

3.2. Improvement of the character model

As pointed out before, one of the error sources in our previous model is the splitting of a letter into 3 pieces. Even though this phenomenon is rather rare, it can cause a delay in the matching between the observation sequence extracted from a word image and the associated sequence of states. This likely leads to a misrecognition of the given word. Therefore, we have modified the character model architecture to allow it to model the splitting of a letter up to 3 segments (Figure 5).

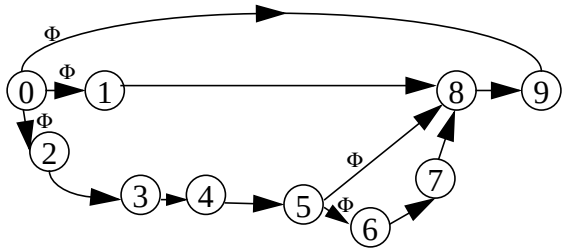

Figure 5. Character model taking into account up to 3 segments ($M^3{}_1$)

Transitions t_{58} and t_{56}, emitting null symbols, represent the probability of segmenting a letter into 2 or 3 pieces respectively. However, given that the latter phenomenon is rare, the associated parameters are likely to be not reliably estimated due to the lack of training samples that exhibit the desired events. The solution to this problem is to call for the "tied states" principle [3] by sharing the transitions involved in this new modeling (t_{56}, t_{58}, t_{67}, t_{78}) over all character models. Nevertheless, this procedure is not carried out when we are dealing with letters `W`, `m` et `w` for which the probability of segmentation into 3 segments is high and therefore there are enough samples to train separately the parameters corresponding to the third segment for each of these letters. According to this scheme, we trained the new model ($M_1{}^3$) on the same learning set as before using the Baum-Welch algorithm. The results obtained on the same test set are given in Table 5.

Lexicon	**10**	**100**	**1000**
RR(1)	98.9	95.4	87.8

Table 5. Results obtained with model $M^3{}_1$

By resuming the interpolation technique of section 3.1, we obtain the following results (Table 6).

Lexicon	**10**	**100**	**1000**
RR(1)	99.1	95.8	88.3

Table 6. Performances obtained with the interpolated model ($M^3{}_3$)

3.3. Context-dependent upper-case letter models

In speech recognition, the incorporation of context-dependent *phoneme* models such as *triphones* [4] have led to high performance recognition systems. This modeling was motivated by the high variability in the pronunciation of a phoneme depending on its left and right contexts and also by the availability of huge amount of training samples to reliably estimate the highly increased number of parameters. In handwritten word recognition, the effect of left and right contexts for a given character seems to be less important than in speech and we seldom have very large databases. However, it is clear that upper-case letters show different shapes according to the context related to the writing style: upper-case words or cursive words beginning with an upper-case letter. Moreover, in our approach, even if the shape of the letter does not change, the features that are extracted from it can be very different since they are based on global features such as ascenders which strongly depend on the writing style by way of the writing baselines. We also add the fact that the first letter in upper-case words often has a larger size compared with the other letters and hence an ascender is likely to be detected for this first letter. These three remarks have led us to consider a different model for an upper-case letter depending on its position in the word: first position (whether in an upper-case, mixed or cursive word) or a different position in an upper-case word. The results with this new modeling, when using the same learning and test databases, are reported in Table 7.

eter estimation is obtained by using a third HMM that may be viewed as an interpolation of the first two. In our approach, the new model (which we will refer to as M_2^2 to designate the second model taking into account up to 2 segments per character, the first one being M_1^2) is not obtained *via* the principle of the tied states but thanks to another concept. As we showed before, each observation emitted along a transition related to a shape is a combination of two observations corresponding to our two feature sets. We can consider a new model in which the two observations are assumed to be statistically independent, while keeping the same structure. In this case, if o_k^1 and o_k^2 respectively stand for the k^{th} symbol in the first and second symbol alphabets, the probability of emitting the pair of observations (o_k^1, o_k^2) while a transition t_{ij} is occurring is:

$$p(o_k^1, o_k^2 | t_{ij}) = p(o_k^1 | t_{ij}) \times p(o_k^2 | t_{ij}) \quad (1)$$

Note that we do not need any learning phase to train the new parameters. Indeed, we can directly compute them from the first model parameters, using:

$$p(o_k^1 | t_{ij}) = \sum_{m=1}^{N_2} p(o_k^1, o_m^2 | t_{ij}) \quad (2)$$

$$p(o_k^2 | t_{ij}) = \sum_{n=1}^{N_1} p(o_n^1, o_k^2 | t_{ij}) \quad (3)$$

N_1 and N_2 are respectively the size of the first and second alphabets. When tested on the same test set used before, this model leads to the following results (Table 2).

Lexicon	10	100	1000
RR(1)	99.0	95.1	86.9

Table 2. Recognition rate obtained with the new model

We see that we obtain almost the same results as those obtained with the first model. In fact, it is clear that the new model is less precise than the first one, since it does not take advantage of the redundancy brought by the dependency between the two sets of features we are using. On the other hand, the assumption of independency between these two sets leads to a more reliable estimation of the new model parameters. In this case indeed, the number of parameters to estimate in the first model is proportional to N_1 x N_2 while in the second one, it is proportional to $N_1 + N_2$. Hence, for the same learning database, the average number of examples to train each parameter is far greater for the second model than for the first one. Thus, the new model has a better generalization capability over unknown data. This interpretation is well proven by the recognition rates obtained with the two models when tested on the learning set (Table 3).

Lexicon	10	100
RR(1); $M^2{}_1$	99.2	96.0
RR(2); $M^2{}_2$	99.0	94.1

Table 3. Recognition rates obtained on the learning set

Now, let

$$p_1(o_k^1, o_k^2 | t_{ij}) = p(o_k^1, o_k^2 | t_{ij}, M_1^2) \quad (4)$$

$$p_2(o_k^1, o_k^2 | t_{ij}) = p(o_k^1, o_k^2 | t_{ij}, M_2^2) = p(o_k^1 | t_{ij}) \times p(o_k^2 | t_{ij}) \quad (5)$$

We may wish to use directly $p_1(o_k^1, o_k^2 | t_{ij})$ when it is reliable, and to replace it by $p_2(o_k^1, o_k^2 | t_{ij})$ when it is not. A convenient way to do this is to choose as a final estimate of $p(o_k^1, o_k^2 | t_{ij})$ a linear combination of $p_1(o_k^1, o_k^2 | t_{ij})$ and $p_2(o_k^1, o_k^2 | t_{ij})$ given by:

$$p(o_k^1, o_k^2 | t_{ij}) = \lambda_{ij} \times p_1(o_k^1, o_k^2 | t_{ij}) + (1 - \lambda_{ij}) \times p_2(o_k^1, o_k^2 | t_{ij}) \quad (6)$$

with λ_{ij} varying between 0 and 1. This equation can be interpreted as an associated Markov source in which each state is replaced by three states. However, since in our model the problem of reliable parameter estimation solely deals with transitions emitting a shape observation (and not null transitions or transitions modeling segmentation points), only states corresponding to these transitions are involved in the duplication process. Thus, we obtain the following associated model (Figure 4).

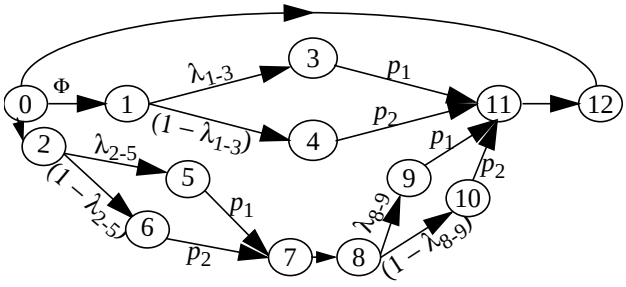

Figure 4. The interpolated character model

In this new HMM, p_1 and p_2 are assumed to be known and are given by trained models M_1 and M_2. Therefore, only parameters λ_{ij} must be trained. However, as these parameters are used to predict which model is better for unseen

of the bidimensional contour transition histogram of each segment in horizontal and vertical directions. We also have segmentation features whose aim is to characterize physical segmentation points or spaces between letters. Given that the two sets of shape-features are independently extracted from the image, a word is represented by two symbolic descriptions of equal length, each consisting of an alternating sequence of shape symbols and segmentation symbols.

2.2. Handwritten word modeling

When considering large vocabularies such as those involved in city name recognition, it is not possible to build a different model for each word. Therefore, our modeling is carried out at the letter level. As our segmentation process may produce either an under-segmentation, a correct segmentation, or an oversegmentation (into two segments) of a letter, we built a seven state left-to-right HMM having three paths to take into account these configurations (Figure 1). In this model, observations are emitted along transitions. Null transitions t_{06}, t_{01} and t_{02} model the three above configurations respectively. Transitions t_{15} and t_{56} respectively emit a shape-symbol and a segmentation-symbol encoding a correctly segmented letter while transitions t_{23}, t_{34}, t_{45} and t_{56} model the over-segmentation case. In the learning phase, the word model is built by concatenating the appropriate elementary letter models (Figure 2). In recognition, the word model is still built by concatenating letter models. However, since no information is available on the writing style of an unknown word, a letter model here actually consists of two models in parallel, associated with the upper and lower case modes of writing a letter (Figure 3). We should eventually add that each emitted observation along a transition corresponding to a shape is in fact a combination of two observations, related to the two sets of shape-symbols we are using.

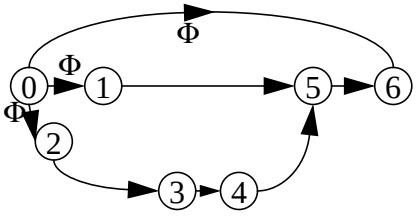

Figure 1. The character model

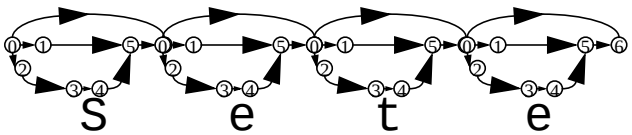

Figure 2. Training model for the word `Sete`

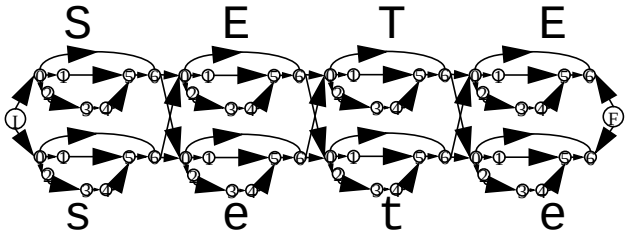

Figure 3. Recognition model for the word SETE

The word recognition system described above was trained on a learning set of 11,410 French city names manually extracted from real mail envelopes, and tested on a different set of 4,313 images. The performances obtained with this model for lexicons of sizes 10, 100 and 1,000 and without rejection are given in Table 1.

Lexicon	**10**	**100**	**1000**
RR(1)	98.9	95.0	86.3

Table 1. Performances in recognition

The system errors mainly come from splitting a letter into three pieces, ambiguity, poor images, or insufficient examples to reliably estimate some model parameters.

3. Optimization of the recognition phase

In this section, we present some strategies to make the recognition phase more robust, including the use of an interpolation technique to estimate more reliably some HMM parameters from sparse data, the development of a better HMM architecture to deal with the case of segmenting a letter into up to 3 segments, and finally the consideration of context-dependent models for upper-case letters where the context is given by the writing style of the word (cursive, mixed or upper-case).

3.1. Reliable parameter estimation from sparse data using an HMM-based interpolation technique

In real-world available databases, we seldom find a uniform distribution of examples over model parameters, mainly because the *a priory* probabilities of model classes are often not equal. As a matter of fact, some parameters will be reliably estimated, while some others will not. This issue was first discussed in [3] from a Markovian point of view. The idea is to keep the used model as is and to build an analogous model with the same structure, in which some states are *tied*. Two states are said to be tied when it exists an equivalence relationship between them; in other words transitions leaving each of these two states are analog and have equal probabilities. The model obtained by *tying* some states leads to a less precise model but having more reliably estimated parameters. Then, the final param-

Improved Model Architecture and Training Phase in an Off-line HMM-based Word Recognition System

A. El-Yacoubi[1], R. Sabourin[1,2], M. Gilloux[3] and C.Y. Suen[1]

[1] Centre for Pattern Recognition and Machine Intelligence, Concordia University
1455 de Maisonneuve Boulevard West
Suite GM-606, Montréal, Canada H3G 1M8

[2] Ecole de Technologie Supérieure
Laboratoire d'imagerie, de vision et d'intelligence artificielle (LIVIA)
1100 Notre-Dame Ouest, Montréal, Canada H3C 1K3

[3] Service de Recherche Technique de La Poste
Département Reconnaissance, Modélisation Optimisation (RMO)
10, rue de l'île Mâbon, 44063 Nantes Cedex 02, France

Abstract

This paper describes the latest developments to enhance the performance of our HMM-based handwritten word recognition system. These methods only deal with the recognition phase and involve the improvement of the HMM architecture as well as the optimization of the training phase. Experiments carried out on real data show that the proposed approaches lead to significant improvements in the accuracy of the system.

Keywords: Handwriting Modeling, Hidden Markov Models, Parameter Estimation.

1. Introduction

Handwriting and speech exhibit many similarities such as the sources of their variabilities, and the sequential nature of their signal. Over the past years, Hidden Markov models (HMMs) [1]-[2] have become the predominant approach to automatic speech recognition [3]-[5]. More recently, many researchers have applied these models to recognize handwritten words whether on-line [6] [7] or off-line [8]-[12] in which case the word bidimensional image is transformed into a sequence of observations as required in Markovian modeling. According to the way this sequence is obtained, two categories of approaches are distinguished: implicit segmentation-based approaches, which are inspired by those used in speech recognition and explicit segmentation-based approaches, which require a segmentation algorithm to split words into basic units such as letters. In [13] [14], we described an off-line explicit segmentation-based recognition system that makes use of HMMs to model characters and then words, and is applied to city name recognition. Although this system achieves high recognition rates, it can still be optimized especially in feature extraction and recognition phases. In this paper, we focus our interest on the optimization of the recognition phase by developing more discriminant letter models and by performing a better estimation of HMM parameters from insufficient data using an HMM-based interpolation technique. The organization of this paper is as follows. Section 2 recalls the main steps of our word recognition system. Section 3 describes the new developments proposed to enhance the performances in recognition. Finally, a conclusion is drawn in section 4.

2. System Overview

Our system is designed to recognize unconstrained handwritten words. Therefore, we must cope with all writing styles: handprinted, cursive or mixed. In this section, we present the outline of our approach. A more detailed description can be found in [13] [14].

2.1. Feature extraction

After the normalization of the word body and the character slant [16], our segmentation algorithm uses the upper contour minima and some heuristics to split the word image into a sequence of *segments*, each of which consisting of a correctly segmented, an under-segmented or an oversegmented letter. This sequence is transformed into a symbolic description by considering two feature sets. The first set is based on global features such as loops, ascenders and descenders while the second one is based on the analysis