WORD MODELING FOR HANDWRITTEN WORD RECOGNITION

Thierry PAQUET
PSI-La3i, Université de Rouen,
UFR des sciences,
76821 Mont Saint Aignan cedex,
France.
Thierry.Paquet@univ-rouen.fr

Manuel AVILA
LAC, I.U.T. de Chateauroux
2 Avenue F. Mitterand
36000 Chateauroux, France.
Manuel.Avila@univ-orleans.fr

Christian OLIVIER
IRCOM-SIC, Université de Poitiers,
UMR CNRS 6615, BP 179,
86960 FUTUROSCOPE cedex, France.
Olivier@sic.univ-poitiers.fr

Abstract

In this paper we investigate three different approaches for the global modeling and recognition of the words used to write the legal amount on French bank checks (27 lexicon entries), mainly written in *mixed cursive and discret* style. The first Model is a global one since it does not require any explicit letter level. The second Model is built to use the explicit concatenation of letter Models and is called "a letter reconstruction based approach". The third Model is able to give each grapheme its corresponding interpretation within a word (either part of a letter, letter or group of letters) and has been called a grapheme reconstruction based approach. To analyse the three approaches independently from a specific description, each of them uses the same segmentation process and feature set. The three approaches have been tested on real images of Bank Checks scanned for the French Postal Technical Research Service (SRTP).

1. Introduction

A computer unconstrained handwriting recognition has been the object of several studies over the past thirteen years and is still a challenging task [1][2][3]. Generally the difficulty of making a reading machine comes from the large variety of writing styles it has to deal with (from pure cursive to hand-printed). Furthermore, there is a wide diversity of handwriting even for the same writer. Up to now, the field of automatic handwriting recognition was only restricted to domains for which specific constrains could restrain the set of the possible solutions. But it is necessary to build reliable reading machines to read addresses on envelopes, amounts on bank checks, handwritten letters... These various applications need a particular lexicon either static or dynamic restricting the possible solutions.

When dealing with dynamic or large lexicon, handwritten words can only be recognized by identifying each of their letter. Except for hand-printed styles, in which the segmentation of words into individual characters is relatively simple, many efforts have been made to overcome the segmentation paradigm [4]. The most sophisticated approaches now include a segmentation-recognition scheme [5][6] to guide the segmentation process by the classification results.

With applications dealing with small lexicon (a few dozen) the segmentation paradigm can be overcome by using a global recognition scheme of individual words thanks to a suited description. From this point of view, ligature between cursive letters are not take into account in the word image. Consequently, neither the learning nor the recognition of word models require the knowledge of segmentation statistics. This could be the most ideal approach for word recognition but it is rather limited to a restricted vocabulary since it involves the computation of a matching score for each of the lexicon entries.

In this paper we investigate three different approaches for the global modeling and recognition of the words used to write the legal amount on French bank checks (27 lexicon entries), mainly written in *mixed cursive and discret* style. In section 2 we describe the three different modeling. In section 3 a brief description of the features used is given, as well as the principle of the segmentation process. Section 4 is devoted to the learning of the global word models and recognition results are presented on

real check images. In section 5 we discuss the results and investigate for future work.

2. Investigating on the global modeling of cursive words

Most languages use linear concatenation, of characters to produce words [7]. The global recognition of handwritten words therefore recognizes a word as a whole, using this *a priori* knowledge depending on the specificity of each lexicon, a global recognition process does not necessarily need to act on letters. For example the French words *francs* and *centimes* can generally be differentiated without having to recognize each letter, by using global descriptors such as upper and lower stroke position. On the contrary, some words such as *un* and *six* or *trente* and *huit* are generally difficult to differentiate without analyzing their letters. Generally not only the size of the lexicon but also the degree of proximity of words in the lexicon will necessitate a letter level analysis. As we have seen, the lexicon of French bank checks sometimes need a global recognition. Indeed, a restricted lexicon of 27 implies various strategies for the global recognition of words. Each of the three strategies encountered in this paper is derived from a particular assumption about the segmentation process involved. *The "global approach"* uses a left to right description of words and does not proceed to any analysis at the letter level. The second approach is an analytical one and assumes that characters of a word can be broken into several parts (over segmentation of characters) localized by the segmentation process. Consequently, the global modeling presented here is based on the reconstruction of letters from the analysis of consecutive segments and will be called a *"letter reconstruction based approach"* in the following sections. The third approach, derived from the Chen's works [8] studies both over and under segmentation of characters. The global modeling reconstructs graphemes from the analysis of consecutive segments and will be called a *"grapheme reconstruction based approach"* in the following pages. The three approaches are based on Hidden Markov Models to model each lexicon entry [9].

2.1. The global approach

A word model consists in a state sequence organized from left to right. According to the global modeling, each state does not necessarily model a specific letter within the word that is being modeled. The only constrain imposed by such a model is the left to right succession of states which reflects the left to right organization of the observed segmented graphemes and will evaluate of the probability to have the model for a particular sequence of graphemes.
Several problems arise when dealing with such models. First of all, let us compare the probabilities of each model to produce the observed grapheme sequence. They all have the same structure, i.e. the same number of states and topology. Consequently, short words such as the French *un* (one) will be modeled just like long words such as the French word *cinquante* (fifty). On the one hand, a single letter will be modeled by an average number of states (more in the first than in the second situation), which tends to reduce the average number of parameters per unit of letter for long word models. On the other hand, short words, described by few graphemes, must be lined up with a state sequence of a fixed length. This is obtained by introducing jumps up to three states between the different states of the model (figure 1). The second problem deals with the ability of such models to represent the various styles of handwriting encountered (presence of capital letters, pure cursive styles, mixed cursive and discrete characters, ...). For long words, the average number of parameters can be critical to render the various distortion of writing. A third problem concerns the choice of a left to right topology in the model either purely left to right or left to right with several parallel paths. The first study allowed us to choose a single left to right topology with fifteen states which, when using our specific segmentation process and features (see section 3), gives the best results. Each state of the model will reflect the most frequent situations encountered in the examples of the training database.
Let us recall that these models are learnt by using the Baum-Welch algorithm which uses an iterative scheme to adjust the parameters so as to maximize the probability of the observed sequence. Model identification is made in the recognition stage using a Bayesian decision by looking for the word model that enhances the probability of the model given the observation sequence. The results of the experiment using the global modeling are presented in section 4 and compared with the two other approaches presented below. The expected qualities of such models are their ability to take implicitly into account the variability of writing; thus no explicit analytical modeling of the segmentation process is required. However, there may be a risk of confusing words with a close global description, i.e. the same number of letters, upper and lower extensions at the same position.

2.2. The letter reconstruction based approach

As already seen in the introduction, analytical modeling depends roughly on the segmentation stage, which is compulsary when using this kind of approach. We assume here that most of the time, the segmentation process produces over segmentation points. Thus, the recognition strategy consist in find the adequate letter segmentation points amongst the set produced by the

segmentation process. This strategy will be guided by the results of a letter recognition process. Since letters can be composed of several graphemes (up to three graphemes, see table 3 in section 4.2), we have decided to model each letter of the lexicon by a left to right model with three states. This model allows to render most variations within a letter as well as ligatures between letters.

The word model will be the concatenation of the model of each letter that constitutes the word to be modeled. This is only possible when dealing with a small vocabulary, either static or dynamic. In this way, the recognition process will be the same as the global method.

However the learning phase is quite different. Indeed, we want to learn letter models in the word context and this implies to know the correct segmentation of the word examples in the learning database. Since we want the letter models to take into account the ligature between letters, it is necessary to learn letters in the context of the whole word. However, the Baum-Welch algorithm does not allow to constrain some intermediate paths in a simple way, in order to adjust the segmentation points between letters. We have thus decided to use the Baum-Welch algorithm to learn word models like for the global method and then to deduce letter models. The underlying hypothesis is that learning will converge to word model, whose structure corresponds to the model we are looking for. This will be discussed in section 4.2.

2.3. The grapheme reconstruction based approach

As we have seen, the global model does not resort to any explicit letter modeling while in the letter reconstruction based approach, the states of the model explicitly correspond to the global model of a specific letter. This third approach can be viewed as a grapheme reconstruction based approach. It is derived from the modeling proposed by M. Chen & al [8] where graphemes are assigned to an explicit piece of letter, letter or group of letters. This approach allows the modeling of both over and under segmentation of letters by introducing an explicit state for each segmentation situation e.g. each state is assigned to a specific grapheme, either part of an over segmented letter or part of two under segmented successive letters. This model is closely related to the segmentation process, since every possible segmentation situation is taken into account by the model, making the following assumptions :

1- a letter α is mostly segmented into three pieces corresponding to left, middle and right part of the letter denoted respectively α_L , α_M , α_R .

2- when a letter α is segmented into two pieces, it is assumed that the segmentation point is between the left part α_L and the middle part $\alpha_M\alpha_R$.

3- two letters at most can be in a single segment.

The first hypothesis is confirmed in most cases as can be seen in first row of table 3 which gives letter segmentation statistics on the learning database. The second hypothesis takes into account the fact that the beginning of a letter is often more careful than its end, and so, ends of letters are frequently absorbed by the middle part of the letter. Finally, no example to the contrary of the third hypothesis has been found in our database. 1 shows the model of letter α where:

- $Br(\alpha)$ stands for probability for letter α to be segmented.
- $1\text{-}Br(\alpha)$ is the probability for letter α to be joined to the next letter.
- $B'_3(\alpha)$ is the probability for letter α to be segmented into three pieces when it is segmented.

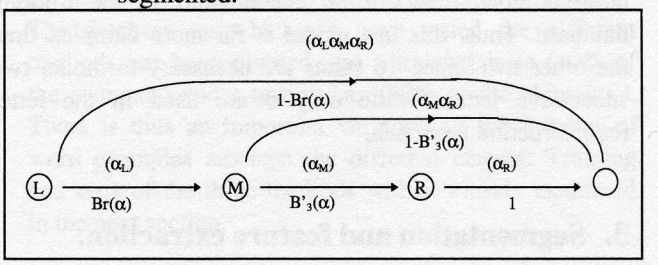


Figure 1: Explicit grapheme model of letter α

As can be seen from figure 1, 5 different states are used to model each possible grapheme within a segmented letter, they are :

$\alpha_L\alpha_M\alpha_R$ is the state when letter α is not broken

α_L is the first state of letter α when it is broken into at least two parts

$\alpha_M\alpha_R$ is the last state of letter α when it is broken into two parts

α_M is the middle state of letter α when it is broken into three parts

α_R is the last state of letter α when it is broken into three parts

Furthermore, in order to model the possible under segmentation situations between two successive letters (α and β) that can be encountered in the lexicon, the following 6 states are introduced keeping with figure 2 :

$\alpha_L\alpha_M\alpha_R\beta_L\beta_M\beta_R$ $\alpha_M\alpha_R\beta_L\beta_M\beta_R$ $\alpha_R\beta_L\beta_M\beta_R$

$\alpha_L\alpha_M\alpha_R\beta_L$ $\alpha_M\alpha_R\beta_L\alpha_R\beta_L$

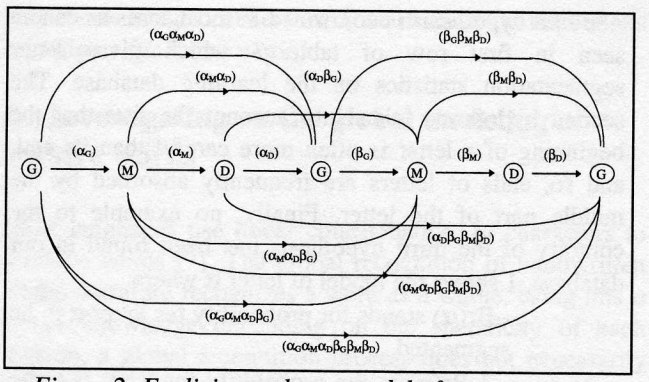


Figure 2: Explicit grapheme model of two successive letters.

Word models are thus derived from all the different state configurations that can be encountered in the training database. Thus, this last model is far more complex than the other two, since 16 states are necessary to model two successive letters while only 6 are used in the letter reconstruction approach.

3. Segmentation and feature extraction.

The three modeling presented in section 2 have been tested using the same segmentation process at the image level and using the same set of structural features for grapheme description. This description is a structural stroke based description. After extraction of strokes, we present the method used to code strokes into graphemes.

3.1 Segmentation and stroke description

The principles of the segmentation and strokes description have been presented earlier in [10]. The word description is based on the extraction of anchor points among the word axis. These points correspond explicitly to the intersection of the word skeleton with the middle axis. Indeed, since no dissection method has proved to be efficient in the context of cursive handwriting recognition, we have adopted a rather simple one. The retrieval of the segmentation into letters implies the problem of word recognition. A stroke description of the handwritten word is obtained in analyzing the word image skeleton between anchor points. In this study, 12 basic strokes have been considered (figure 3) and represent the most frequent stroke configurations between two successive anchor points. Using the stroke coding of figure 5, the stroke detection procedure can assign each word image a code sequence where the '/' symbol represents a segmentation point (anchor point) shown on figure 4.

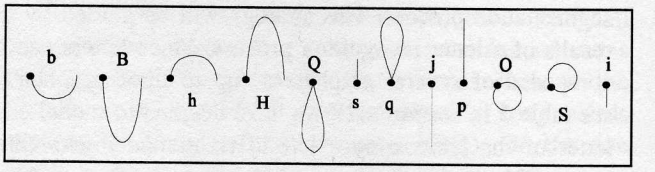
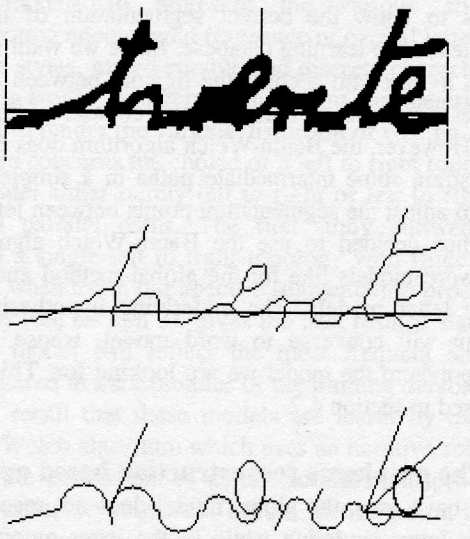
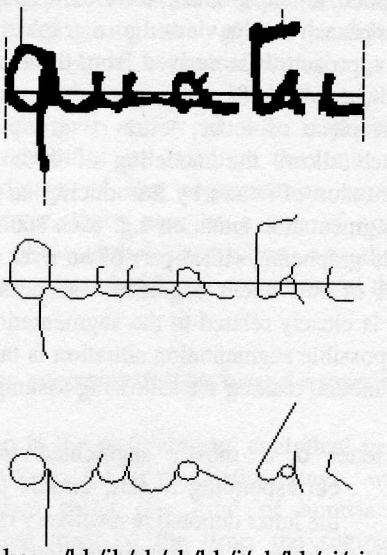


Figure 3: The 12 basic strokes and their coding.

Segmentation and stroke extraction are performed after pre-processing presented in [11] which consists in base line slant correction and normalization of the lower character height.



Strokes : /ih/ihp/b/h/b/hq/hi/sb/h/pbi/q



Strokes : /bh/jb/sb/sb/hb/i/pb/hb/si/si

Figure 4: Examples of stroke extraction and coding of words.

3.2 *grapheme coding*

The extracted stroke sequence can be organized in order to represent the unknown word as a primary grapheme sequence. A grapheme is made of the set of strokes extracted from one anchor point. A binary vector with 12 components allows the coding of the various situations observed on the training database. Nearly 500 different configurations have been listed on the database, from which 39,000 grapheme segments have been extracted (see section 4 for database description).

The selection of a grapheme alphabet was presented in [12]. The methodology consists in training Markov models for different order with various alphabets. We have chosen the alphabet which is the best compromise between the recognition rate, the size of the grapheme alphabet and the order of the optimal Markov model. Figure 5 shows the retained grapheme alphabet. It is built on a hierarchy on stroke information using the Shannon mutual entropy of each grapheme class in relation to the 27 words of the lexicon. As a consequence, each segment will be assigned one of the 14 classes depending on the strokes detected on the segment. As one can see in figure 5, the most informative classes of graphemes, in the vocabulary used, include upper and lower strokes, while small loops (code O or S) appear at the end of the hierarchy and bring little information as for upper and lower strokes.

if (p or q or H) and (j or q or B)	**then**	class n°1
else if (p or q) and h	**then**	class n°2
else if (p or q or H) and b	**then**	class n°3
else if p or q or h	**then**	class n°4
else if (j or Q) and h and b	**then**	class n°5
else if (j or Q or B) and h	**then**	class n°6
else if (j or Q) and b	**then**	class n°7
else if (j or Q or B)	**then**	class n°8
else if h and b	**then**	class n°9
else if h and i	**then**	class n°10
else if b and s	**then**	class n°11
else if (O or S) and i	**then**	class n°12
else if s and i	**then**	class n°13
else	**then**	class n°14

Figure 5: Grapheme alphabet encoding using stroke coding of figure 4.

A preliminary study [13] has allowed us to select the order for the grapheme alphabet using information criteria like Akaïke Information Criteria (AIC). Other criteria are presented in [12]. The order we have found for the alphabet used in this study is one. A simple Hidden Markov Model of order one is sufficient to represent words correctly. This criterion takes into account the size of the grapheme alphabet, and the size of the training database. These results concerning the order shows that it is not necessary to implement higher order with this alphabet.

4. Learning and recognition

We recall that the images used to test our methods have been provided by the Technical Research Service of La Poste (French Postal Technical Research Service (SRTP)). Databases are composed of binary images of French bank checks. The sentences are labeled at the word level. For the third method, we need to label words at the letter level to be able to learn the letter parameters. The database on which we have tested the different methods has been divided into a training base (40% of the elements) and a testing base(60% of the elements). There is thus an important variation of the number of word examples amongst the different classes. Training and tests of the three methods will be closely examined in the next section.

4.1 *Global modeling approach results*

In this approach, the model is composed of a state transition probabilities matrix, a matrix of observation probabilities, a vector of initial state probabilities and a vector of final state probabilities. We used an iterative method for training, based on the Baum-Welch algorithm. The transition state matrix and the observation matrix have been randomly initialized. The initial and the final state vectors have been initialized so as to ensure beginning in the first state and ending in the last state. During the training phase, 10 iterations have been performed to provide convergence on the training database. During the recognition phase, a recursive procedure is used to compute the probability of each model given the observed sequence of graphemes. Table 1 gives the results on training and testing databases.

A detailed analysis of the results shows that better results are obtained for the most frequent class of word such as "*francs*", "*vingt*", "*quatre*" and "*cent*" (see. **Erreur! Source du renvoi introuvable.** in appendix). The recognition rates are up to 78,2% of good recognition in TOP 1 for the word "*cent*". These results also show that some word classes are confused with others which have the same global shape. For example, the word "*deux*" is confused with words "*dix*" and "*trois*". Word "*quarante*" is confused with word "*quatre*" and word "*six*" is confused with word "*dix*". This analysis shows the

overall ability of the model to assimilate word shapes and word deformations, as shown by the kind of errors reported.

4.2 Letter reconstruction based approach results

A letter is composed of three states. This is justified by the fact that more than 95,7% of letters are composed of 3 graphemes at most (see first row of table 3). This model of letter can be viewed as a global letter model. The parameters of this method are made of the state transition probabilities matrix, the observation probabilities matrix, and the initial and final probability vectors. Initial and final vectors are initialized using statistics of letters in sentences in the database. The training phase is organized as described by the
algorithm.

1-Initialize global model letter.
by fixing letter topolog and using lexicon information on letters.
2-For each word of the training database :
2-1- Compose the local model word.
2-2- Use the same technical of estimation used in global method.
2-3- Report local cumulus to global cumulus. This report takes into account the frequency letter in words.
3-Re-estimate global model with global cumulus.
4-Go to step 2 until end test is valid.

The learning is stopped after 6 iterations on the training database. Table 2 shows that up to 91% of letters are correctly segmented with a gap of one grapheme. This justifies the use of a three states model for each letter.

In order to validate the learning of letter models, we analyzed the letter segmentation performances of the learnt word models on the training database using a Viterbi algorithm. The second row of table 3 gives the segmentation statistics computed using the results of the Viterbi algorithm. We can see that we are able to segment the word images using the learnt model in a similar manner to the real situation. So we can conclude that the observations are correctly aligned with states corresponding to letters and validate the learning algorithm. During the recognition, the same recursion procedure as in the global model is applied. Table 4 shows the performances of the method.

The detailed analysis of the results shows that better performances are obtained on word composed of the most frequent letters on the learning database : word "*cent*" is the best recognized word with up to 83% of good recognition. This method does not make any typical confusion between word models. Finally, we note that this method does not require a letter labeled training database to learn letter model parameters.

4.3 Grapheme reconstruction based approach results

The training stage consists in two phases. The first corresponds to the learning of the cursive script parameters, the second corresponds to the learning of the lexicon statistics. Thus components of the transition matrix are composed of statistics on cursive scripts and statistics on the lexicon. Performances of this method are given in Table 5.

The detailed analysis of these results shows major confusions for words composed of the same letters. For example, word "*dix*" is confused with words "*six*" and "*deux*" which have two letters in common. The word "*cent*" is confused with "*deux*" and "*huit*". In this case, we have only one letter in common; but we can also note that the letters "n" and "u" are often similar in the cursive script style.
The main confusion is between letters.

5. Discussion and Conclusion

The recognition results show that none of the three modeling prevails over the others. They all perform the correct recognition in 56% up to 58,7% of the cases for the first proposition (Top 1), while the correct solution is in the 10 first propositions (Top 10) in 82,9% up to 91,7 % of the cases. However the global model is always better than the two others. Table 6 gives recognition results for each of the 27 entries of the lexicon, and for the three approaches. Results are given by examining the presence of the correct solution in a list of 1, 2 and 5 propositions (Top 1, Top 2, Top 5).
The specific results of each approach for some particular entries of the lexicon are noticeable: Short words with two or three letters are always better recognized when a letter recognition is used. The global method gives the best results for the most frequent words in the learning database. The letter reconstruction method also gives good results for words having the most frequent letters. These last two remarks are closely related to the size of the databases, and particularly to the low number of examples for some lexicon entries which does not guaranty any significant learning of the global models. The letter reconstruction method is also sensible to the number of examples of letters used for learning. However this database effect is less important in this case

since a word can contain both frequent letters and some rare ones. This explains the lower results of the letter reconstruction method compared to the global one. In all cases, this explains the low number of iterations of the Baum algorithm.

Previous works on the same problem shown better results [14][15][16], however the experiments were conducted on different databases. A second remark can be made here about the feature set used in our experiment. This feature set was designed for the global approach for which robust features were retained. However they cannot describe letter variability in an omni-scriptor context and would be more adequate for a writer dependent system.

These database effects reflect however the general problems for different kinds of applications. Indeed, a global approach can only be applied to a restricted lexicon for which sufficient examples of each lexicon entries can be provided. When this is not possible, the only way of modeling handwritten words is to use analytical approach for which large databases of letter examples can be provided. This explains some good performances of the letter reconstruction approach for rare words that are constituted by more frequent letters. The grapheme approach is closely related to the number of letter transitions in the database. Some rare words such as *cinq* are badly learnt by the global method but contains some frequent letter sequences (that occur in word *cinquante* for example) which enforces the grapheme approach in this case.

This study shows that global and analytical approaches are complementary for two main reasons : - They are complementary in the way that the lack of examples learnt by the global approach is balanced in some cases by the number of examples learnt by the letter reconstruction based approach. However, even in the case of frequent words in the learning database such as *cent* or *dix*, the second approach gives better results. In this case of short words, letter information is of primary importance to take a decision. Finally a specific cooperation scheme could be designed from these results to improve the overall performances. Indeed, since the three approaches use the same feature set, the time performances would not be altered when introducing a cooperation scheme.

References

[1] R.G. Casey and E. Lecolinet, « A Survey of Methods and Strategies in character», *IEEE Transaction on PAMI*, Vol 18, N° 7, pp 690-706, July 1996.

[2] Y. Lu, M. Shridhar, « Character Segmentation in Handwritten Words-an Overview », Pattern Recognition, Vol. 29, No. 1, pp. 77-96, 1996.

[3] R.M. Bozinovic, S.N. Srihari, « Off-Line cursive Script Recognition », IEEE Transaction on PAMI , Vol. 11, No. 1, pp. 68-82, 1989.

[4] K.M. Sayre, « Machine Recognition of Handwritten Words : A Project Report », Pattern Recognition, Vol. 5, pp. 213-228, 1973.

[5] G. Kim, V. Govindaraju, "A Lexicon driven Approach to Handwritten Word Recognition for Real-Time Applications", *IEEE Transaction on PAMI*, Vol. 19,No. 4, pp. 366-379, April 1997.

[6] E. Lethelier, M. Leroux, M. Gilloux, "An Automatic Reading System for Handwritten Numeral Amounts on French Checks", *Proc. Of the third ICDAR*, pp. 92-97, Montreal, 1995, Canada.

[7] W. Cho, S.W. Lee & J.H. Kim, "Modeling and Recognition of Cursive Words with Hidden Markov Models", *Pattern Recognition*, Vol 28, N° 12, pp 1941-1953, 1995.

[8] M.Y. Chen, A. Kundu, J. Zhou, "Off-Line Handwritten Word Recognition Using a Hidden Markov Model Type Stochastic Network", *IEEE Transaction on PAMI*, Vol ; 16, No. 5, pp. 481-496, May 1994.

[9] L.R. Rabiner, "A Tutorial on Hidden Markov Model and Selectted Applications in Speech Recognition", roc. Of IEE, Vol. 77, No. 2, pp. 257-286, 1989.

[10] T. Paquet, Y. Lecourtier, "Recognition of Handwritten Sentences using a restricted Lexicon", *Pattern Recognition*, Vol. 26, No. 3, pp. 391-407, 1993.

[11] A. El Yacoubi, "Modélisation Markovienne de l'écriture manuscrite; Application à la reconnaissance des adresses postales", Thèse de doctorat, Université de Rennes I, (France), 1996.

[12] M. Avila, "Optimisation de Modèles Markoviens pour la reconnaissance de l'écrit ", Thèse de doctorat, Université de Rouen, (France), 1996.

[13] C. Olivier, T. Paquet, M. Avila, Y. Lecourtier, "Optimal Order of Markov Models Applied to Bakn Checks", *International Journal of Pattern Recognition and Artificial Intelligence,* Vol. 11, No. 5, pp. 789-800, 1997.

[14] J.C. Simon, O. Baret, "A System for the Recognition of Literal Amounts of Checks", *Proc. of DAS'94*, pp. 135-155, 1994.

[15] J. V. Moreau, B. Plessis, O. Bougeois, and J.L. Plagnaud, "A Postal Checks reading System", Proc. of ICDAR'91, pp. 758-766, Saint-Malo, 1991, France.

[16] M. Gilloux, M. Leroux, "Recognition of cursive script amounts on Postal Cheques", *Proc. of* JET POST'93, 1993, Nantes, France.

TOP	1	2	3	4	5	6	7	8	9	10
Training Database	89.5%	97.2%	99.1%	99.7%	99.7%	99.9%	100.0%	100.0%	100.0%	100.0%
Test Database	58.7%	71.0%	76.8%	80.4%	83.1%	85.5%	87.3%	88.9%	90.2%	91.7%

Table 1 : Global method performances.

grapheme gap	-5	-4	-3	-2	-1	0	1	2	3	4	5
Percentage	0,1%	0,1%	0,5%	2,5%	**15,1%**	**61,6%**	**14,2%**	3,5%	1,0%	0,5%	1,0%

Table 2: Average positions of letter segmentation points in the training database.

Number of graphemes per letter	1	2	3	4	5
Observed on the database	41.5%	45.8%	10.6%	1.9%	0.2%
Computed using Viterbi	43.9%	40.1%	11.7%	2.9%	1.4%

Table 3: Letter segmentation statistics.

TOP	1	2	3	4	5	6	7	8	9	10
Training Database	75.0%	86.7%	90.6%	92.9%	94.4%	95.4%	96.0%	96.3%	96.7%	96.8%
Test Database	55.9%	67.3%	73.5%	77.6%	80.6%	82.3%	84.1%	85.4%	86.5%	87.5%

Table 4 : Letter reconstruction based approach performances.

TOP	1	2	3	4	5	6	7	8	9	10
Training Database	79.36	88.52	91.68	93.02	93.51	93.99	94.08	94.22	94.31	94.44
Test Database	57.88	66.74	71.63	73.95	76.37	78.11	79.45	80.40	81.61	82.97

Table 5: Grapheme reconstruction based approach performances.

	Global method			Letter approach			Grapheme approach		
TOP	1	2	5	1	2	5	1	2	5
zéro	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
un	33.3%	33.3%	33.3%	**44.4%**	**66.7%**	**88.9%**	**44.4%**	44.4%	44.4%
deux	**46.9%**	**69.6%**	90.3%	41.1%	64.7%	**90.8%**	45.8%	66.1%	75.3%
trois	30.5%	42.9%	62.9%	21.9%	37.1%	62.9%	**48.5%**	**63.8%**	**67.6%**
quatre	**69.9%**	**82.9%**	**94.7%**	64.6%	70.3%	79.7%	69.1%	76.8%	82.1%
cinq	39.3%	54.9%	**83.6%**	40.2%	53.3%	81.1%	**48.3%**	**58.2%**	63.9%
six	8.3%	12.5%	18.8%	8.3%	**39.6%**	**93.8%**	**14.5%**	20.8%	25.0%
sept	17.6%	19.6%	27.5%	23.5%	**47.1%**	**74.5%**	**27.4%**	31.3%	35.2%
huit	13.7%	25.5%	47.1%	15.7%	27.5%	**56.9%**	**27.4%**	**37.2%**	39.2%
neuf	**40.0%**	**56.7%**	63.3%	26.7%	46.7%	**76.7%**	35.0%	46.6%	48.3%
dix	46.7%	57.9%	78.5%	**72.9%**	**88.8%**	**100.0%**	10.2%	12.1%	12.1%
onze	0.0%	0.0%	0.0%	0.0%	0.0%	25.0%	0.0%	0.0%	0.0%
douze	0.0%	6.7%	13.3%	0.0%	13.3%	**26.7%**	**13.3%**	**13.3%**	13.3%
treize	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
quatorze	0.0%	0.0%	9.1%	**9.1%**	**18.2%**	**36.4%**	0.0%	0.0%	0.0%
quinze	14.3%	22.9%	28.6%	**14.3%**	**28.6%**	**51.4%**	14.2%	14.2%	14.2%
seize	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
vingt	**72.5%**	**83.9%**	**97.7%**	61.0%	78.0%	89.4%	71.5%	79.8%	83.0%
trente	**20.2%**	**25.5%**	41.5%	10.6%	20.2%	**43.6%**	0.0%	0.0%	0.0%
quarante	47.2%	64.2%	**86.8%**	**52.8%**	**66.0%**	81.1%	45.2%	52.8%	54.7%
cinquante	45.7%	61.4%	**78.6%**	**54.3%**	**65.7%**	71.4%	22.8%	24.2%	30.0%
soixante	**62.8%**	**77.0%**	**92.9%**	39.8%	51.3%	67.3%	53.1%	61.9%	65.4%
cent	78.2%	89.7%	**98.6%**	**83.3%**	**89.9%**	94.3%	75.0%	86.8%	94.0%
mille	52.9%	58.8%	70.6%	47.1%	56.9%	71.6%	**58.8%**	**69.6%**	**76.4%**
et	17.9%	28.6%	32.1%	**46.9%**	**62.5%**	**96.9%**	28.1%	31.2%	37.5%
francs	77.9%	**91.6%**	**98.5%**	68.6%	77.4%	84.6%	**82.8%**	90.2%	94.7%
centimes	0.0%	0.0%	1.9%	0.0%	0.0%	1.9%	0.0%	0.0%	7.6%
TOTAL	**58.7%**	**71.0%**	**83.1%**	55.9%	67.3%	80.6%	57.88%	66.7%	76.37%

Table 6: Comparison of performances, bold face numbers indicate the best approach.