An activation-verification model for on-line texts recognition

Loïc Oudot, Lionel Prevost, Maurice Milgram
Laboratoire des Instruments et Systèmes d'Ile de France, BP 252
4 Place Jussieu, 75252 Paris Cedex 5
loic.oudot@lis.jussieu.fr

Abstract

The multiplication of handheld devices using the pen (electronic book, tablet PC, PDA, smart phone…) as a way of interaction, require an efficient recognition system in order to substitute both keyboard and mouse.

In this paper, we present a new writer-independent system dedicated to the automatic recognition of on-line texts. This system uses a very large French lexicon (200 000 words) which covers a vast field of application.

This recognition process is based on the activation-verification model proposed in perceptive psychology. A set of experts encodes the input signal and extract probabilistic informations at several levels of abstraction (geometrical and morphological). A neural expert generates a tree of segmentation hypotheses. This tree is explored by a probabilistic fusion expert that uses all the available informations (geometrical and lexical) in order to provide the best transcription of the input signal.

Keywords: Handwriting recognition, perceptive concept, data fusion, neural networks.

1. Introduction

Handwriting recognition is more than ever an important task. It is easy to observe the multiplication of pocket size consumer devices which use a pen as input modality (keyboardless and mouseless). Today, personal computers, PDA, electronic books and even cellular phones integrate a pen. Their success is growing up even if they always do not have an efficient system of handwriting recognition. To be user-friendly, the recognition rate must be sufficiently hight such that the user needs to make a minimal number of corrections to the recognized text. The pen must add an additional comfort compared to the traditional keyboard/mouse input modalities.

The problem lies in the lack of handwriting recognition algorithms. The currently consumer products based on handwriting recognition (*Calligrapher*, *Graffiti*…) are still too constraining since they impose a fixed style of writing and the user have to relearn to write. On the other hand, handwriting recognition systems developed in laboratory are able to recognize natural writing but do not obtain sufficient performances to be commercially affordable.

We present in this paper, an handwriting texts recognition system in a writer-independent frame. We choose to recognize script style to place the user on a intermediate level of constraint. We release the user to the writing style constraint while maintaining the segmentation constraint. This system is designed to be implemented in tablet PC style computer.

This paper is organized as follows. In section 2, we present the research progress in perceptive psychology and the handwriting recognition systems inspire on this results. In section 3, we describe in detail the reading engine. Finally, conclusions and future work are presented in the 4 section.

2. Perceptive modelling based Systems

To explain the process of reading, the cognitive psychology researchers compare the reader brain to an information processing system (IPS) which includes in its minimal version a processor, several memories and receptors [1].

The reading cognitive models proposed by the specialists do not only copy this IPS, but also try to explain the importance of the context and the contextual effects observed on humans (word superiority effect, priming effects…) during the visual words recognition task. The relevance of a reading cognitive model is precisely evaluated according to its aptitude to explain the various contextual effects.

The general triple ways model (semantics, phonological [3]) was not exploited yet by handwriting recognition system. Some systems like [4, 9] exploit the interactive-activation model of [6]. But, the majority of the read-

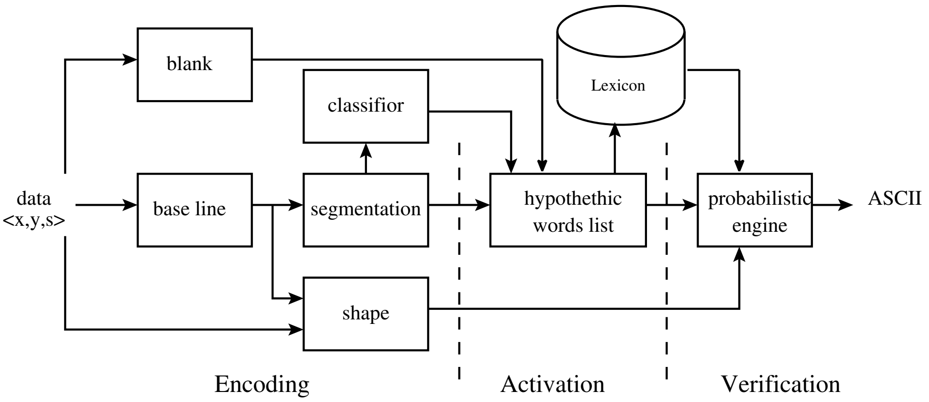


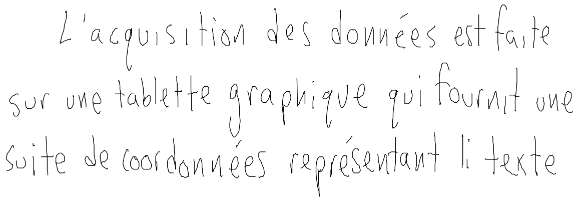
Figure 1. Writer-independent baseline system overview.

ing systems ([2]...), and our, are based on the activation-verification model introduced by Paap [8] in 1982.

In this model, the input visual stimulus activate a list of candidates words of the lexicon (same shape, close semantic, same sonority...). With the verification stage, this list of words is compared with the input signal in order to determine the best candidate. The probability of a word is the balanced sum of the probability based on a lexical evidence and that based on an alphabetical evidence.

3. Writer-independant baseline system

We collected a database of 90 texts written by 54 different writers regrouping 26 000 letters, 5 400 words and 3 000 punctuation marks (figure 2). This database was divided into two equal parts: a training database for the estimation of the system parameters (B_{Tr}) and a test database for the system evaluation (B_{Te}). Each database contains 45 texts. B_{Tr} contains 25 writers including 14 in common with B_{Te}. B_{Te} contains 29 writers and includes 15 completely unknown writers to B_{Tr}. We do not completely differentiating the writers of the two databases to evaluate the results and the relevance of the system in an omni-writer and multi-writer frame.



L'acquisition des données est faite
sur une tablette graphique qui fournit une
suite de coordonnées représentant li texte

Figure 2. Example of text in B_{Tr}.

We evaluate this system using two French lexicons of different size. The first D_{fr} contains 185 000 words and covers a very large fields of application while the second D_{rd}, contains the 8 000 most frequent words of the French language.

The Writer-independent baseline system overview is presented figure 1. The precept of the system is based on the activation-verification cognitive model. The baseline consist of a set of encoding experts who allow to extract geometrical and morphological informations of the input data. All of these experts provide probabilistic information to the stroke level. This local and global informations are merged in order to activate a list of hypothetical words in the lexicon. The meaningful of each hypothesis in the list is then evaluated by a probabilistic engine which finally, give the best probabilistic retranscription on the input data.

3.1. Encoding experts

3.1.1 Base lines detection

The most popular techniques of base lines detection use the image contour of the text or adjust parallel parabolic lines by an iterative process.

Our expert approximates the base lines with piece of straight lines. The expert select the strokes representing median letters (*a, c, e, i, n...*) to connect them by straight lines. The base line is obtained by connecting the bottom of the bounding boxes of this selected strokes and the body line is obtained by connecting the top of the bounding boxes.

3.1.2 Shape characterization

This expert determine the word shape, which represent global informations, by using the base lines informations [10]. This expert considers for each stroke s_i two probabilities corresponding to the presence of an ascender $p(a|s_i)$ and of a descender $p(d|s_i)$. With the help of this two informations, we assign to the stroke an activation class Act_i among the four following ones: *medium (m)*, *ascender (a)*, *descender (d)* and *ascender/descender(f)*.

The estimation of these probabilities use the base lines calculated by the preceding expert. The estimation of $p(m|s_i)$ and $p(f|s_i)$ is deduced from the two other probabilities:

- $p(m|s_i) = \frac{1 - \max(p(a|s_i), p(d|s_i))}{2}$
- $p(f|s_i) = \frac{p(a|s_i) + p(d|s_i)}{2}$

The activation class Act_i is determined by a neural network. The (2-3-4) MLP is trained on B_{Tr}. We obtain a good activation rate of 94,7 % on B_{Te}.

3.1.3 Blank detection

This expert provides for each inter-strokes separator (ISS) (blank separating two consecutive strokes) iss_i a probability $p(b|iss_i)$ to be a blank separating two distinct words. This information is essential for the localization of the words in a line of text.

A neural network minimizes the classification error probability starting from the inter-strokes horizontal distance informations.

We obtain a good classifications rate of 95.4 %. We note that the detector has tendency to under segmenting the words by merging consecutive words.

3.1.4 Segmentation

Inter-strokes separator (ISS) like strokes, contain a great quantity of information. As [5] we uses a neural network to classify these ISS. From the input data, we builds an undirected graph $G = (S, I)$ with nodes made up by the strokes $S = \{s_1, s_2, \ldots, s_n\}$ and edges $I = \{(s_1, s_2), (s_2, s_3), \ldots, (s_{n-1}, s_n)\}$ corresponding to the ISS. We distinguishes five classes of ISS: *intra-letter* (I_1), *inter-letters & intra-word* (I_2), *inter-words* (I_3), *diacritic* (I_4) and *punctuation* (I_5).

We extract 32 parameters to characterize an ISS. We try to find the more writers invariants parameters (omni-writers parameters). They are obtained from the coordinates of the two consecutive strokes, their bounding boxes, their positions compared to the bases lines and the pen up line. The pen up line is defined like the segment joining the last point of s_n to the first point of s_{n+1}.

Best classifier are obtained with a (32-20-5) MLP. The recognition rate is 92.3 % for the first answer (top_1) and 99.1 % for top_2 on B_{Te}.

3.1.5 Characters classifier

The characters classifier evaluates for each stroke the membership probabilities on the 62 classes (26 lower case, 26 upper case and 10 digit): $p(c|data)$ with $c \in \Gamma = \{a, \ldots, z, A, \ldots, Z, 0 \ldots, 9\}$.

The prototypes based classifier use a k-ppv algorithm. Unknown example are compared to the prototypes corpus of the corresponding class. We used the Dynamic Time Warping distance (DTW), largely used in dynamic characters recognition [12] to obtain the probabilities with the help of the softmax rule.

Prototypes database	top_1	top_2
UD	78.9 %	80.2 %
OWD	84.8 %	87.3 %
MWD	88.7 %	90.5 %

Table 1. Characters classifier Recognition rate on B_{Te} vs. prototypes database.

The prototypes database of the classifier is obtained by a clustering algorithm named SMAC [11]. A first prototypes database (UD) extracted from UNIPEN gives disappointing results (table 1). To improve these results, two other databases were created starting from the characters composing B_{Tr}: An omni-writer data-base (OWD), extracted from the texts written by the users who do not appear in the B_{Te} and a multi-writer database (MWD) extracted from all the texts of the B_{Tr}.

3.2. Activation-verification step

3.2.1 Segmentation tree generation

Classical technique to find the segmentation of a text, is to generate a hypothesis segmentation tree [10] and let the decision to the following modules (in general the characters classifier and/or the lexicon). The difficulty of this technique is to find a good compromise between the number of generated hypothesis and the recognition speed. Generating many hypothesis makes it possible to increase the probability of obtaining the good segmentation but risk to disturb and slow down the system by a too great number of hypothesis. It is thus necessary to generate a minimum of the most relevant segmentation hypothesis.

If, for the segmentation tree generation, we kept the two best results (top_2) of the segmentor, we reach 99.1 % of appearance of the good segmentation. Such a segmentation tree has approximately 10^{12} hypothesis for a normal text line of 40 strokes and is thus unusable. Point the fact, that the appearance rate of good segmentation corresponds to the existence of the real segmentation in the tree and not to the good segmentation rate.

To reduce the number of segmentation hypothesis, we use the blank detector (class I_3) to fix anchoring points in the line. This points cut out the segmentation tree in several smaller tree and break up the combinational. By considering the ISS iss_i having a probability $p(b|iss_i) = 1$ as an anchoring point, we detect 50 % of the blank with less than 1 % of false detections.

The segmentation tree (figure 3) is generated gradually by adding the class of the most probable ISS until reaching N_{hypo} segmentation hypothesis. The relevance of a class for a given ISS is calculated by combining the *a priori* and the *a posteriori* probabilities of the segmentor:

$$p(I_i|data) = \sum_{a=I_1}^{I_5} p(I_i|a)p(a|data)$$

With $p(a|data)$ the *a posteriori* probability resulting from the segmentor and $p(I_i|a)$ the *a priori* probability deduced from the confusion matrix of the segmentor. This matrix is estimated on B_{Tr} and summarize the segmentor behavior observed on this database.

With this construction, we can keep until top_5 of the segmentor for the ambiguous ISS and only top_1 for sure ones. This method gives very good results: with a tree limited to $N_{hypo} = 500$ hypothesis, we obtains 98.4 % of good appearance of the segmentation and for $N_{hypo} = 2\ 000$ more than 98.8 %.

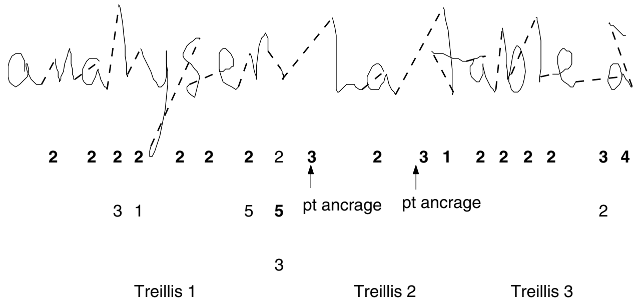


Figure 3. Example of segmentation tree with anchor points (26 hypothesis).

3.2.2 Hypothetic words activation

The activation of hypothetic words in the lexicon is introduced by a lexical search into a restricted part of the principal lexicon. Since search is carried out only on the words with the same length, the organization of the lexicon being based on the number of characters composing the words.

First, we construct a seed word by concatenating each letter hypothesis (the most probable class of the characters classifier having the same shape observed by the shape characterizer). A lexical search starting from this seed word with a N_{err} tolerance characters draws up the list of the hypothetic words. The threshold $N_{err} = \mathrm{E}(N_{car}/2)$ was chooses to have the best compromise between the number of generated words and the appearance of the good word in this list. We obtain 97.7 % of good appearance of the real word on B_{Te} with an average of 50 hypothetical words. The 2.3 % of error are due to too many errors of classification, especially on the short words where more of the half of the letters are false.

3.2.3 Verification & text retranscription

The text line retranscription takes place into two stages. The first stages (verification) assigns for each hypothetical word a meaningful probability and construct a words tree. The second stages carries out a search in this words tree to find the most probable text line.

The meaningful probability of a word $p(word|data)$ is the product of the probability of its characters $p(char|data)$ by the probability of its segmentation $p(seg|data)$.

Probability of all the characters $p(ch|data)$ composing a word is deduce by the probabilities of each character $p(char^i|data)$ estimated from the input data: $p(ch|data) = \prod_{i=1}^{N_{ch}} p(ch^i|data)$.

The estimation of the characters probability takes into account the *a posteriori* probabilities of the characters classifier and the shape characterizer and the *a priori* probabilities deduced from the confusion matrix of these two experts.

$$p(ch^i|data) = \sum_{(c\in\Gamma, s\in\{m,h,j,f\})} p(ch^i|(c,s))p((c,s)|data)$$

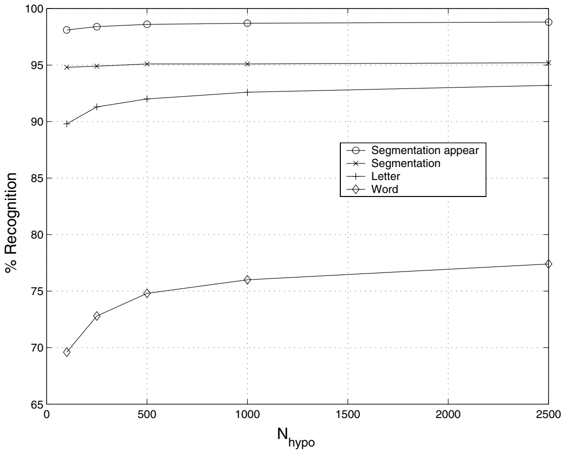
The $p((c,s)|data)$ term corresponds to the probability of observing the character c having the shape s knowing the input data. This information is given by two experts: characters classifier and shape characterizer. If we consider independence of the two sources, we have: $p((c,s)|data) = p(c|data)p(s|data)$. The $p(char^i|(c,s))$ term is deduced from the confusion matrix.

The probability of the segmentation is deduced from the probability previously calculated during the tree generation (fusion of the *a priori* and *a posteriori* probabilities of the segmentor): $p(seg|data) = \prod_{i=1}^{N_{char}} p(iss_i|data)$

Each word of each segmentation hypothesis of a line has an meaningful probability and a position within the line (beginning and end in number of stroke) creating a words hypothesis tree. By using a dynamic programming algorithm we finds the most probable path inside the words hypothesis tree. The power of this algorithm lies in the global character of the search: the most probable path found is a combination of several segmentation hypothesis (table 2).

```
d l s d.o u.n.e l s e.s.t t.u.e t.e
d.e       u.n.e.s  /    \f.a.i.t.e
d.e.s d.o.n.n.e.e /
     \
  e.s d.o.n.n.e.e.s
          h.u.e.e.s e.x.t.r.a.i.r.e
                e.s ?.?.?.?.?.?
```

Table 2. Words hypothesis tree example (dots are intra-word space and blanks represent inter-words space).



	D_{fr}	D_{rd}
OWD		
word	71.9 %	74.8 %
letter	91.6 %	92.0 %
MWD		
word	74.8 %	77.5 %
letter	93.2 %	93.4 %

Table 3. Words and letters recognition rate.

4 Results

The words and letters recognition rate obtained for $N_{hypo} = 500$ are given in table 3. Utilization of a large lexicon reduce the words recognition rate of only 3 %. The improvement of the recognition rate by the use of a prototypes database closer to the writer (MWD) shows the interest of an writer adaptation of this system. Moreover, the results can be still improved by considering a greater number of hypothesis N_{hypo}.

5. Conclusions & futur works

A complete end-to-end system for reading on-line texts has been implemented, based on the activation-verification perceptive model. This system is writer-inde-pendent and is able to manage very large lexicons while preserving high recognition rates. This system has only one threshold N_{hypo} which makes it possible to fix the maximum depth of the segmentation tree and thus to manage the *computing time* vs. *recognition rate*. We can choose the optimum recognition rate according to the computing power available.

The segmentation technic, with an average of 95 % of good segmentation, has to be further improved. Most of the segmentation errors are false detections of blank between words which decrease the word recognition rate. These false detections mainly appear with writers not accustomed to handprinted writing which separate equally words and subwords, and lead to blank detector failures. On the other hand, knowing the segmentation of the text in words increases the recognition rate (92 %) on words larger than two letters.

The recognition rate increases in the multi-writer framework. Recently, we tried to show that adapting the system to a given writer (writer-dependent framework) could improve drastically its accuracy. The modular architecture of the character classifier and its independent prototype database, is perfectly fitted to take into account new prototypes. Our last works on adaptation combine supervised and self-supervised strategies [7] to reach a high word recognition rate (higher than 90 %).

References

[1] T. Baccino and P. Colé. *La lecture experte*. Presse universitaire de France, 1995.

[2] S. Carbonnel and E. Anquetil. Modélisation et intégration de connaissance lexicales pour le post-traitement de l'écriture manuscrite en-ligne. *RFIA*, 3:1313–1322, 2004.

[3] M. Coltheart. *Lexical access in simple reading task*. Underwood, Strategies of information processing, Academic Press, 1978.

[4] M. Côté. *Utilisation d'un modèle d'accès lexical et de concepts perceptifs pour la reconnaissance d'images de mots cursifs*. PhD thesis, ENST, 1997.

[5] P. D. Gader, M. Mohamed, and J.-H. Chiang. Handwritten word recognition with character and inter-character neural networks. *IEEE Transactions on Systems, Man, and Cybernetics*, 27(1):158–164, 1997.

[6] J. L. McClelland and D. E. Rumelhart. An interactive activation model of context effects in letter perception. *Psychological Review*, 88:375–407, 1981.

[7] L. Oudot, L. Prevost, and M. Milgram. Self-supervised adaptation for on-line text recognition. *IWFHR*, Submited, 2004.

[8] K. Paap, S. L. Newsome, J. E. McDonald, and R. W. Schvaneveldt. An activation-verification model for letter and word recognition: The word superiority effect. *Psychological Review*, 89:573–594, 1982.

[9] L. Pasquer. *Conception d'un modèle d'interprétation multi-contextuelle, application à la reconnaissance en-ligne*

d'écriture manuscrite. PhD thesis, Université Rennes 1, 2000.

[10] R. K. Powalka, N. Sherkat, and R. J. Whitrow. Word shape analysis for a hybrid recognition system. *Pattern Recognition*, 30(3):421–445, 1997.

[11] L. Prevost and M. Milgram. Modelizing character allographs in omni-scriptor frame: a new non-supervised algorithm. *Pattern Recognition Letters*, 21(4):295–302, 2000.

[12] V. Vuori, J. Laaksonen, and E. Oja. On-line adaptation in recognition of handwritten alphanumeric charaters. *ICDAR*, 1:792–795, 1999.