Un modèle d'activation vérification pour la lecture de textes manuscrits dynamiques

Loïc Oudot, Lionel Prevost, Maurice Milgram

Un modèle d'activation vérification pour la lecture de textes manuscrits dynamiques

Loïc Oudot – Lionel Prevost – Maurice Milgram

Laboratoire des Instruments et Systèmes d'Ile de France
4 Place Jussieu
75252 Paris Cedex 5
loic.oudot@lis.jussieu.fr

Résumé : *La multiplication des appareils électroniques (livre électronique, tablet pc, PDA, smart phone…) qui utilisent un stylo comme moyen d'interaction nécessitent un système de lecture de l'écriture manuscrite fiable pour pouvoir substituer le stylo au clavier et à la souris.*
Nous présentons dans cet article un nouveau système de lecture automatique de textes dynamiques omni-utilisateur. Ce système utilise un lexique de très grande taille (200 000 mots) qui couvre un très large champs d'applications.
Ce moteur est basé sur le modèle cognitif d'activation-vérification proposé en psychologie perceptive. Un ensemble d'experts encode le signal d'entrée et extrait des informations probabilistes à des niveaux d'abstraction différents (géométrique et morphologique). Un expert de segmentation neuronal génère un treillis d'hypothèse de segmentation. Ce dernier est exploré par un moteur de fusion probabiliste qui utilise les informations issues des experts d'encodage et du lexique pour fournir la meilleure transcription du signal d'entrée.

Mots-clés : Lecture automatique, écriture dynamique scripte, concepts perceptifs, fusion d'informations

1 Introduction

La reconnaissance automatique de l'écriture est aujourd'hui plus que jamais un enjeu important. Il suffit d'observer la multiplication des appareils électroniques qui utilisent un stylo comme moyen d'interaction avec la machine pour s'en convaincre. Les ordinateurs, les assistants personnels, les livres électroniques et même les téléphones portables intègrent tous aujourd'hui un stylo. Leur succès est grandissant même s'ils ne possèdent pas toujours un système de reconnaissance de l'écriture efficace. Pour pouvoir être adopté comme moyen d'acquisition, le stylo doit amener un confort au moins équivalent voire supplémentaire par rapport au traditionnel couple clavier/souris.
Le problème réside dans l'absence d'algorithme de lecture automatique de l'écrit. Les systèmes de reconnaissance d'écriture actuellement commercialisés (*Calligrapher*, *Handwriting Recognition Microsoft*…) sont encore trop contraignants dès lors qu'ils imposent un style d'écriture fixe obligeant l'utilisateur à ré-apprendre à écrire. A l'opposé, les systèmes développés en laboratoire, capables de reconnaître l'écriture naturelle, n'obtiennent pas de performances suffisantes pour envisager une commercialisation.
Nous présentons dans cet article un moteur de lecture automatique de textes scripts omni-scripteur. En choisissant de reconnaître l'écriture scripte, nous nous positionnons à un niveau de contrainte intermédiaire puisque nous libérons l'utilisateur de la contrainte sur le style d'écriture tout en maintenant une contrainte sur la segmentation.
Dans la deuxième section nous présenterons l'état d'avancement des recherches en psychologie perceptive et des moteurs de lecture s'en inspirant. La troisième section décrit en détail le fonctionnement du moteur de lecture. Les conclusions et perspectives sont présentées dans la quatrième et dernière partie.

2 Systèmes basés sur une modélisation perceptive de la lecture

Pour expliquer le déroulement du processus de lecture, les chercheurs en psychologie cognitive apparentent le cerveau du lecteur à un système informatique de traitement de l'information qui comprend dans sa version minimale un processeur, plusieurs mémoires, des organes récepteurs et d'éventuelles effecteurs [BAC 95].
Les modèles cognitifs de lecture développés par les spécialistes ne cherchent pas seulement à modéliser ce traitement de l'information, mais tentent également de modéliser l'importance du contexte et des effets contextuels observés chez l'Homme (effet de supériorité du mot, effet d'amorçage… [TAF 91]) dans la reconnaissance visuelle des mots. En effet, la pertinence d'un modèle cognitif de lecture est précisément évaluée en fonction de son aptitude à rendre compte des différents effets contextuels.
Le modèle général à triple voies (sémantique, phonologiques [COL 78]) n'a pas encore été exploité. Quelques systèmes [CÔT 97, PAS 00] exploitent le modèle d'activation interactive de [MCC 81]. La plupart des systèmes ([CAR 04, CON 00]…), dont le notre, se basent sur le modèle d'activation vérification proposé par Paap [PAA 82].
Dans ce modèle, le stimulus visuel d'entrée déclenche l'activation de certains mots du lexique (silhouette identique, sémantique proche, même sonorité…). Les mots ainsi activés constituent une liste de mots candidats. A l'étape de vérification, cette liste de mots est confrontée aux informations données en entrée (le stimulus de départ) afin de déterminer le meilleur candidat. La probabilité d'une réponse est la

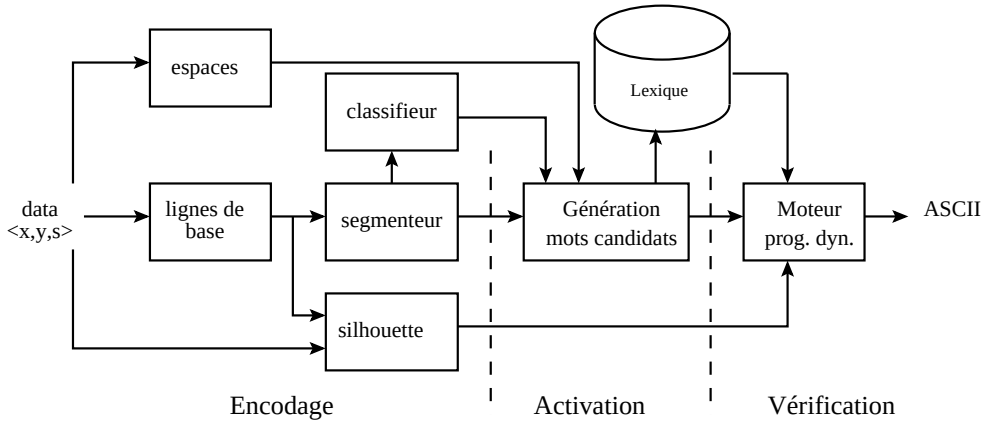


FIG. 1 – Structure générale du système de lecture.

somme pondérée de la probabilité de réponse correcte basée sur une évidence lexicale et celle basée sur une évidence alphabétique.

3 Système de lecture

Nous avons collecté une base de 90 textes écrits par 54 scripteurs différents réunissant 26 000 lettres, 5 400 mots et 3 000 signes de ponctuation (figure 2). Cette base a été divisée en deux parties égales : une première base pour l'estimation des paramètres du système de lecture (BA_{texte}) et une seconde pour son évaluation (BT_{texte}). Chaque base contient 45 textes. La base d'apprentissage comprend 25 scripteurs dont 14 en commun avec la base de test. La base de test quand à elle, contient 29 scripteurs et inclut donc 15 scripteurs totalement inconnus de la base d'apprentissage. Le fait de ne pas totalement différencier les scripteurs entre les deux bases permet de mieux évaluer les résultats et la pertinence du système et de comparer simultanément son comportement dans les cadres omni-scripteur et multi-scripteur.

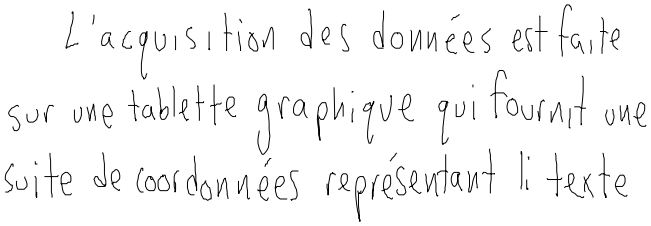


FIG. 2 – Exemple de texte de BA_{texte}.

Nous avons évalué ce système à l'aide de deux lexiques français de taille différente. Le premier $Dico_{fr}$ contient 185 000 mots et couvre un très large champs d'application tandis que le second $Dico_{rd}$, contient les 8 000 mots les plus fréquents de la langue française.

La structure générale du moteur de lecture (figure 1) s'inspire du modèle d'activation vérification. Le système se présente sous la forme d'une série d'experts d'encodage qui permettent d'extraire les informations de type géométrique et morphologique du texte d'entrée. Ces experts fournissent des informations probabilistes au niveau *stroke*. Un premier expert de pré-traitement, non représenté sur la figure, segmente le flux d'entrée en ligne car notre système traite les informations ligne de texte par ligne de texte. Toutes ces informations, aussi bien au niveau local qu'au niveau global permettent d'activer une liste de mots du lexique. La cohérence de chaque mot hypothèse de la liste est ensuite évaluée par un moteur probabiliste qui valide la meilleure hypothèse pour la retranscription.

3.1 Experts d'encodage

3.1.1 Détection des sauts de ligne

Cet expert recherche les sauts de lignes pour segmenter le flux de données d'entrée en ligne. Les techniques les plus courantes utilisent le contour du texte [SEN 98] ou ajustent des lignes paraboliques parallèles par un processus itératif [CAE 93].

Notre expert approxime les lignes d'assises par morceau avec des lignes droites. Pour cela, il recherche les strokes représentant des lettres médianes (*a, c, e, i, n...*) pour les relier ensuite par des lignes droites. On relie le bas des boîtes englobantes de ces derniers pour obtenir la ligne de base et le haut des boîtes pour obtenir la ligne de corps.

3.1.2 Caractérisation de la silhouette

Le rôle de cet expert est de déterminer la forme ou la silhouette des mots en utilisant les lignes de bases et d'extraire ainsi des informations globales [POW 97]. Cet expert estime pour chaque stroke s_i deux probabilités correspondant à la présence d'une hampe $p(h|s_i)$ et d'un jambage $p(j|s_i)$. De plus, à partir de ces deux informations, on attribue au stroke une classe d'activation Act_i parmi les quatre suivantes : *médiane (m), hampe (h), jambage (j)* et *hampe/jambage (f)*.

L'estimation de ces probabilités est basée sur l'utilisation des lignes d'assises calculées par l'expert précédent. L'estimation de $p(m|s_i)$ et $p(f|s_i)$ est déduite des deux probabilités précédentes :

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

La classe d'activation Act_i est déterminée par un réseau de neurones de type MLP (2-3-4) entraîné sur la base d'apprentissage. Nous obtenons un taux de bonne activation de 94,7 % sur la base de test.

3.1.3 Détection des espaces

Cet expert fournit pour chaque arc inter-strokes a_i une probabilité $p(e|a_i)$ d'être un espace séparant deux mots distincts. Cette information est précieuse pour la localisation des mots dans un ligne de texte.

Un réseau de neurones minimise la probabilité d'erreur de classification à partir de la distance horizontale inter-strokes. De manière à avoir une probabilité en sortie de ce réseau, la cellule de sortie utilise une fonction de transfert linéaire saturée entre 0 et 1. Le taux de bonnes classifications obtenu est de 95.4 %. On constate que le détecteur a tendance à sous segmenter les mots en fusionnant les mots consécutifs.

3.1.4 Segmenteur

Les levés de stylo tout comme les *strokes*, contiennent une grande quantité d'informations. Comme [GAD 97] on utilise un réseau de neurones pour classifier ces arcs inter-strokes. A partir des données d'entrée, on construit un graphe non orienté $G = (S, A)$ avec un ensemble de sommets constitués des strokes $S = \{s_1, s_2, \ldots, s_n\}$ et des arêtes $A = \{(s_1, s_2), (s_2, s_3), \ldots, (s_{n-1}, s_n)\}$ correspondant aux arcs reliant deux strokes consécutifs du point de vue temporel. On distingue cinq classes d'arcs : *intra-lettre* (A_1), *inter-lettres & intra-mot* (A_2), *inter-mots* (A_3), *diacritique* (A_4) et *ponctuation* (A_5).

32 paramètres servent à caractériser les arcs inter-strokes. Ils sont issus des coordonnées de deux strokes consécutifs s_n et s_{n+1}, de leurs boîtes englobantes, de leurs positions par rapport aux lignes de bases et du lever de stylo. Le lever de stylo est défini comme le segment joignant le dernier point de s_n au premier point de s_{n+1}. L'idéal étant de trouver des paramètres invariants aux scripteurs (paramètres omni-scripteurs).

Le meilleur classifieur obtenu est un réseau MLP (32-20-5). Le taux de reconnaissance est de 92.3 % pour la première réponse (top_1) et de 99.1 % pour le top_2 sur la base de test.

3.1.5 Classifieur de caractères

Le classifieur de caractères évalue pour chaque stroke les probabilités d'appartenance aux 62 classes (26 minuscules, 26 majuscules et 10 chiffres) : $p(c|data)$ $\Gamma = \{A, \ldots, Z, a, \ldots, z, 0 \ldots, 9\}$.

Le classifieur à base de prototypes [PRE 98] est de type 1-ppv. L'exemple inconnu sera comparé au corpus de prototypes de la classe correspondante. La métrique utilisée est la distance élastique, largement utilisée en reconnaissance de caractères dynamiques [VUO 99]. Les probabilités sont obtenues par l'utilisation de la règle du *softmax*.

Les bases de prototypes du classifieur sont obtenus par un algorithme de clustering nommé SMAC [PRE 00]. Une première base de prototypes extraite de la base UNIPEN donne des résultats décevants (tableau 1). Pour améliorer ces résultats, deux autres bases ont été crées à partir de la base de texte d'apprentissage BA_{texte} : une première, omni-scripteur, extraite des textes écrits par les scripteurs qui n'apparaissent pas dans la base de test et la seconde extraite de tous les textes de la base d'apprentissage.

Base de prototypes	top_1	top_2
UNIPEN	78.9 %	80.2 %
Omni-scripteur	84.8 %	87.3 %
Multi-scripteur	88.7 %	90.5 %

TAB. 1 – Taux de reconnaissance du classifieur dynamique sur la base de textes en fonction de la base de prototypes.

3.2 Moteur de lecture

3.2.1 Génération du treillis de segmentation

Une technique classique pour retrouver la segmentation d'un texte, est de générer un treillis d'hypothèses de segmentation [POW 97, HEN 01] et de laisser aux modules suivants (en général le classifieur de caractère et/ou le lexique) la prise de décision. L'inconvenient avec cette technique est de trouver un bon compromis entre le nombre d'hypothèses générées et la vitesse de traitement. Générer beaucoup d'hypothèses permet d'augmenter la probabilité d'obtenir la bonne segmentation mais risque de perturber et ralentir le système par un trop grand nombre d'hypothèses. Il faut donc réussir à générer un minimum d'hypothèses les plus pertinentes possibles.

Générer simplement le treillis en ne conservant que les deux meilleurs résultats (top_2) du segmenteur permet d'atteindre 99.1 % d'apparition de la bonne segmentation. Un tel treillis possède environ 10^{12} hypothèses pour une ligne moyenne de 40 *strokes* et est donc inutilisable. Rappelons que le taux d'apparition correspond à l'existence de la segmentation réelle dans le treillis et non au taux de bonne segmentation.

Pour réduire le nombre d'hypothèses de segmentation, nous utilisons le détecteur d'espaces inter-mots (classe A_3) pour fixer des points d'ancrage dans la ligne. Ces derniers découpent le treillis de segmentation en plusieurs sous-treillis plus petits et cassent la combinatoire. En considérant les arcs ayant une probabilité $p_{espace} = 1$ comme étant des point d'ancrage, on retrouve 50 % des espaces inter-mots avec moins de 1 % de fausses détections.

Le treillis de segmentation est généré progressivement en ajoutant les types d'arcs les plus probables jusqu'à atteindre N_{hypo} hypothèses de segmentation. La pertinence d'une classe d'arc pour un arc donné est calculée en fusionnant les probabilités *a priori* et *a posteriori* du segmenteur :

$$p(A_i|data) = \sum_{a=A_1}^{A_5} p(A_i|a)p(a|data)$$

Avec $p(a|data)$ la probabilité *a posteriori* issue du segmenteur et $p(A_i|a)$ la probabilité *a priori* déduite de la matrice de confusion de ce dernier. Cette matrice est estimée sur la base d'apprentissage BA_{texte} et synthétise le comportement du segmenteur observé sur cette base. Si la base d'apprentissage est statistiquement représentative, il y a de fortes probabilités que ce comportement soit le même dans les autres cas.

Cette méthode donne de très bons résultats : avec un treillis limité à $N_{hypo} = 500$ hypothèses, on obtient 98.4 % de bonne apparition de la segmentation et pour $N_{hypo} = 2\,000$ plus de 98.8 %.

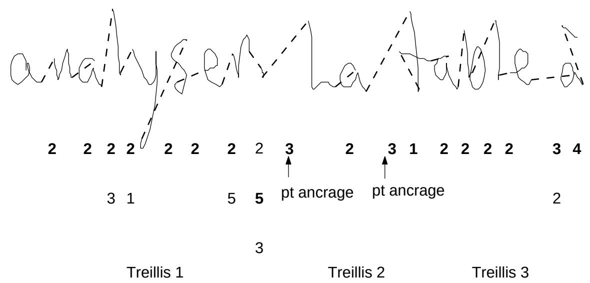


FIG. 3 – Exemple de treillis de segmentation avec points d'ancrages (26 hypothèses).

3.2.2 Activation des mots candidats

L'activation des mots candidats du lexique (modèle d'activation vérification) est introduit par une recherche lexicale dans une sous partie restreinte du lexique principal. En effet, la recherche ne s'effectue que sur les mots de même longueur, l'organisation du lexique étant basée sur le nombre de caractères des mots.
Les mots hypothèses sont obtenus en conservant pour chaque lettre hypothèse, la classe la plus probable du classifieur de caractères et possédant la même silhouette observée par le détecteur de silhouette. Une recherche lexicale en tolérant N_{err} caractères de différence constitue la liste des mots lexicalement corrects. Le seuil $N_{err} = \mathrm{E}(N_{car}/2)$ a été choisit de manière à avoir le meilleur compromis entre le nombre de mots générés et l'apparition du bon mot dans cette liste. Nous obtenons 97.7 % de bonne apparition du mot réel sur la base de test avec une moyenne de 50 mots candidats par proposition. Les 2.3 % d'erreur sont dus à de trop nombreuses erreurs de classification, surtout sur les mots courts où plus de la moitié des lettres sont fausses.

3.2.3 Retranscription du texte

La retranscription d'une ligne de texte se décompose en deux étapes. La première assigne à chaque mot hypothèse une probabilité de cohérence et la seconde effectue une recherche de la ligne la plus probable.
La probabilité d'un mot de la liste $p(mot|data)$ est le produit de la probabilité des caractères le composant $p(car|data)$ par la probabilité de sa segmentation $p(seg|data)$.
La probabilité de tous les caractères $p(car|data)$ composant un mot revient à estimer les probabilités de chaque caractère $p(car^i|data)$ à partir des données d'entrée : $p(car|data) = \prod_{i=1}^{N_{car}} p(car^i|data)$.
L'estimation de la probabilité des caractères tient compte des probabilités *a posteriori* du classifieur de caractères ainsi que du détecteur de silhouette et des probabilités *a priori* déduites de la matrice de confusion des ces deux experts.

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

Le terme $p((c,s)|data)$ correspond à la probabilité d'observer le caractère c ayant la silhouette s connaissant les données d'entrée. Ces informations sont données par deux experts : classifieur de caractères et caractérisation de la silhouette. Si nous considérons les deux sources indépendantes, alors : $p((c,s)|data) = p(c|data)p(s|data)$. Le terme $p(car^i|(c,s))$ est déduit de la matrice de confusion.
La probabilité de la segmentation est déduite des probabilité calculées précédemment lors de la génération du treillis (fusion des probabilités *a priori* et *a posteriori* du segmenteur) : $p(seg|data) = \prod_{i=1}^{N_{car}} p(s|a_i)$

```
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 ?.?.?.?.?.?
```

TAB. 2 – Exemple de treillis de mots (les points correspondent à des espaces intra-mots, les blancs à des espaces inter-mots).

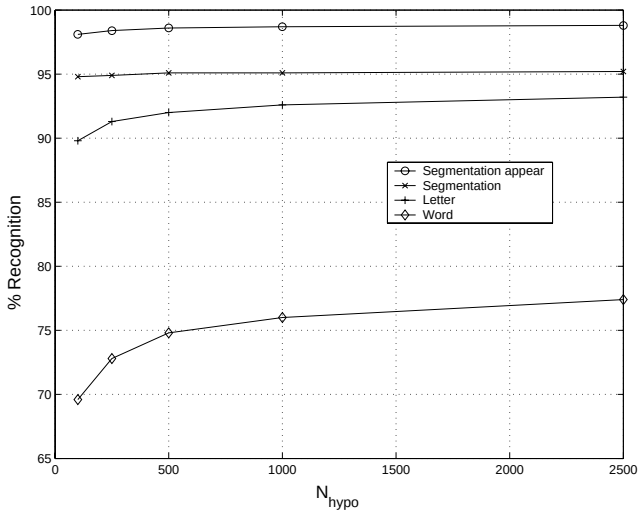
Chaque mot de chaque hypothèse de segmentation d'une ligne a donc une probabilité associée et une position de début et de fin (en numéro de *stroke*) dans la ligne. Ceci permet de générer un treillis de mots hypothèses. En utilisant un algorithme de programmation dynamique on retrouve le chemin le plus probable à l'intérieur du treillis de mots hypothèses. La force de cet algorithme réside dans le caractère global de la recherche. Le chemin le plus probable retrouvé est une combinaison de plusieurs hypothèses de segmentation (tableau 2).

4 Résultats

Les taux de reconnaissance en mots et en lettres obtenus pour $N_{hypo} = 500$ sont donnés dans le tableau 4.
On constate que l'utilisation d'un grand lexique ne fait chuter le taux de reconnaissance en mots que d'environ 3 %. L'amélioration du taux de reconnaissance par l'utilisation d'une base de prototypes plus proche du scripteur montre l'intérêt d'une adaptation du moteur de lecture. De plus, les résultats peuvent être encore améliorés en considérant un plus grand nombre d'hypothèses N_{hypo}.

5 Conclusions et perspectives

Nous venons de présenter un système de lecture de textes basé sur le modèle perceptif d'activation vérification. Ce système fonctionne dans un cadre omni-scripteur et est en mesure d'utiliser des lexiques de grande taille en conservant des taux de reconnaissance élevés. Ce moteur possède de plus un seuil N_t qui permet de fixer la profondeur maximum du treillis de segmentation et ainsi de gérer le compromis *temps de calcul* / *taux de reconnaissance*. On peut donc choisir le taux de reconnaissance optimum suivant la puissance de calcul disponible.
Notre méthode de génération / validation de la segmentation donne également de très bons résultats avec une moyenne de 95 % de bonne segmentation. De plus, la majorité des erreurs de segmentation sont de fausses détections d'espace inter-mots ce qui entraîne inévitablement une chute du taux de reconnaissance en mots. Les mauvaises détections apparaissent pour des scripteurs peu habitués à l'écriture scripte



Base de proto.	$Dico_{fr}$	$Dico_{rd}$
Omni-scripteur		
mot	71.9 %	74.8 %
lettre	91.6 %	92.0 %
Multi-scripteur		
mot	74.8 %	77.5 %
lettre	93.2 %	93.4 %

FIG. 4 – Taux de reconnaissance en mots et en lettres du moteur de lecture.

et qui séparent les lettres d'un même mot autant que les mots entre eux, rendant inefficace le détecteur d'espace. Lorsque la segmentation en mots est connue, le système de lecture de mots voit son taux de reconnaissance augmenter, dépassant 92 % sur les mots de trois lettres et plus.

La très nette augmentation du taux de reconnaissance obtenue sur la base multi-scripteur, démontre l'avantage de s'orienter vers un moteur de reconnaissance multi-scripteur voire même mono-scripteur. L'architecture modulaire du classifieur de caractères et sa base de prototypes indépendante, le rendent parfaitement adapté pour la prise en compte de nouveaux prototypes. Les derniers travaux sur l'adaptation [OUD 04] ont permis d'atteindre un taux de reconnaissance de plus de 90 %.

Références

[BAC 95] BACCINO T., COLÉ P., *La lecture experte*, Presse universitaire de France, 1995.

[CAE 93] CAESAR T., GLOGER J. M., MANDLER E., Preprocessing and Features Extraction for a Handriting recognition System, *ICDAR*, vol. 1, 1993, pp. 408–411.

[CAR 04] CARBONNEL S., ANQUETIL E., Modélisation et intégration de connaissance lexicales pour le post-traitement de l'écriture manuscrite en-ligne, *RFIA*, vol. 3, 2004, pp. 1313–1322.

[COL 78] COLTHEART M., *Lexical access in simple reading task*, Underwood, Strategies of information processing, Academic Press, 1978.

[CON 00] CONNEL S. D., On-line Handwriting Recognition Using Multiple Pattern Class Models, PhD thesis, Department of Computer Science and enginnering, Michigan State University, 2000.

[CÔT 97] CÔTÉ M., Utilisation d'un modèle d'accès lexical et de concepts perceptifs pour la reconnaissance d'images de mots cursifs, PhD thesis, ENST, 1997.

[GAD 97] GADER P. D., MOHAMED M., CHIANG J.-H., Handwritten Word recognition with Character and Inter-Character Neural Networks, *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 27, n° 1, 1997, pp. 158–164.

[HEN 01] HENNING A., SHERKAT N., Cursive script recognition using wildcards and multiple experts, *Pattern Analysis & Applications*, vol. 4, 2001, pp. 51–60.

[MCC 81] MCCLELLAND J. L., RUMELHART D. E., An interactive activation model of context effects in letter perception, *Psychological Review*, vol. 88, 1981, pp. 375–407.

[OUD 04] OUDOT L., PREVOST L., MOISES A., Techniques d'adaptation au scripteur pour la lecture de textes manuscrits dynamiques, *CIFED*, vol. Soumis, 2004.

[PAA 82] PAAP K., NEWSOME S. L., MCDONALD J. E., SCHVANEVELDT R. W., An activation-verification model for letter and word recognition : The word superiority effect, *Psychological Review*, vol. 89, 1982, pp. 573–594.

[PAS 00] PASQUER L., Conception d'un modèle d'interprétation multi-contextuelle, application à la reconnaissance en-ligne d'écriture manuscrite, PhD thesis, Université Rennes 1, 2000.

[POW 97] POWALKA R. K., SHERKAT N., WHITROW R. J., Word Shape Analysis for a Hybrid Recognition System, *Pattern Recognition*, vol. 30, n° 3, 1997, pp. 421–445.

[PRE 98] PREVOST L., Reconnaissance de l'Ecrit Dynamique : Application à l'Analyse des Expressions Mathématiques, PhD thesis, Université Paris VI, 1998.

[PRE 00] PREVOST L., MILGRAM M., Modelizing character allographs in omni-scriptor frame : a new non-supervised algorithm, *Pattern Recognition Letters*, vol. 21, n° 4, 2000, pp. 295–302.

[SEN 98] SENIOR A. W., ROBINSON A. J., An Off-Line Cursive Handwriting Recognition System, *PAMI*, vol. 20, n° 3, 1998, pp. 309–321.

[TAF 91] TAFT M., *Reading and mental lexicon*, Erlbaum Edition, 1991.

[VUO 99] VUORI V., LAAKSONEN J., OJA E., On-line Adaptation in Recognition of Handwritten Alphanumeric Charaters, *ICDAR*, vol. 1, 1999, pp. 792–795.