

Reconnaissance de l'Écrit Dynamique

Application à l'Analyse de Texte

Loïc OUDOT, Lionel PREVOST, Maurice MILGRAM

Université Paris VI, LISIF / PARC, Case 164

4 Place Jussieu, 75252 Paris Cedex 5, France

loic.oudot@lis.jussieu.fr, lionel.prevost@lis.jussieu.fr, maum@ccr.jussieu.fr

Résumé - Nous présentons dans cette publication un nouveau système dédié à la reconnaissance de l'écriture dynamique. Il apparaît comme une alternative entre les assistants personnels (*Personal Digital Assistants*) du commerce qui sont très contraignants pour l'utilisateur et les moteurs de reconnaissance automatique de l'écriture cursive développés en laboratoire. Ces derniers n'ont pas encore atteint un niveau de performances commercialisable. Le traitement complet, de l'acquisition à la lecture, est associé à une interface utilisateur. Les résultats obtenus sur une base de textes omni-scripteur sont très encourageants. Ils peuvent être améliorés par l'utilisation d'un nouvel expert lexical qui adapte la reconnaissance au scripteur.

Abstract - We present in this contribution a new system dedicated to the analysis of hand-printed dynamic text. It appears as an alternative to merchandized personal digital assistants (PDAs) which are very user restricting and automatic cursive word recognizers that have been developed in laboratories and have still not reached a marketable target in spite of their accuracy. The whole treatment process, from the acquisition to the efficient reading, is integrated in a user-friendly interface. The results on an omni-writer text database are very encouraging. They should be improved by using a new lexicon driven expert that adapts the recognition to the writer.

1. Introduction

Nous détaillons ici un système de reconnaissance de l'écrit dynamique, dédié à l'analyse de textes scripts qui peuvent contenir plusieurs lignes de texte structurées. L'écriture scripte est moins contraignante que la plupart des alphabets utilisés dans les assistants personnels (Graphiti : figure 1.a). L'utilisateur n'a qu'à effectuer un lever de stylo entre chaque caractère (figure 1.b). Ceci permet de limiter les problèmes de segmentation de l'écriture cursive (figure 1.c). De plus, les moteurs de reconnaissance de l'écriture cursive ne sont pas encore assez robustes lorsqu'il y a beaucoup de classes (62 classes) et un grand lexique (188 795 entrées).

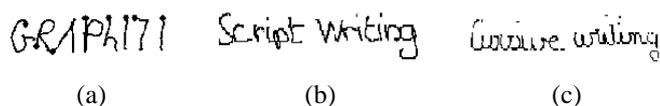


FIG. 1 : écriture contrainte (a), manuscrite (b) et cursive (c)

1.1 Acquisition des données

Le système de reconnaissance d'écriture en-ligne utilise une tablette à digitaliser qui délivre une suite de coordonnées (x, y) représentant le texte (contexte) à analyser.

L'apparition des ordinateurs nomades (sans clavier ni souris) montre tout l'intérêt de l'utilisation de la reconnaissance de l'écriture manuscrite. Le stylo remplace efficacement les moyens d'interaction traditionnels entre l'homme et la machine que sont le clavier et la souris. Le développement rapide des nouveaux PDAs et leur montée en puissance permettent d'envisager ce type de traitement évolué.

1.2 Structure du système

L'analyse de textes offre de plus larges perspectives que la simple analyse de mots effectuée par les moteurs de reconnaissance d'écriture cursive [2, 7, 8]. Mais la plus grande richesse des données d'acquisition (ligne de texte, ponctuation, silhouettes, etc.) rend l'analyse plus complexe.

Nous avons adopté provisoirement une structure « gauche droite » d'experts regroupés autour d'un classifieur de caractères de type script ; un premier groupe pour les pré-traitements et un second pour les post-traitements. Ces experts sont liés entre eux par un principe *d'induction / déduction* d'hypothèses (figure 2) qui permet d'améliorer les résultats au fil des experts. Un expert utilise le contexte et les hypothèses déduites par le module précédent pour émettre de nouvelles hypothèses. Ces hypothèses sont validées par l'expert lui-même (principe *d'émission / validation* d'hypothèses) pour déduire de nouvelles hypothèses. Nous pensons qu'un traitement cyclique comme proposé dans [1] pourrait augmenter l'efficacité du système.

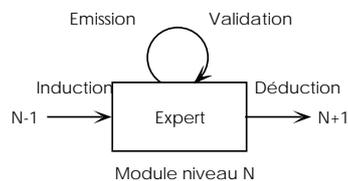


FIG. 2 : schéma d'un expert.

Les pré-traitements utilisent la suite de coordonnées délivrées par la tablette à digitaliser. Premièrement, le texte est segmenté en lignes. Ensuite, pour chaque ligne, on fusionne les traits (*strokes*) en caractères qui sont ensuite étiquetés : la ponctuation et les signes diacritiques sont laissés de côté

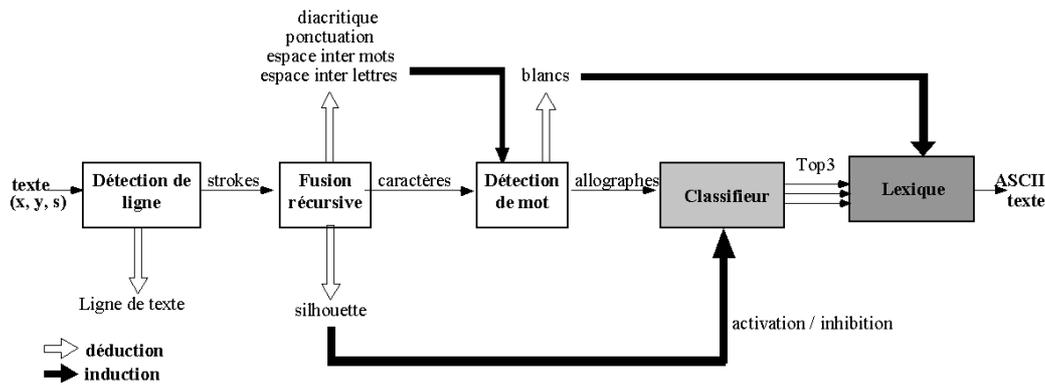


FIG. 3 : schéma synoptique du système.

tandis que les lettres sont classées. Un détecteur de blanc (espace) permet de regrouper les caractères en hypothèses de mot qui sont comparées aux entrées du lexique. La figure 3 précise le schéma complet du système.

La partie suivante est dédiée aux pré-traitements. Nous détaillerons les six experts qui réalisent respectivement la détection des sauts de lignes, la réorganisation des strokes, l'isolation des caractères, la détection des lignes de base, l'étiquetage des caractères et enfin la détection des blancs. La section 3 rappellera brièvement les derniers travaux sur les classificateurs de caractères isolés en-ligne [6]. Nous montrerons l'impact du post-traitement lexical sur le taux de reconnaissance du système dans la section 4. Nous finirons par la conclusion et les perspectives dans la section 5.

2. Pré-traitements

2.1 Détection des sauts de ligne

Le texte pouvant contenir plusieurs lignes, il est nécessaire d'extraire les strokes qui appartiennent à une même ligne. L'écriture latine étant de type gauche-droite, les sauts de ligne sont facilement détectables : on observe un brusque déplacement du stylo vers le bas et la gauche (figure 4.a). D'autres types de saut de ligne peuvent apparaître : lorsqu'on signe un document le déplacement du stylo se fait vers le bas et la droite (figure 4.b). Pour augmenter la fiabilité de l'expert nous utilisons un système d'émission/validation d'hypothèses. Cet expert peut être séparé en deux parties : la première émet des hypothèses sur les sauts de ligne et la seconde valide ou non ces dernières ne gardant que les sauts de lignes réels.

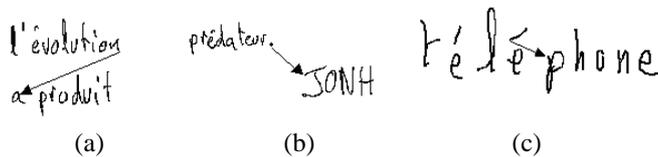


FIG. 4 : saut de ligne (a & b). Fausse détection (c).

- *Emission d'hypothèses* : tous les déplacements latéraux et vers le bas entre deux strokes consécutifs sont considérés comme des sauts de ligne hypothétiques. Les seuils de décision sont déduits de la largeur moyenne W_m et hauteur moyenne H_m des strokes composant le texte. Chaque déplacement (D_x, D_y) tel que $D_x > 2W_m$ et que $D_y > H_m$

est un saut hypothétique. Les seuils sont calculés en considérant les boîtes englobantes des strokes (à ce stade du traitement, les caractères ne sont pas encore segmentés).

- *Validation d'hypothèses* : grâce à ces sauts de ligne, les strokes appartenant à une même ligne hypothétique sont isolés et permettent l'approximation de lignes de base rectilignes. En comparant deux lignes de base consécutives, on peut vérifier s'il s'agit réellement d'un saut de ligne (les lignes sont distinctes : figure 5.a) ou non (les lignes sont entrelacées : figure 5.b). A la fin de cette étape et en raison du manque d'informations, les lignes de base sont considérées comme droites.

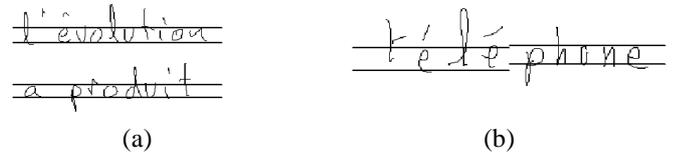


FIG. 5 : validation des sauts de lignes.

2.2 Réorganisation temporelle des strokes

L'analyse de texte manuscrit s'effectue caractère par caractère, il est donc nécessaire que les diacritiques suivent temporellement les caractères auxquels ils sont rattachés. De plus, il est fréquent que le scripteur revienne en arrière pour écrire ces signes. Pour éviter ces cassures dans le flot de donnée du contexte, il est nécessaire de réorganiser le signal de manière à avoir les strokes spatialement proches dans l'ordre chronologique. Donc, lorsqu'on détecte un retour en arrière entre deux strokes consécutifs, on détermine la lettre à laquelle est associé le signe diacritique (figure 6). Le flux de données étant organisé correctement, on peut effectuer la segmentation.



FIG. 6 : réorganisation temporelle des strokes.

2.3 Isolation des caractères

Dans l'écriture cursive, les mots doivent être segmentés en lettres. Dans l'écriture scripte, cette segmentation est implicite, mais les caractères peuvent contenir plusieurs strokes. Ces derniers doivent donc être fusionnés pour former un seul caractère. Cet expert utilise une procédure de fusion récursive qui regroupe les strokes consécutifs dont les boîtes englobantes répondent à certaines règles topologiques. La profondeur de récursivité est portée à quatre car la lettre contenant le maximum de strokes est le E majuscule avec quatre strokes. La figure 7 montre les résultats de la fusion des strokes.

Considérons deux strokes t_1 et t_2 avec leur boîte englobante respective $(x_{min1}, x_{max1}, y_{min1}, y_{max1})$ et $(x_{min2}, x_{max2}, y_{min2}, y_{max2})$. sachant que $x_{min1} < x_{min2}$, les règles de fusion sont les suivantes :

- recouvrement complet : $x_{max1} > x_{max2}$
- décalage à gauche : $x_{min2} - x_{min1} < \delta$
- décalage à droite : $x_{max2} - x_{max1} < \delta$
- recouvrement partiel : $x_{max1} - x_{min2} < \delta$

Le seuil δ est auto-adaptatif et varie durant la phase de fusion. Le meilleur compromis (peu de sur-segmentation et de sous-segmentation) a été obtenu [4] avec $\delta = \text{abs}(\delta_{max} - \delta_{min})/2$ où $\delta_{min} = (x_{max1} - x_{min1})/2$ et $\delta_{max} = (x_{max2} - x_{min2})/2$.

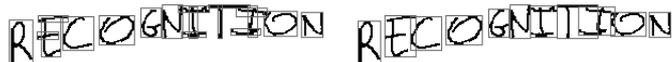


FIG. 7 : fusion récursive des strokes.

2.4 Etiquetage

A ce stade du traitement, nous avons assez d'informations pour savoir si un stroke est une lettre, une ponctuation ou une accentuation. Un premier étiquetage est effectué en étudiant les positions des boîtes englobantes par rapport aux lignes de base. Les strokes dont la boîte est au-dessus de la ligne de corps sont des diacritiques et les strokes descendant sous la ligne de base sont étiquetés comme ponctuation, il n'est donc pas nécessaire de classer ces derniers. Cette étape permet de créer trois classes différentes :

- *lettre* (à classer)
- *ponctuation* (mis de coté)
- *diacritique* (mis de coté)

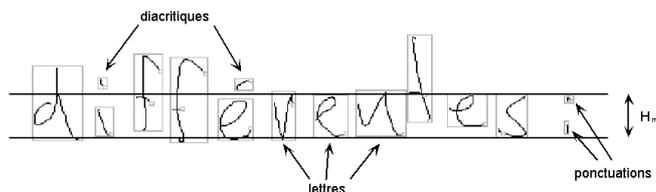


FIG. 8 : étiquetage des caractères.

2.5 Calcul des lignes de base et étiquetage des lettres

La connaissance de la position du caractère par rapport aux lignes de base permet de déterminer sa classe. Grâce à la silhouette des caractères, on peut les regrouper en plusieurs sous parties :

- *lettres moyennes* : **a, e, i**
- *grandes lettres* : majuscule, chiffre et ascendant : **b, d, h**
- *lettres avec jambage* : **g, j, p**
- *ascendant et descendant* : **f**

Grâce à ce regroupement, il est facile d'inhiber ou d'activer les experts de classification (voir §3) pour augmenter la vitesse de traitement et le taux de reconnaissance du classifieur. Ceci implique de calculer le plus précisément possible les lignes de base et de corps. C'est pour cela que nous avons décidé de calculer des lignes de base brisées au lieu de lignes droites. Nous commençons par isoler les caractères moyen en comparant la hauteur de leur boîte englobante à la hauteur moyenne H_m (voir §2.1). Ensuite, la ligne de base brisée est obtenue en reliant par des lignes droites le bas des boîtes englobantes restantes. De même pour la ligne de corps avec le haut des boîtes.

Un étiquetage des caractères sur leur topologie (ascendant ou descendant) est effectué en s'aidant de la position des boîtes englobantes par rapport aux lignes de base brisées.

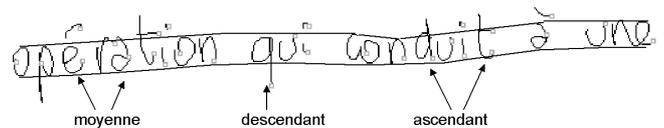


FIG. 9 : Lignes de base et de corps brisées.

2.6 Détection des blancs

Comme le post-traitement est de type lexical, il est nécessaire d'isoler les mots en recherchant les blancs qui les séparent. Pour cela, on étudie les espaces séparant les boîtes englobantes. Nous pouvons calculer deux seuils : le premier (T_l : inter-lettre) en dessous duquel un blanc est considéré comme séparant deux lettres et un second (T_w : inter-mot) au-dessus duquel un blanc sépare deux mots. Les blancs situés entre ces deux seuils sont considérés hypothétiques (figure 10). L'ambiguïté sera levée au moment du post-traitement lexical.

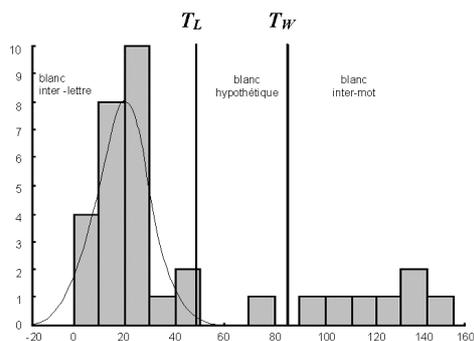


FIG. 10 : détection de blanc. Seuils inter-lettre et inter mot.

3. Classification

Le classifieur utilise des prototypes d'allographes (obtenus par l'algorithme de clustering décrit dans [6]) et une classification de type k -ppv. Il travaille sur 62 classes : minuscules, majuscules et chiffres. Il s'agit d'un classifieur par modélisation, donc chaque classe possède son propre expert (figure 11) et il devient possible d'activer ou d'inhiber les experts inutiles [5]. A la fin du processus de classification, nous gardons les trois meilleures hypothèses de lettres (top1, top2 et top3) qui seront utilisées en phase de correction lexicale.

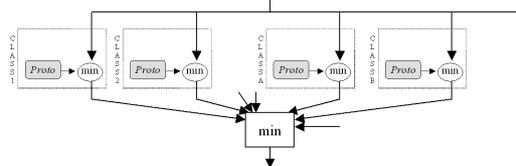


FIG. 11 : Structure du classifieur.

TAB. 1 : Taux de reconnaissance (Unipen Train R01-V07)

Classes	Prototypes	Tests	Top1	Top2	Top3
62	2 944	34 439	96,8 %	99,0 %	99,4 %

4. Correction lexicale

Les précédentes étapes nous fournissent des informations sur la classification, les blancs, la ponctuation et les diacritiques. Les blancs sont utilisés pour émettre des hypothèses sur les mots construites à partir des hypothèses de lettres top1, top2 et top3. Ces hypothèses sont validées ou corrigées en utilisant un dictionnaire français contenant environ 200 000 mots. Quand un mot est inconnu, nous effectuons une recherche de type 30-ppv sur le dictionnaire et choisissons le mot qui minimise le score du mot grâce au top1-3.

Si un blanc hypothétique apparaît, une règle de fusion simple a été implémentée : le blanc est supprimé si le score (la plus petite distance euclidienne entre le mot hypothèse et les mots du dictionnaire) du mot fusionné est plus petit ou égal à la somme des scores des mots séparés. Deux mots consécutifs m_1 et m_2 seront fusionnés s'il vérifie la relation suivante :

$$d(m_1+m_2) \leq d(m_1) + d(m_2)$$

5. Conclusion et perspectives

Le système a été testé sur une base de données de 1 000 mots (environ 5 000 lettres) écrits par 10 scripteurs différents. Nous avons obtenu les résultats suivants : 86 % de reconnaissance au niveau des mots et 97,2 % au niveau des lettres. Les erreurs sont réparties comme suit : sur/sous-segmentation 1,75 %, activation 2,17 %, classification 3,5 %. Le post-traitement lexical corrige 63,3 % de lettres fausses et 77,1 % des espaces hypothétiques. Le tableau 2 propose une comparaison du système avec différents moteurs de reconnaissance de mots cursifs. Les taux de reconnaissance sont encourageants ; ils peuvent être améliorés en utilisant un expert d'adaptation qui se baserait par exemple sur les invariants morphologiques du scripteur [3]. L'algorithme de clustering SMAC est appliqué sur les caractères du texte étiquetés en utilisant les résultats de la reconnaissance et viennent enrichir la base de prototypes personnelle du scripteur. Le système est donc toujours à vocation omni-scripteur et devrait être plus robuste en mode mono-scripteur.

TAB. 2 : Taux de reconnaissance des classifieurs (Unipen Train R01-V07)

Systèmes	mots	scripteurs	taille du lexique	taux de reconnaissance
Anquetil [8]	7 896	7	6 915	84,0 %
Kosmala [2]	2 000	3	200 000	90,6 %
Wimmer [7]	8 781	9	20 200	98,1 %
Oudot	1 054	10	188 795	86,0 %

Références

- [1] Cote M., Cheriet M., Lecolinet E. & Suen C.Y. *Automatic reading of cursive scripts using human knowledge*. ICDAR'97, pp 107-111, 1997.
- [2] Kosmala A., Willet D. & Rigoll G. *Advanced state clustering for very large vocabulary HMM-based on-line handwriting recognition*. ICDAR'99, pp 442-445, 1999.
- [3] Noury A., Heutte L., Paquet T. & Lecourtier Y. *Defining writer's invariants to adapt recognition*. ICDAR'99, pp 765-768, 1999.
- [4] Prevost L. & Milgram M. *Reconnaissance automatique de l'écriture scripte en mode omni-scripteur : un premier pas dans la conception d'un analyseur d'équations*. CIFED'98, pp 364-373, 1998.
- [5] Prevost L. & Milgram M. *Automatic Allograph Selection and Multiple Expert Classification for Totally Unconstrained Handwritten Character Recognition*. ICPR'98, (1) pp 295-302, 2000.
- [6] Prevost L. & Milgram M. *Modelizing character allographs in omni-scriptor frame : a new non-supervised algorithm*. Pattern Recognition Letters, 21(4) pp 295-302, 2000.
- [7] Wimmer Z., Dorizzi B. & Gallinari P. *Dictionary pre-selection in a neuro-markovian word recognition system*. ICDAR'99, pp 539-542, 1999.
- [8] Anquetil E. & Lorette G. *Perceptual Model of Handwriting Drawing Application to the Handwriting Segmentation Problem*, ICDAR'97, (2), 1997.