



# Apprentissage automatique d'une distance d'édition dédiée à la reconnaissance de l'écriture manuscrite

Sabine Carbonnel, Eric Anquetil

► **To cite this version:**

Sabine Carbonnel, Eric Anquetil. Apprentissage automatique d'une distance d'édition dédiée à la reconnaissance de l'écriture manuscrite. Jun 2004, 2004.

**HAL Id: sic\_00001198**

**[https://archivesic.ccsd.cnrs.fr/sic\\_00001198](https://archivesic.ccsd.cnrs.fr/sic_00001198)**

Submitted on 7 Dec 2004

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Apprentissage automatique d'une distance d'édition dédiée à la reconnaissance de l'écriture manuscrite

Sabine Carbonnel – Éric Anquetil

IRISA, INSA Rennes  
Campus universitaire de Beaulieu  
Avenue de Général Leclerc  
F-35 042 Rennes cedex

{sabine.carbonnel, eric.anquetil}@irisa.fr

**Résumé :** *Ce travail s'inscrit dans le cadre d'un post-traitement lexical basé à la fois sur une organisation structurée de dictionnaire (à partir de caractéristiques globales des mots) et sur un algorithme d'appariement de mots spécifique à l'écriture manuscrite en-ligne (distance d'édition). L'objectif est de compenser les erreurs de reconnaissance et de segmentation en s'appuyant sur les informations lexicales extraites d'un dictionnaire. Pour cela nous avons adapté une distance d'édition à l'écriture manuscrite pour le post-traitement lexical en reconnaissance en-ligne de mots. Cet article présente une méthode d'apprentissage automatique de la distance d'édition. L'objectif est de pouvoir automatiquement spécialiser la distance d'édition en fonction des propriétés du reconnaiseur. Les résultats expérimentaux obtenus montrent que l'approche proposée pour adapter automatiquement la distance d'édition obtient d'aussi bons résultats qu'une distance d'édition calibrée empiriquement à la main de façon longue et fastidieuse.*

**Mots-clés :** Reconnaissance d'écriture manuscrite en-ligne, post-traitement lexical, apprentissage de la distance d'édition.

## 1 Introduction

La communication par l'écrit et le geste graphique joue un rôle prépondérant dans les nouvelles technologies associées à l'informatique mobile. Les assistants personnels (PDA), les ordinateurs tablettes (tablette PC) ou encore les téléphones mobiles de nouvelle génération (smartphones) axent leurs interfaces homme machine sur des modalités d'interactions orientées stylet : l'utilisateur interagit et communique avec la machine en écrivant directement sur son écran. La qualité des systèmes de reconnaissance d'écriture est un élément clé dans l'ergonomie associée à ce type d'interactions. Nous collaborons depuis plusieurs années avec des industriels afin d'embarquer sur des téléphones mobiles des systèmes de reconnaissance de lettres et de mots manuscrits isolés (RESIFCar [ANQ 02] et RESIFMot [ANQ 97b]).

Dans cet article, nous nous focalisons sur l'étude de l'intégration de connaissances lexicales au système RESIFMot (reconnaissance de mots manuscrits cursifs liés) [ANQ 97b]; plus particulièrement sur la distance d'édition utilisée lors du post-traitement lexical. Une distance d'édition classique ne permettant pas de résoudre

les problèmes spécifiques rencontrés en reconnaissance d'écriture, des adaptations ont donc été réalisées pour mieux les prendre en compte. Les résultats obtenus sont meilleurs mais il faut estimer toutes les opérations d'édition possibles et leurs coûts de façon manuelle et empirique [CAR 03]. Dans cet article, nous présentons une méthode automatique pour établir à partir des propriétés du reconnaiseur l'ensemble des opérations d'édition ainsi que les coûts qui leurs sont associés.

Dans le paragraphe suivant nous présentons succinctement les principes du système de reconnaissance RESIFMot et du post-traitement lexical qui lui est associé. Nous introduisons dans le paragraphe 3 la notion de distance d'édition avant de proposer, dans le paragraphe 4 une méthode d'apprentissage automatique de distance d'édition. Dans le dernier paragraphe sont exposés les résultats expérimentaux permettant de comparer la distance d'édition résultant de l'apprentissage automatique présenté avec une distance d'édition élaborée de façon empirique.

## 2 Présentation générale du système de reconnaissance et du post-traitement

### 2.1 Le système de reconnaissance de mots isolés RESIFMot

Le système de reconnaissance de mots manuscrits en-ligne RESIFMot est basé sur une approche analytique où les mots sont segmentés suivant différentes hypothèses d'allographes<sup>1</sup> de lettres [ANQ 96, ANQ 97b]. Les hypothèses d'allographes sont organisées dans un graphe de segmentation structuré représentant l'ensemble des segmentations possibles (Figure 1).

Après l'étape de segmentation, une version adaptée de RESIFCar est utilisée pour identifier chaque hypothèse d'allographe. Cette approche analytique engendre plusieurs hypothèses de reconnaissance (séquences d'allographes de lettres) qui sont associées à deux types de problèmes : les ambiguïtés de segmentation et les confusions inter-lettres induites par la phase d'identification des allographes.

---

<sup>1</sup>variantes de styles dans une même classe de lettre

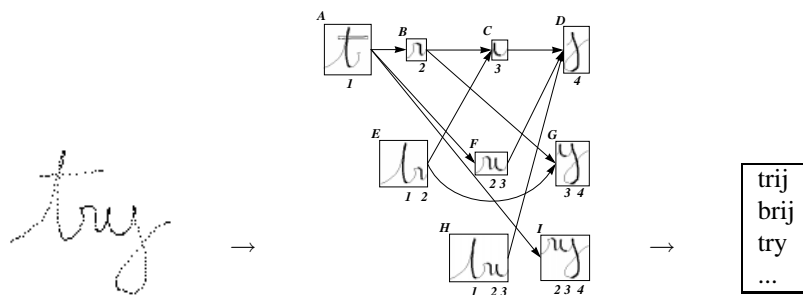


FIG. 1 – Graphe de segmentation et hypothèses de séquences de lettres associées.

Le rôle du post-traitement lexical que nous présentons dans cet article est d'identifier le mot manuscrit à partir d'une part des hypothèses produites par la reconnaissance (séquences de lettres) et d'autre part en fonction d'un corpus choisi (dictionnaire de mots).

## 2.2 Principe de base du post-traitement lexical

Nous nous intéressons ici à l'exploitation d'informations de niveau lexical pour la reconnaissance de mots isolés. Le post-traitement lexical correspond à une étape de désambiguïsation des hypothèses de reconnaissance par l'introduction de connaissances contextuelles de nature lexicale. Autrement dit il va permettre de corriger ou de valider les résultats de la reconnaissance.

Ce post-traitement est souvent basé sur deux sortes d'informations [DEN 97] : soit une représentation statistique de n-grammes de lettres appris dans des corpus de textes, soit un lexique (ou dictionnaire) contenant une liste de mots valides. Les méthodes se basant sur des informations probabilistes sont souvent utilisées pour ordonner les hypothèses de mots issues de la reconnaissance, mais pas pour valider leur appartenance à un lexique [GUY 96].

Le principal objectif de la phase de post-traitement est ici de déterminer quel est le meilleur mot du lexique pouvant correspondre aux hypothèses de reconnaissance. Nous présentons dans cet article une approche se basant sur un lexique. Il est cependant intéressant de souligner qu'une approche combinant des informations de nature statistique et lexicales est une perspective envisagée.

Une méthode de post-traitement basée sur un lexique s'accompagne d'un algorithme d'appariement de chaînes (distance d'édition) permettant de comparer ces propositions avec les mots du lexique. La complexité du calcul de la distance entre deux chaînes de caractères peut rendre le post-traitement inefficace dans le contexte d'un grand vocabulaire (plus de 10 000 mots). Il est alors intéressant d'organiser le lexique de façon à ne pas réaliser toutes les comparaisons. Ce qui revient à créer des sous-lexiques et ne comparer les hypothèses de reconnaissance qu'avec un ou plusieurs sous-lexiques sélectionnés de façon pertinente (et dans lequel doit donc se trouver *a priori* le mot à reconnaître) : c'est ce que l'on appelle une *réduction* du lexique [GIL 98, SEN 96, CAR 04].

Le post-traitement lexical que nous utilisons est composée de deux phases :

- apprentissage et indexation a priori des sous-lexiques à

- partir du lexique ;
- exploitation des sous-lexiques.

Nous avons présenté et comparé deux approches pour l'apprentissage et l'indexation des sous-lexiques dans [CAR 04]. Les sous-lexiques sont définis à partir d'une notion de proximité visuelle des mots. Pour les deux approches, la phase de post-traitement se base sur la comparaison des hypothèses de reconnaissance avec les mots d'un sous-lexique pertinent sélectionné (taux de réduction du lexique : de 1/25 à 1/8 selon l'approche). Cette comparaison s'appuie sur une distance d'édition spécifique à l'écriture manuscrite. Nous présentons dans cet article une méthode d'apprentissage automatique de la distance d'édition spécifique qui va permettre de s'adapter automatiquement au système de reconnaissance utilisé.

## 3 Principe d'une distance d'édition dédiée à la reconnaissance d'écriture manuscrite

Une distance d'édition permet de comparer deux chaînes de caractères : la distance entre deux chaînes correspond au coût minimal du passage de l'une à l'autre en terme d'opérations d'éditations sur des caractères (substitution, insertion, suppression). Plusieurs distances peuvent être utilisées pour corriger et valider les hypothèses. Nous présentons d'abord une version étendue à l'écriture manuscrite de la distance de Levenshtein [DAM 64, LEV 66], puis une version optimisée plus adaptée au reconnaiseur RESIFMot.

### 3.1 Distance d'édition étendue pour l'écriture manuscrite

La distance de Damerau et Levenshtein [DAM 64, LEV 66] est composée de trois opérations de base adaptées aux erreurs de saisie au clavier : substitution, suppression (*deletion*) et insertion de caractères. Seni, Kripasundar et Srihari [SEN 95] ont étendu la distance de Damerau et Levenshtein pour compenser les erreurs de segmentation produites par le processus de reconnaissance d'écriture manuscrite. Trois opérations à coûts statiques ont été rajoutées : fusion, division et substitution de paires. À chaque étape, cette distance entre deux séquences de caractères  $X = x_1 \dots x_i \dots x_n$  et  $Y = y_1 \dots y_j \dots y_m$  est calculée de la façon suivante :

$$d(i, j) = \min \begin{cases} d(i-1, j-1) & +c_{Sub}(x_i, y_j) \\ d(i-1, j) & +c_{Ins}(x_i) \\ d(i, j-1) & +c_{Del}(y_j) \\ d(i-2, j-1) & +c_{Div}(x_{i-1}x_i, y_j) \\ d(i-1, j-2) & +c_{Fus}(x_i, y_{j-1}y_j) \\ d(i-2, j-2) & +c_{PSub}(x_{i-1}x_i, y_{j-1}y_j) \end{cases}$$

où  $d(0, 0) = 0$ ,  $d(i, 0) = \sum_{k=1}^i c_I(x_k)$ ,  $d(0, j) = \sum_{k=1}^j c_D(y_k)$  et  $c_{Sub}$ ,  $c_{Del}$ ,  $c_{Ins}$ ,  $c_{Div}$ ,  $c_{Fus}$  et  $c_{PSub}$  représentent respectivement les coûts des opérations d'édition: substitution, suppression, insertion, division, fusion et substitution de paires. Ces coûts sont statiques, c'est à dire qu'ils peuvent être de trois catégories pour chaque opération (très probable, probable et improbable). Les tables des caractères ainsi que les valeurs associées à chaque catégorie de chaque opération sont déterminés de façon empirique (table 1).

| fusion très probable | fusion probable |
|----------------------|-----------------|
| cl → d               | ls → h          |
| rr → u               | oj → g          |
| cr → a               | cc → n          |
| ... → ...            | ... → ...       |

TAB. 1 – Exemple d'opérations d'éditons.

### 3.2 Raffinement de la distance d'édition étendue : distance modifiée

Nous avons récemment optimisé [CAR 03] cette distance étendue en introduisant des poids dynamiques aux coûts statiques en rapport avec des caractéristiques globales des lettres. Cette nouvelle distance est notée  $dist_{modif}$  pour distance modifiée. Les poids dynamiques sont définis suivant un facteur de pénalisation reflétant les différences structurelles entre les caractères comparés (premier niveau de modélisation du système RESIFCar [ANQ 02]); c'est à dire qu'ils dépendent des silhouettes des caractères et sont ajoutés aux coûts statiques définis empiriquement. Un autre facteur de pénalisation a été introduit. Il se base sur l'absence ou la présence de signes diacritiques (point, barre de t, accent, etc.) qui sont détectés de manière globale lors de la reconnaissance [ANQ 97a]. Pour optimiser le temps de calcul en évitant des comparaisons inutiles, nous avons également introduit un seuil dynamique pour élaguer le calcul de la distance dans la matrice d'édition [CAR 03].

Cette nouvelle distance d'édition offre de bons résultats en comparaison avec la distance étendue (diminution du taux d'erreur de reconnaissance : 11%). Cependant elle nécessite, comme la distance étendue de définir de façon empirique la liste exhaustive des différentes opérations possibles ainsi que l'ensemble des coûts qui leurs sont associés (voir l'exemple du tableau 1). Étant donné la multiplicité des opérations possibles (substitution, fusion de caractères, etc), ce travail s'avère long et fastidieux.

Nous avons donc introduit un mécanisme d'apprentissage automatique, présenté dans la suite de l'article, qui va permettre de définir l'ensemble des opérations possibles ainsi que leurs coûts respectifs.

## 4 Apprentissage automatique de la distance d'édition

Il est assez difficile de déterminer manuellement l'ensemble des tables et les coûts associées aux différentes opérations d'édition. De plus ces opérations sont dépendantes de façon plus ou moins forte des propriétés du reconnaisseur utilisé. Pour automatiser la constitution de toutes ces tables nous avons mis en place un apprentissage automatique reposant sur le principe suivant (voir figure 2). L'apprentissage est basé sur deux étapes :

- la première étape utilise le système de reconnaissance de mots en mode supervisé afin d'estimer l'ensemble des opérations d'édition possibles, en s'appuyant sur une distance reflétant uniquement les règles de base d'appariement ( $dist_{appr}$ );
- la seconde étape évalue à partir de ces estimatifs le coût des opérations rencontrées afin de définir l'ensemble des paramètres de la distance d'édition.

Ces deux étapes sont présentées en détail dans les paragraphes 4.2 et 4.3.

### 4.1 Définitions des règles de base de l'appariement $dist_{appr}$

Les règles de base d'appariement sont définies à partir de la silhouette des mots (cf. figure 3). La silhouette d'un mot représente la suite des traits descendants fondamentaux qui le composent.

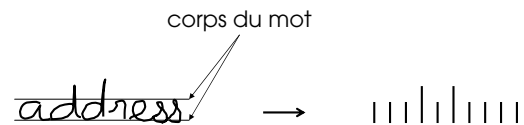


FIG. 3 – Exemple de silhouette.

Le tableau 2 référence les différents types de traits descendants permettant de définir la silhouette d'un mot. À partir de ce tableau, la silhouette du mot *address* peut être définie par *mmmamamm*. La silhouette est robuste et indépendante du style d'écriture pour la plupart des caractères.

| nom des traits descendants | description                         | notation |
|----------------------------|-------------------------------------|----------|
| médian                     | dans le corps du mot                | m        |
| ascendant                  | dépasse vers le haut                | a        |
| descendant                 | dépasse vers le bas                 | d        |
| long                       | dépasse vers le haut et vers le bas | l        |

TAB. 2 – Traits descendants fondamentaux composant les silhouettes.

#### Notations :

Nous utilisons la notation  $sil(m)$  pour désigner la silhouette du mot  $m$ . On nomme  $mot_{ref}$ , la transcription informatique du mot qui a été écrit pas le scribeur et doit être reconnu.

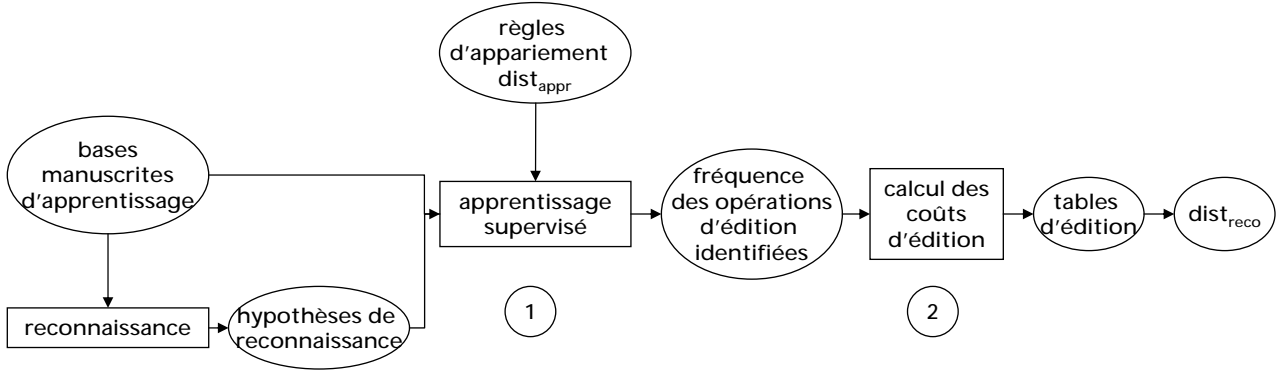


FIG. 2 – Principe de l'apprentissage de la distance d'édition.

$hyp$  désigne une suite de caractères résultant du processus de reconnaissance, et qui est donc une hypothèse pour le post-traitement.

Les règles de base d'appariement reposent sur la définition d'une distance notée  $dist_{appr}$  qui s'appuie sur les traits descendants. Cette distance d'apprentissage  $dist_{appr}$  est composée des six opérations : substitution, insertion, suppression, fusion, division et substitution de paires de caractères. À chaque opération est associé un coût fixe évalué en fonction de la silhouette des caractères mis en jeu. Pour les opérations du type  $op_1$  (substitution, fusion, division et substitution de paire), on a :

$$cout_{op_1}(A, B) = \begin{cases} 0 & \text{si } A = B \\ cout_{fixe_{op_1}} & \text{si } sil(A) = sil(B) \\ cout_{infini} & \text{sinon} \end{cases}$$

Les coûts des opérations du type  $op_2$  (insertion et suppression) sont calculés de la façon suivante :

$$cout_{op_2}(A) = \begin{cases} cout_{fixe_{op_2}} & \text{si } A \text{ n'est composé que de} \\ & \text{traits médians} \\ cout_{infini} & \text{sinon} \end{cases}$$

Les coûts fixes associés aux opérations du type  $op_1$  et  $op_2$  ont été définis expérimentalement en s'inspirant de la distance modifiée (paragraphe 3.2). Cette distance basique ( $dist_{appr}$ ) repose sur des principes associés aux structures fondamentales des mots (silhouette). Elle est donc indépendante du reconnaissseur utilisé. Elle permet d'apparier le mot de référence ( $mot_{ref}$ ) avec l'hypothèse ( $hyp$ ) fournie par le reconnaissseur. Les opérations d'édition retenues et comptabilisées dans les tables d'apprentissage correspondent à celles utilisées pour le calcul de la distance minimale.

## 4.2 Étape 1 : calcul des fréquence des opérations d'édition

On utilise  $dist_{appr}$  pour apparier  $mot_{ref}$  et  $hyp$ . On dispose pour l'appariement d'une information issue du processus de reconnaissance sur la taille des hypothèses (nombres de traits descendants) ainsi que sur la répartition de ces traits dans les caractères composant les hypothèses. Les opérations d'insertion et de suppression ne sont autorisées que si le nombre de traits descendants de l'hypothèse est mal détecté.

À la fin de chaque appariement, les couples de caractères (ou de suites de caractères) en confusion sont stockés dans des

tables. Les tables permettant de stocker les occurrences de chaque opération sont de la forme :

- table de substitution :  $(a, x) \rightarrow nb_{sub}$
- table de fusion :  $(ab, x) \rightarrow nb_{fus}$
- table de division :  $(a, xy) \rightarrow nb_{div}$
- table de substitution de paires :  $(ab, xy) \rightarrow nb_{sp}$
- table d'insertion :  $(\varepsilon, x) \rightarrow nb_{ins}$
- table de suppression :  $(a, \varepsilon) \rightarrow nb_{sup}$

où  $a$  et  $b$  représentent des caractères de  $hyp$ ,  $x$  et  $y$  des caractères des  $mot_{ref}$  et  $\varepsilon$  le caractère vide ;  $nb_{op}$  correspond à la fréquence de confusion lors de l'appariement de  $mot_{ref}$  et  $hyp$  pendant l'apprentissage pour chaque opération d'édition ( $op =$  substitution, fusion, division, substitution de paire, insertion ou suppression).

Prenons un exemple simple pour illustrer le principe. Considérons l'appariement de la séquence  $vmaximum$  ( $hyp$ ) et du mot  $maximum$  ( $mot_{ref}$ ). Le calcul de la distance permet de déterminer que la distance minimale (coût du passage de  $hyp$  à  $mot_{ref}$ ) est obtenue en supprimant  $v$  et en fusionnant les caractères  $s$  et  $c$  en  $x$ . On rajoute donc le caractère  $v$  dans la table des suppressions et le couple  $(sc, x)$  dans la table de fusion s'ils ne sont pas déjà présent, sinon on incrémente le compteur qui leur est associé.

Cet apprentissage est réalisé à partir des appariements des  $n$  premières hypothèses de reconnaissance ( $hyp$ ) avec le mots de référence ( $mot_{ref}$ ). Il est possible de donner plus d'importance aux opérations provenant de l'appariement entre  $mot_{ref}$  et l'hypothèse de reconnaissance la plus proche (distance minimale) en pondérant dans les tables de confusion les couples associés à ces opérations.

## 4.3 Étape 2 : calcul des coûts d'édition

Cette étape correspond à l'attribution de coûts à l'ensemble des opérations (tables de confusions) rencontrées lors de l'apprentissage à partir de leurs fréquence d'apparition. L'idée principale est de donner un coût faible aux confusions très courantes (les plus rencontrées en apprentissage) et un coût fort aux moins courantes. Pour cela une règle simple consiste à évaluer le coût de chaque opération de façon inversement proportionnelle à sa fréquence d'apparition.

Il est intéressant de remarquer que cette technique d'apprentissage pourrait être itérée plusieurs fois afin de raffiner les coûts en utilisant à chaque étape la dernière distance apprise ( $dist_{reco}$ ) comme nouvelle distance d'apprentissage

( $dist_{appr}$ ). Une autre piste intéressante serait de s'intéresser aux techniques de *boosting* pour favoriser, lors du processus itératif, les opérations correspondant aux mots qui sont mal reconnus.

## 5 Expérimentations et résultats

Le but de ces expérimentations est de comparer les taux de reconnaissance obtenus avec  $dist_{modif}$  (distance modifiée mise au point de façon empirique, cf. paragraphe 3) et  $dist_{reco}$  (nouvelle distance d'édition apprise automatiquement par la méthode présentée, cf. paragraphe 4). L'apprentissage de  $dist_{reco}$  a été réalisé à partir d'une base de 3 000 mots manuscrits écrits par quatre scripteurs différents. L'évaluation de la distance a été ensuite effectuée sur une autre base composée de 3 000 mots manuscrits écrits par trois autres scripteurs. L'apprentissage du moteur de reconnaissance est également indépendant de ces sept scripteurs. Le cadre expérimental de cette évaluation peut être qualifié "d'omni-scripteur" puisque l'ensemble des scripteurs utilisé pour l'apprentissage est complètement différent de celui utilisé pour l'évaluation (néanmoins il sera nécessaire de poursuivre ces expérimentations sur un plus grand nombre de scripteurs pour valider réellement le caractère omni-scripteur des résultats). Le lexique que nous avons utilisé pour le post-traitement contient 25 000 mots de langue anglaise et comprend les mots de la base manuscrite. La fréquence des opérations apprises lors de l'appariement entre le mot de référence et la meilleure hypothèse a été pondérée par 10, et nous avons utilisé pour chaque mot de la base d'apprentissage les 10 premières hypothèses de reconnaissance.

| <i>rang</i> | $dist_{modif}$ | $dist_{reco}$ |
|-------------|----------------|---------------|
| 1           | 72.69%         | 74.17%        |
| 2           | 78.81%         | 80.56%        |
| 3           | 81.27%         | 83.34%        |
| 4           | 83.37%         | 84.76%        |
| 5           | 84.23%         | 85.79%        |

TAB. 3 – Taux de reconnaissance ; cadre omni-scripteur, vocabulaire de 25 000 mots.

Les résultats reportés dans la table 3 montrent que la distance apprise automatiquement ( $dist_{reco}$ ) permet d'obtenir des résultats comparables et même légèrement meilleurs à ceux obtenus par la distance modifiée mise au point de façon empirique  $dist_{modif}$  (réduction du taux d'erreur : 5.7%). Ces résultats sont très encourageants. Ils laissent notamment entrevoir la possibilité de suivre au mieux les évolutions du système de reconnaissance en adaptant automatiquement la distance d'édition. L'analyse des cas d'erreurs restant montre que les limites d'une correction par distance d'édition sont presque atteintes ; et qu'il faudra donc envisager l'intégration d'autres connaissances pour améliorer les résultats de reconnaissance.

## 6 Conclusion

Nous avons présenté dans cet article une méthode d'apprentissage automatique d'une distance d'édition dédiée à l'écriture manuscrite. Cette méthode permet de s'abstraire

des phases de mise au point longues et fastidieuses qu'il fallait opérer sur les distances d'édition précédentes. Les résultats expérimentaux obtenus sont très encourageants. Ils ont montré qu'il était possible de s'adapter automatiquement aux propriétés du reconnaiseur d'écriture et ainsi de suivre ses évolutions au fur et à mesure de son optimisation et de sa mise au point. Nous avons évoqué dans cet article plusieurs pistes d'optimisation de procédé d'apprentissage en raffinant le cycle itératif du calcul de la distance d'édition. Ceci ouvre des perspectives que nous allons explorer dans nos prochains travaux qui s'orienteront aussi vers la combinaison d'informations lexicales de nature statistique avec celles que nous utilisons déjà.

## 7 Remerciements

Une partie de ce travail a été réalisée avec la collaboration de France Telecom R&D. Nous remercions Éric Petit et Sylvie Vidal de France Telecom R&D ainsi que Guy Lorette de L'université de Rennes 1 pour leurs nombreux commentaires constructifs fournis au cours de ce travail.

## Références

- [ANQ 96] ANQUETIL E., LORETTE G., On-line Handwriting Character Recognition System Based on Hierarchical Qualitative Fuzzy Modeling, *Fifth International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, Colchester, Angleterre, September 1996, pp. 47-52.
- [ANQ 97a] ANQUETIL E., Modélisation et reconnaissance par la logique floue : application à la lecture automatique en-ligne de l'écriture manuscrite omni-scripteur, PhD thesis, université de Rennes 1, janvier 1997.
- [ANQ 97b] ANQUETIL E., LORETTE G., Perceptual Model of Handwriting Drawing Application to the Handwriting Segmentation Problem, *International Conference on Document Analysis and Recognition*, vol. 1, Ulm, Germany, August 1997, pp. 112-117.
- [ANQ 02] ANQUETIL E., BOUCHEREAU H., Integration of an On-Line Handwriting Recognition System in a Smart Phone Device, *International Conference on Pattern Recognition*, vol. 3, Quebec, Canada, August 2002, pp. 192-196.
- [CAR 03] CARBONNEL S., ANQUETIL E., Lexical Post-Processing Optimization for Handwritten Word Recognition, *International Conference on Document Analysis and Recognition (ICDAR'03)*, Edinburgh, Août 2003, pp. 477-481.
- [CAR 04] CARBONNEL S., ANQUETIL E., Modélisation et intégration de connaissances lexicales pour le post-traitement de l'écriture manuscrite en-ligne, *Reconnaissance de Forme et Intelligence Artificielle (RFIA'04)*, vol. 3, Toulouse, France, Janvier 2004, pp. 1313-1322.
- [DAM 64] DAMEREAU F., A technique for computer detection and correction of spelling errors, *Communications of the ACM*, vol. 7, 1964, pp. 659-664.
- [DEN 97] DENGEL A., HOCH R., HÖNES F., JÄGER T., MALBURG M., WEIGEL A., Techniques for Improving OCR Results, *Handbook on Character Recognition and*

*Document Image Analysis*, Chapitre 8, World Scientific Publishing Company, 1997.

- [GIL 98] GILLOUX M., Réduction dynamique du lexique par la méthode Tabou, *CIFED*, Quebec, Canada, May 1998, pp. 24-31.
- [GUY 96] GUYON I., SCHENKEL M., DENKER J., Overview and synthesis of on-line cursive handwriting recognition techniques, *Handbook on Optical Character Recognition and Document Image Analysis*, Chapitre 8, World Scientific Publishing Company, 1996.
- [LEV 66] LEVENSHTAIN, Binary Codes Capable of Correcting Deletions, Insertions and Reversals, *Soviet Physics Doklady*, vol. 10, 1966, pp. 707-710.
- [SEN 95] SENI G., KRIPASUNDAR V., SRIHARI R., Generalizing edit distance for handwritten text recognition, *Proceedings of SPIE/IS&T Conference on Document Recognition*, San Jose, CA, February 1995, pp. 54-65.
- [SEN 96] SENI G., SRIHARI R., NASRABADI N., Large Vocabulary Recognition of on-line Handwritten Cursive Words, *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 18, n° 7, 1996.