Modélisation et intégration de connaissances lexicales pour le post-traitement de l'écriture manuscrite en-ligne

Lexical Knowledges Modelisation and Integration for on-line Handwriting Recognition Post-Processing

Sabine Carbonnel Éric Anquetil

IRISA, INSA

Campus de Beaulieu
F-35042 Rennes Cedex, France
sabine.carbonnel@irisa.fr

Résumé

Cet article présente des optimisations de post-traitement lexical pour la reconnaissance de mots manuscrits. Le but de ce travail est d'explorer la combinaison de différentes approches de post-traitement lexical, pour optimiser le taux de reconnaissance, le temps de reconnaissance et l'encombrement mémoire d'un système de reconnaissance en-ligne d'écriture manuscrite. Les méthodes présentées se basent sur : une organisation de dictionnaire avec filtrage des mots, à partir de caractéristiques globales des mots ; un algorithme de comparaison de mots spécifique à l'écriture manuscrite en-ligne (pour compenser les erreurs de reconnaissance et de segmentation) et une stratégie d'exploration des résultats provenant du processus analytique de reconnaissance de mots. Les résultats expérimentaux obtenus (avec des lexiques de 1 000, 7 000 et 25 000 mots) permettent d'évaluer les différentes stratégies d'optimisation suivant le temps de reconnaissance et l'encombrement mémoire.

Mots Clef

Reconnaissance d'écriture manuscrite, système omni-scripteur, post-traitement lexical, réduction du dictionnaire, classification non supervisée

Abstract

This paper presents a lexical post-processing optimization for on-line handwritten word recognition. The aim of this work is to explore the combination of different lexical post-processing approaches in order to optimize the recognition rate, the recognition time and memory requirements of an on-line handwritten recognizer system. The present method focuses on the following tasks: a lexicon organization with word filtering, based on holistic word features to deal with large vocabulary (creation of static sub-lexicon compressed in a trie structure); a dedicated string matching algorithm for on-line handwriting (to compensate the recognition and the segmentation errors); and a specific exploration strategy of the results provided by the analytical word recognition process. Experimental results are reported using several lexicon sizes (about 1,000; 7,000 and 25,000 entries) to evaluate different optimization strategies according to the recognition rate, computational cost and memory requirements.

Keywords

Handwriting recognition, lexical post-processing, lexicon reduction, clustering

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. Plusieurs systèmes de reconnaissance d'écriture ont été intégrés à ces architectures mobiles. On peut les caractériser en mettant en rapport leurs taux de reconnaissance, leurs performances (nombre de classes de caractères modélisés, taille du dictionnaire de mots associé) et les contraintes qui leurs sont associées : étendue des styles d'écriture reconnus, gestion du multi-traits , ressources mémoires utilisées, temps de calcul par caractère, par mot,

etc. Ces travaux s'intéressent plus particulièrement à l'embarquement de systèmes de reconnaissance d'écriture manuscrite en-ligne sur des plate-formes à capacité réduite telles que les téléphones mobiles de nouvelle génération ou smartphones. Dans ce contexte, nous avons collaboré durant ces deux dernières années avec un industriel, spécialisé dans la conception de téléphones portables de nouvelle génération, afin d'embarquer le système de reconnaissance de caractères manuscrits isolés RESIFCar, résultat de nos travaux de recherche débutés en 1993 [1]. L'architecture mobile sur laquelle nous avons travaillé est basée sur un processeur ARM 7 TDMI cadencé à 13 Mhz et les ressources mémoires disponibles pour le module embarqué de reconnaissance de caractères étaient limitées à 50Ko en RAM et à 200Ko en ROM. Le challenge de ce travail était de pouvoir répondre à ces contraintes matérielles en maintenant les capacités du système de reconnaissance de caractères pour aboutir à une interaction conviviale et peu contrainte avec l'utilisateur : tolérance sur les styles d'écriture, gestion homogène de la reconnaissance des lettres, des chiffres et des symboles spéciaux, temps de reconnaissance raisonnable (de l'ordre de 0,5 seconde par caractère).

Nos travaux de recherche se poursuivent aujourd'hui sur l'embarquement d'un système de reconnaissance de mots manuscrits cursifs liés. Ce système, dénommé RESIFMot, s'appuie sur la technologie RESIFCar. Une première version du système de reconnaissance RESIFMot a été présentée dans [3].

Dans cet article, nous nous focalisons sur l'étude de l'intégration et de la modélisation de connaissances lexicales au système RESIFMot. L'objectif est de gérer des lexiques de grande taille en tenant compte des contraintes matérielles très fortes (ressources et mémoire limités). Le système RESIFMot étant actuellement en phase de finalisation, les taux de reconnaissance présentés dans cet article ont pour objectif de mettre en évidence les apports relatifs des différentes approches de post-traitement lexical.

Ce travail s'articule autour de l'étude de combinaisons de différentes méthodes afin de réduire le temps de calcul et les ressources mémoires impliquées dans le post-traitement lexical et ceci en limitant au maximum les dégradations des taux de reconnaissance.

Dans le paragraphe suivant nous présentons le sytème de reconnaissance RESIFMot et les principes d'un post-traitement lexical. Puis nous présentons les principes de deux approches présentées pour le post-traitment lexical, nous détaillons dans le paragraphe 3 pour chacune d'elle la phase d'apprentissage (ou d'organisation) des sous-lexiques. L'exploitation des connaissances lexicales est présenté au paragraphe 4. Dans le dernier paragraphe sont exposés les résultats expérimentaux comparatifs des approches de post-traitement présentées sur la base de dictionnaires de différentes tailles.

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

2.1 Le système de reconnaissance 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 (variantes de styles dans une même classe de lettre) de lettres. Ce processus de segmentation [3] utilise des structures d'ancrage correspondant aux traits descendants pertinents modélisés pour chaque classe de lettres [2]. Les hypothèses d'allographes sont organisées dans un graphe de segmentation structuré représentant l'ensemble des segmentations possibles (Figure 1).

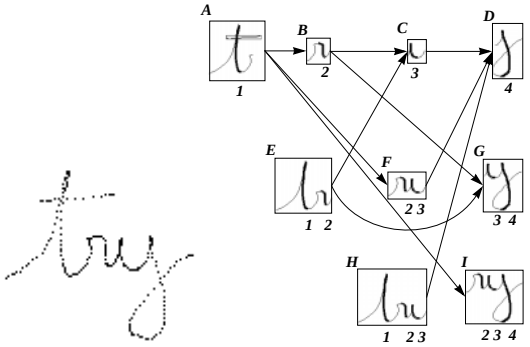

FIG. 1 – *Graphe de segmentation du mot try*

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 ;
- les confusions inter-lettres induites par la phase d'identification des allographes.

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

Trois niveaux de connaissances sont généralement distingués en linguistique : lexical, syntaxique et sémantique.
Nous nous intéressons ici uniquement à l'organisation et à l'exploitation du 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 lexicales. Autrement dit le post-traitement lexical va permettre de valider ou de corriger les résultats de la reconnaissance.

Ce post-traitement est souvent basé sur deux sortes d'informations [7] :

- un lexique (dictionnaire) contenant une liste de mots valides. Les hypothèses de reconnaissance (séquences de lettres ne correspondant pas forcément à des mots) sont alors comparées aux mots du lexique en vue d'être validées ou éventuellement corrigées.
- une représentation statistique de n-grammes de lettres, souvent formalisées par des modèles probabilistes tels que des modèles de Markov à longueur variable [10]. Cela permet de réordonner les hypothèses de reconnaissance d'après des informations statistiques (apprises dans des corpus de textes) des n-grammes qui les composent.

Une méthode basée sur un lexique produit des informations pertinentes pour valider et corriger les propositions issues de l'étape de segmentation. Les méthodes permettant de comparer ces propositions avec les mots du lexique s'appuient sur des distances d'éditions. 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 (10 000 à 100 000 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 sous-lexique sélectionné 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 [9, 15].

Les méthodes se basant sur des informations probabilistes donnent une représentation plus compacte mais aussi plus partielle des connaissances lexicales. Ces méthodes sont souvent utilisées pour ordonner les hypothèses de mots issues de la reconnaissance, mais pas pour valider leur appartenance à un vocabulaire. Les informations statistiques sur les n-grammes de lettres sont apprises à partir de corpus de textes (ex : le corpus de Brown [5]).

Le reconnaisseur de mots RESIFMot est basé sur une approche analytique qui produit des séquences de lettres (hypothèses de reconnaissance) à partir de la segmentation et du processus de reconnaissance de lettres. Le principal objectif de la phase de post-traitement est ici de déterminer le meilleur mot correspondant.

Nous présentons dans cet article des approches se basant sur des lexiques. Il est cependant intéressant de souligner qu'une combinaison à des informations de nature statistique est une perspective envisagée.

Les deux approches de post-traitement lexical que nous présentons sont composées de deux phases :

- l'apprentissage et l'indexation des sous-lexiques à partir du lexique complet (phase d'organisation : figure 2) ;
- l'exploitation du lexique organisé en sous-lexiques (phase de post-traitement : figure 3).

Pour les deux approches, la phase de post-traitement se base sur la comparaison (à l'aide d'une distance d'édition) des hypothèses de reconnaissance avec les mots d'un sous-lexique pertinent sélectionné parmi tous les sous-lexiques. Les choix concernant l'organisation du lexique, les index et la façon d'effectuer la réduction sont donc très importants car ils influent directement sur la qualité du post-traitement (rapidité, taux de reconnaissance et encombrement mémoire).

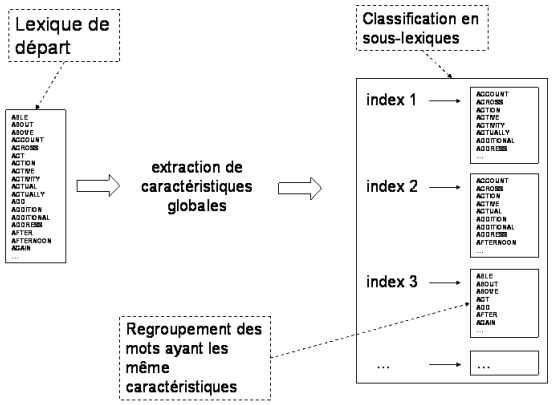

FIG. 2 – *Apprentissage des sous-lexiques*

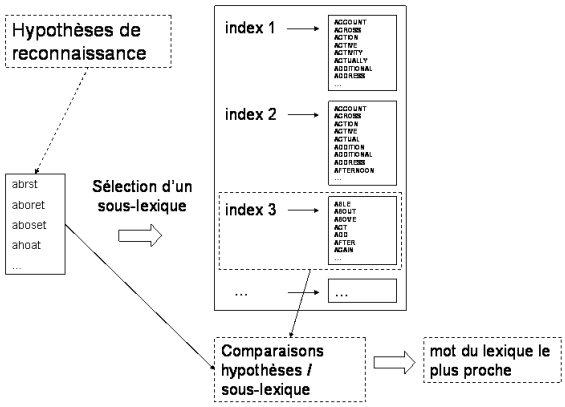

FIG. 3 – *Exploitation des connaissances lexicales*

Nous détaillons ces deux phases, organisation du lexique et exploitation pour le post-traitement dans les paragraphes 3 et 4.

3 Modélisation des connaissances lexicales

3.1 Introduction

Lors du post-traitement, des caractéristiques sont extraites des hypothèses de reconnaissance et comparées aux index des sous-lexiques de façon à sélectionner un ou plusieurs sous-lexiques pertinents.

Nous utilisons des caractéristiques *globales* aux mots pour générer des sous-lexiques en fonction de leur silhouette.

Ces caractéristiques peuvent être considérées comme des informations orthogonales à celles produites par le processus de reconnaissance analytique [13]. Ces indices globaux extraits de la forme du tracé indépendamment de la reconnaissance des lettres sont les signes diacritiques, des informations sur la silhouette et une estimation de la longueur du mot (au sens du nombre de traits descendants). Les caractéristiques globales utilisées pour caractériser les silhouettes des mots sont basées sur les traits descendants les plus pertinents (Figure 4), qui font partie des éléments les plus invariants et les plus robustes de l'écriture manuscrite [3]. Nous utilisons pour définir ces silhouettes quatre sortes de traits descendants:

- traits descendants haut (ex : un dans la lettre ℓ)
- traits descendants bas (ex : un dans la lettre g)
- traits descendants long (ex : un dans la lettre f)
- traits descendants médian (ex : deux dans la lettre a)

FIG. 4 – *Traits descendants dans le mot blue*

Le fait d'utiliser ces informations complémentaires peut permettre d'améliorer le taux de reconnaissance, en corrigeant des erreurs de reconnaissance (confusion de lettres ou groupe de lettres).

3.2 Organisation du lexique et réduction

Dans le cadre d'un post-traitement avec lexique, la réduction consiste à sélectionner un sous-lexique pertinent à l'aide de différents critères. L'objectif est de gagner du temps en n'explorant qu'une sous-partie du dictionnaire, et la problématique est de savoir comment délimiter cette sous-partie. La réduction peut être dynamique (les sous-lexiques sont générés pendant la phase de post-traitement suivant les caractéristiques des hypothèses [9]) ou statique (les sous-lexiques sont déterminés *a priori* et sont indexés par une caractéristique commune). Nous avons opté pour des réductions statiques qui sont moins coûteuses en temps de calcul que des réductions dynamiques. Le but est de générer les sous-lexiques les plus réduits, tout en gardant un critère d'accès robuste : le temps de post-traitement décroît quand la taille des sous-lexiques décroît, mais le risque de sélectionner le *mauvais* sous-lexique est plus élevé.
Nous présentons par la suite deux approches d'organisation du lexique. Dans la première approche, les mots sont regroupés d'après leur similarité visuelle et leur taille. L'étape de réduction consiste alors à sélectionner les sous-lexiques ayant exactement la même silhouette générique et une taille proche de celles des hypothèses de reconnaissance.
Dans la deuxième approche, des vecteurs de caractéristiques sont extraits des mot et regroupés en partitions floues (indexées par leurs centres ou vecteurs *moyens*) à l'aide d'un algorithme de classification non supervisé. La réduction s'effectue par le calcul des distances entre les vecteurs des hypothèses et les index des sous-lexiques : les sous-lexiques les plus proches sont sélectionnés. La comparaison des hypothèses et des mots des sous-lexiques sélectionnés permet de déterminer le mot le plus cohérent.

Classification directe des mots à partir de leur silhouettes génériques.
L'idée principale de cette représentation est de regrouper les mots ayant des silhouettes et des tailles proches dans les mêmes sous-lexiques. Les traits descendants haut, bas et long sont des traits que l'on qualifie de proéminents. Dans l'écriture manuscrite, surtout si elle est dégradée, les traits proéminents sont plus robustes (plus facilement détectables) que les traits médians. Les silhouettes des mots sont donc compressées afin d'être plus stables : tous les traits proéminents sont représentés, mais les traits médians consécutifs sont remplacés par un seul. Cette étape permet d'obtenir la silhouette dite *générique* finale d'un mot (Figure 5). Suivant le style de l'écriture (scripte, cursive ou mixte) certains mots peuvent appartenir à plusieurs sous-lexiques (comme *blue* dans la figure 5)

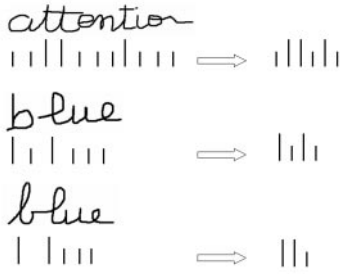

FIG. 5 – *Silhouettes génériques*

Pour compenser la perte d'information produite par la méthode de compression, la longueur du mot (basée sur une approximation du nombre de traits descendants) est ajoutée pour caractériser la silhouette. Ces deux caractéristiques globales permettent d'organiser le lexique en sous-lexiques. Les mots visuellement proches (*ie* : ayant la même silhouette générique et des longueurs proches) sont regroupés dans les même sous-lexiques indexés par leur silhouette générique. La classification est donc directement basée sur l'index représenté par la silhouette générique et la longueur du mot (Figure 6). Un index est représentée par un couple $(S,[l_{min},l_{max}])$ et permet de regrouper dans un sous-lexique tous les mots ayant la silhouette générique S, et dont le nombre de traits est compris entre l_{min} et l_{max}. Un certain chevauchement des sous-lexiques est introduit par le choix des intervalles de taille pour rendre plus robuste l'étape de sélection.

Classification non supervisée de mots à partir d'un vecteur de caractéristiques mots.
La seconde approche proposée pour regrouper les mots vi-

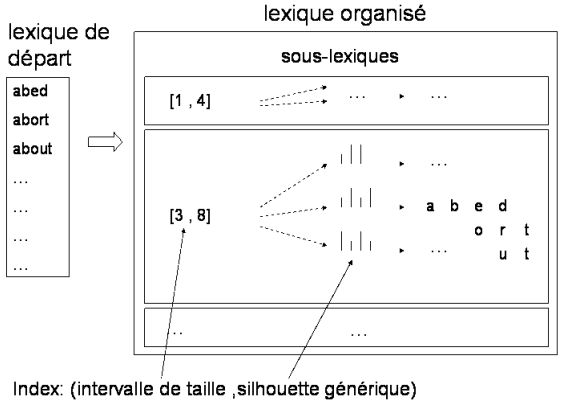

FIG. 6 – *Organisation du lexique par taille et silhouette générique*

suellement proches est d'utiliser un algorithme de classification floue non supervisée. Les mots sont représentés sous forme de vecteurs de caractéristiques à n dimensions, qui comme pour les silhouettes génériques reposent sur des informations les plus invariantes possibles au style de l'écriture. Nous avons choisi de définir les caractéristiques à partir d'éléments de la silhouette des mots : comme les nombres de traits descendants et des informations sur les positions relatives de certains traits proéminents dans le mot.

Nous utilisons l'algorithme de classification non supervisé des FCM (fuzzy c-mean) [8, 4] pour regrouper les mots ayant des caractéristiques proches dans cet espace à n dimensions et indexer ces regroupements par leur centre (ou prototype).

Cet algorithme est basé sur un processus itératif qui vise à produire un partitionnement flou des individus (des vecteurs de caractéristiques de mots) en fonction de L regroupements naturels. L'algorithme est déduit de la minimisation de la fonction objectif suivante :

$$J_{P,U,E} = \sum_{l=1}^{L}\sum_{j=1}^{N}(u_{lj})^m d^2(e_j,P^l). \qquad (1)$$

où $E = \{e_j | j = 1,\ldots,N\}$ est l'ensemble des individus de la base d'apprentissage. $P = \{P^l | l = 1,\ldots,L\}$ est l'ensemble des centres correspondant aux L regroupements recherchés. $U = [u_{lj}]$ est une matrice $L \times N$ dont chaque élément u_{lj} représente l'appartenance de e_j au regroupement l. $m > 1$ est le degré de flou de la partition. $d(e_j,P^l)$ est la distance entre e_j et P^l.

Cette fonction objectif impose la minimisation de la distance entre les individus e_j et les centres P^l. La mise à jour des centres et de la matrice U à chaque pas de l'itération est définie par :

$$P^l = \frac{\sum_{j=1}^{N}(u_{lj})^m e_j}{\sum_{j=1}^{N}(u_{lj})^m}, \qquad (2)$$

et :

$$\left.\begin{array}{ll} u_{lj} = \dfrac{1}{\sum_{k=1}^{L}\left(\frac{d^2(e_j,P^l)}{d^2(e_j,P^k)}\right)^{\frac{1}{m-1}}} & \text{si } L_j = \emptyset \\ \left.\begin{array}{ll} u_{lj} = 0 & \text{si } l \notin L_j \\ \sum_{l\in L_j} u_{lj} = 1 & \text{si } l \in L_j \end{array}\right\} & \text{si } L_j \neq \emptyset \end{array}\right\} \qquad (3)$$

avec $L_j = \{l | 1 \leq l \leq L, d^2(e_j,P^l) = 0\}$.

À partir des regroupements naturels flous, des sous-lexiques indexés par leur centres (ou prototype) sont créés. Les mots sont répartis dans un ou plusieurs sous-lexiques en fonction de leur degré d'appartenance. Ceci peut donc introduire de la redondance dans les sous-lexiques, notamment pour les mots dont la forme globale serait proche de plusieurs prototypes.

Plusieurs paramètres permettent de faire varier les performances de cette organisation :

- les choix *a priori* du nombre de regroupements influent sur la taille des sous-lexiques et donc sur le temps de post-traitement et sa qualité ;
- le choix des caractéristiques utilisées, c'est à dire de l'espace de représentation des mots est aussi un facteur important qui va influer sur la qualité de l'organisation.

3.3 Représentation compacte des sous-lexiques

Pour réduire l'encombrement mémoire, chaque sous-lexique est représenté sous forme d'un arbre lexical (ou trie [11]) où chaque nœud est un caractère. Cette factorisation des préfixes permet de compresser les sous-lexiques (Figure 7) mais est également intéressante lors de l'exploitation des connaissances lexicales pour factoriser les calculs de comparaisons entre les hypothèses de reconnaissance et les mots du sous-lexique..

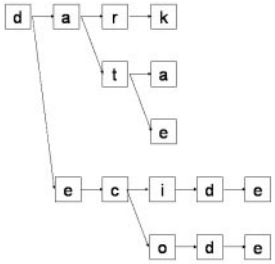

FIG. 7 – *Organisation du lexique : silhouette générique et sous-lexique correspondant*

4 Exploitation des connaissances lexicales

Le post-traitement lexical est réalisé en deux étapes :

- réduction du lexique qui permet de sélectionner un sous-lexique pertinent (étape **1** dans la fig. 8) ;

– comparaison des hypothèses de reconnaissance et des mots du sous-lexique avec une distance d'édition de façon à déterminer le mot du lexique le plus proche du mot manuscrit (étape **2** dans la fig. 8).

Le post-traitement lexical se base sur la liste des hypothèses de reconnaissance. Ces hypothèses, ordonnées par vraisemblance sont déduites du parcours complet du graphe de segmentation (Figure 7). Chaque hypothèse est composée d'une chaîne de caractères et de caractéristiques globales comme des informations sur la silhouette des lettres et sur les signes diacritiques. Nous présentons ici des stratégies de sélection du lexique (paragraphe 4.1) suivant les deux approches d'organisation présentées dans le paragraphe 3, puis la distance d'édition mise en oeuvre (paragraphe 4.2) et enfin des stratégies de comparaisons hypothèses / sous-lexique (paragraphe 4.3.

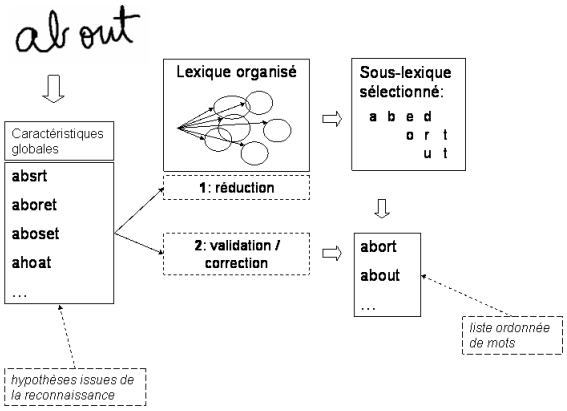

FIG. 8 – *Étapes du post-traitement lexical*

4.1 Stratégies de sélection du sous-lexique

Lors de la sélection du sous-lexique, la stratégie dépend essentiellement de l'organisation de dictionnaire choisie. Avec l'organisation utilisant les silhouettes génériques, la sélection s'effectue par comparaison exacte des couples (silhouettes génériques , taille) des n premières hypothèses avec ceux des index des sous-lexiques. Le nombre de sous-lexiques sélectionnés est directement lié aux nombre n d'hypothèses utilisées pour la reconnaissance (car plusieurs silhouettes peuvent être proposées d'après les hypothèses) et influe donc sur le temps et le taux de reconnaissance.

Avec l'organisation utilisant la classification non supervisée FCM sur les vecteurs de mots, les vecteurs caractéristiques des n premières hypothèses sont calculés et comparés aux prototypes des sous-lexiques avec la même distance que celle utilisée dans l'algorithme de classification : les sous-lexiques les plus proches sont sélectionnés. Le nombre de sous-lexiques sélectionnés dépend donc du nombre d'hypothèses utilisées, mais aussi d'un seuil ε défini de la façon suivante :

$$\frac{d_{min}(i)}{d(i,j)} \leq \varepsilon$$

avec $d(i,j)$ est la distance entre le vecteur de l'hypothèse i et le prototype j, et $d_{min}(i)$ est la distance entre le vecteur de l'hypothèse i et son prototype le plus proche. Ce seuil permet d'adapter le nombre de sous-lexiques sélectionnés en fonction des vecteurs de caractéristiques des hypothèses : si ces vecteurs sont très proches d'un seul prototype, un seul sous-lexique est sélectionné, par contre si les vecteurs sont assez proches d'autres prototypes d'après le seuil ε plusieurs sont sélectionnés.

4.2 Mise en oeuvre d'une distance d'édition

Après avoir sélectionné un ou plusieurs lexique(s) l'étape suivante consiste à effectuer des comparaisons entre les hypothèses et les mots des sous-lexiques de façon à déterminer le meilleur mot : c'est à dire celui qui a la distance minimale avec les hypothèses.

Une distance d'édition permet de comparer deux chaînes de caractères : la distance entre deux chaînes correspond au coût minimal (en termes de transformation ou d'opération d'éditions sur des caractères) du passage de l'une à l'autre.

Principe de la distance d'édition.

Plusieurs distances peuvent être utilisées pour corriger et valider les hypothèses. La première, la distance de Damerau et Levenshtein[12, 6] (DLM_0) 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 [14] ont défini une seconde distance (DLM_1) étendue de la distance de Damereau 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 $X = x_1...x_i...x_n$ et $Y = y_1...y_j...y_m$ 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 statiques des opérations d'édition : substitution, suppression, insertion, division, fusion et substitution de paires.

Optimisations effectuées.

Nous avons optimisé cette distance étendue en introduisant des coûts dynamiques en rapport avec des caractéristiques globales des lettres (DLM_2). Ces coûts dynamiques sont définis suivant un facteur de pénalisation lorsque les caractéristiques de base des lettres ou les signes diacritiques diffèrent.

Par exemple la suppression d'un trait descendant proéminent a un coût plus élevé que la suppression d'un trait mé-

dian et la substitution de deux caractères contenant les mêmes traits descendants a un coût plus faible que celle de deux caractères ayant des traits descendants différents.
Un coefficient de pénalisation C_p est utilisé pour calculer les coûts d'opérations avec pénalisation :

$$\begin{array}{ll} c_{Ins-pen}(x) & = c_{Ins}(x) + C_p \times (\#s(x) - 1) \\ c_{Del-pen}(x) & = c_{Del}(x) + C_p \times (\#s(x) - 1) \\ c_{Sub-pen}(x,y) & = c_{Sub}(x,y) + C_p \times (nbDif(x,y)) \\ c_{Fus-pen}(x,y_1y_2) & = c_{Fus}(x,y_1y_2) + C_p \times (\neq (x,y_1y_2) \\ c_{Div-pen}(x_1x_2,y) & = c_{Div}(x_1x_2,y) + C_p \times (\neq (x_1x_2,y) \\ c_{PSub-pen}(x_1x_2,y_1y_2) & = c_{PSub}(x_1x_2,y_1y_2) \end{array}$$

où $\#s(x)$ est le nombre de traits descendants du caractère x; $\neq (x,y)$ est défini par $|\#s(x) - \#s(y)|$ et $nbDif(x,y)$ est le nombre de différences (en terme de traits descendants) entre x et y. c_{ins}, c_{Del}, c_{Sub}, c_{Fus}, c_{Div} et c_{PSub} sont les coûts statiques définis pour compenser les erreurs de reconnaissance et de segmentation. Il est intéressant de noter que ces coûts peuvent être estimés automatiquement à partir de la matrice de confusion du reconnaisseur de lettres.
De plus, une pénalisation globale se basant sur les signes diacritiques est introduite. Cette pénalisation C_{diac} (initialisée à 0) est définie sur les chaînes hypothèses complètes, ce qui correspond à une information globale, c'est à dire que les signes diacritiques sont déterminés à partir de la forme globale du mot, et non des hypothèses issues du processus de reconnaissance analytique :

$$C_{diac} += \begin{cases} coef_d \times (n_d - m_d) & if \quad m_d < n_d \\ coef_d \times (n_b - m_b) & if \quad m_b < n_b \end{cases}$$

où n_d et n_b sont utilisées respectivement pour représenter le nombre de points (associés aux lettres i et j) et le nombre de barres (associés à la lettre t) décelés dans le mot manuscrit. m_d et m_b sont les nombres de signes diacritiques du mot du lexique ; et $coef_d$ est une valeur de pénalisation fixée. Il y a pénalisation uniquement lorsque plus de signes sont présents dans le mot manuscrit que dans le mot du lexique. Cette information sur les signes diacritiques est robuste et pertinente : c'est à dire que le nombre et la nature des signes diacritiques détectés dans le mots manuscrit doivent forcément être présents dans le mot du lexique le plus proche, mais certains peuvent ne pas avoir été détectés. Cette pénalisation permet de compenser des erreurs dues à l'approche analytique.
Nous avons introduit un seuil dynamique pour élaguer le calcul de la matrice d'édition (Figure 9) permet d'optimiser le temps de calcul de la distance :

- la zone en dehors de la diagonale est élaguée suivant la longueur des chaînes à comparer,
- le calcul est arrêté si la distance intermédiaire est supérieure à un seuil (les deux chaînes de caractères sont trop différentes)

De plus l'organisation des sous-lexiques sous forme d'arbres lexicaux a un impact sur le calcul de la distance car la factorisation des préfixes permet de diminuer le nombre global de comparaisons de caractères. De la même façon des branches de l'arbre lexical peuvent être élaguées si le seuil est atteint.

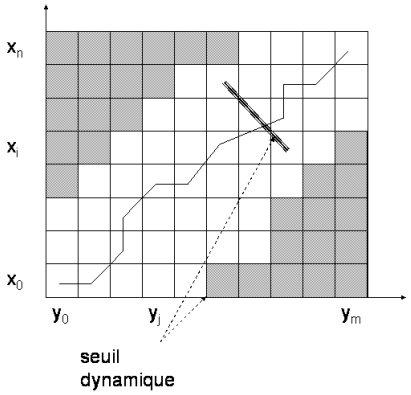

FIG. 9 – *Élagage de la matrice d'édition*

4.3 Stratégies de comparaison hypothèses / sous-lexique

Les hypothèses de reconnaissance (séquences de lettres résultant de l'exploration du graphe de segmentation), comparées aux mots du lexique réduit permettent de déterminer le mot du lexique le plus proche. Plusieurs stratégies peuvent être définies en se basant notamment sur la profondeur d'exploration du graphe de segmentation.
Par exemple, pour réduire le temps de calcul, seules les premières hypothèses (les plus pertinentes) peuvent être utilisées. Dans ce cas l'utilisation d'une distance d'édition adaptée à l'écriture manuscrite ($DLM2$) est indispensable pour compenser les erreurs potentielles de segmentation.
Au contraire, on peut envisager de prendre en compte toutes les hypothèses produites par l'exploration du graphe de segmentation. Ceci peut engendrer un temps de calcul important si la distance utilisée est trop complexe.
Combiner deux distances d'édition permet de trouver un bon compromis entre taux de reconnaissance et temps de calcul. Une stratégie proposée consiste à utiliser la distance modifiée ($DLM2$) (présentée dans 4.2) pour comparer les premières hypothèses au lexique réduit, et une distance simple réduite (SD) pour effectuer les comparaisons des hypothèses suivantes. La distance simple réduite n'autorise qu'une seule opération d'édition (la substitution) pour explorer rapidement les hypothèses correctes (d'après le lexique) dans la suite de la liste sans dégradation importante.
Dans la partie suivante, nous reportons et comparons les intérêts de ces stratégies pour les deux organisations du lexique présentées, sur la base de plusieurs lexiques de tailles différentes.

5 Expérimentations et résultats

Le but de ces expérimentations est de comparer les différents éléments de post-traitement lexical présentés dans cet article par rapport au taux de reconnaissance, au coûts de calcul et à l'encombrement mémoire. Nous avons mené les

test sur une base de 2 000 mots manuscrits de langue anglaise. Ces mots correspondent aux 1 000 mots les plus utilisés de la langue anglaise écrits par deux scripteurs francophones. Les tests ont porté sur des lexiques de 1 000, 7 000 et 25 000 mots, chacun comprenant au moins les mots des bases manuscrites. L'ensemble des mots a été saisi en lettres minuscules, sans autre contrainte pour les scripteurs. Les tests ont été réalisés avec des bases de scripteurs n'ayant pas participé à l'apprentissage du système. Nous présentons d'abord l'influence de la distance d'édition sur le temps de calcul et le taux de reconnaissance. Ensuite nous présentons les résultats obtenus pour les deux approches proposées, avant de les comparer.

5.1 Influence de la distance d'édition

La distance d'édition permet lors du post-traitement de corriger les hypothèses de reconnaissance, il est donc important qu'elle soit adaptée aux problèmes de correction spécifiques de l'écriture manuscrite. Les résultats de la table 1 permettent de comparer les taux de reconnaissances obtenus avec les distances DLM_0, DLM_1 et DLM_2. Ces résultats illustrent les apports de la distance DLM_1 qui prend en compte les erreurs de segmentation. Les résultats avec la distance DLM_2 montrent que l'utilisation d'informations globales (silhouettes et signes diacritiques) permet de corriger plus d'hypothèses que les autres distances présentées : ces informations sont complémentaires de celles issues du processus analytique de reconnaissance (cd qui correspond à une diminution de 11% du taux d'erreur par rapport à l'utilisation de la distance DLM_0).

TAB. 1 – *Influence de la distance d'édition sur le taux de reconnaissance pour les lexiques de 1 000 et 25 000 mots*

1 000 mots	25 000 mots	distance
89.0%	78.1%	DLM_0
89.2%	78.7%	DLM_1
89.8%	80.6%	DLM_2

Les résultats de la table 2 permettent de comparer les performances de différentes stratégies d'exploration du graphe de segmentation pour la distance DLM_2. Les distances utilisées et le nombre d'hypothèses de reconnaissance sont indiqués dans la colonne *stratégie*. Les taux de reconnaissance et les temps de post-traitement augmentent si l'on explore le graphe de segmentation en profondeur. Un bon compromis entre taux et temps de calcul est obtenu en combinant : la distance DLM_2 (comparaison des 20 premières hypothèses avec le sous-lexique réduit) avec la distance SD (comparaison des 10 hypothèses suivantes).

Toutes les expérimentations présentées dans la suite de cet article sont basées sur la stratégie qui consiste à combiner les distance $DLM2$ et SD.

TAB. 2 – *Influence de la stratégie (combinaison de distances) pour le lexique de 25 000 mots*

taux de reconnaissance	temps	stratégie
80.6%	1.6 s	DLM_2 (30)
50.0%	0.1 s	DLM_2 (1)
80.0%	1.1 s	DLM_2 (20) + SD (10)

5.2 Expérimentations pour l'approche par classification de silhouettes génériques de mots

Nous présentons ici les résultats dans le cadre d'un post-traitement avec 25 000 mots (table 3). Assez peu de paramètres peuvent être modifiés dans cette approche, car les index des sous-lexiques et la méthode de sélection sont basées sur des comparaisons exactes. Le taux de sélections correctes du sous-lexique est de 98%, c'est à dire qu'avec cette approche, le mot à reconnaître se trouve dans le bon sous-lexique sélectionné 98 fois sur 100.

TAB. 3 – *Taux de reconnaissance avec un lexique de 25 000 mots utilisé pour le post-traitement lexical (mot correct dans les n premières propositions)*

n	taux de reconnaissance
1	80.0%
2	85.9%
3	88.5%
4	90.5%

5.3 Expérimentations pour l'approche par classification de vecteurs

Les résultats fournis ici sont basés sur un post-traitement avec un lexique de 25 000 mots. La figure 10 montre le nombre de regroupements influe directement sur la qualité de l'organisation du lexique : plus le nombre de regroupement est important, plus le post-traitement est rapide (les sous-lexiques formés sont plus petits). Les taux de reconnaissance évoluent différemment : sur les tests que nous avons effectués le taux maximal (78%) est atteint pour 250 regroupements, et il y a une dégradation pour des valeurs supérieures due à des erreurs de sélection du sous-lexique.

Dans la figure 11 nous présentons l'impact des variations du seuil ε (défini dans le paragraphe 4.1) qui permet d'adapter aux hypothèses le nombre de sous-lexiques à sélectionner. Ce paramètre a une influence sur le nombre de sous-lexiques sélectionnés et donc sur le temps de calcul du post-traitement. Plus ce seuil est proche de 1, moins il y a de sous-lexiques sélectionnés. Son étude permet de déterminer sa valeur optimale (ε = 0.95 pour un taux reconnaissance de 80%) pour un lexique donné. Les résultats de la figure 11 sont donnés pour un post-traitement avec un lexique de 25 000 et 250 regroupements.

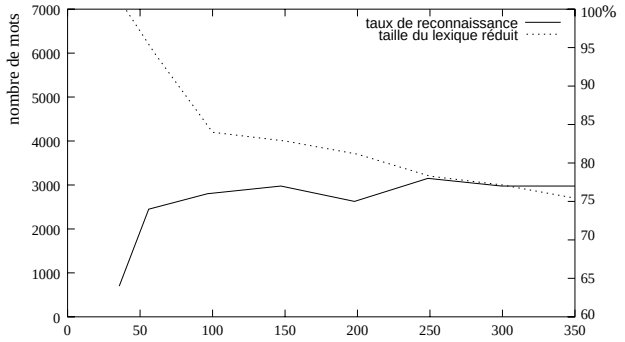

FIG. 10 – *Influence du nombre de regroupements sur les taux de reconnaissance et la taille moyenne des sous-lexiques*

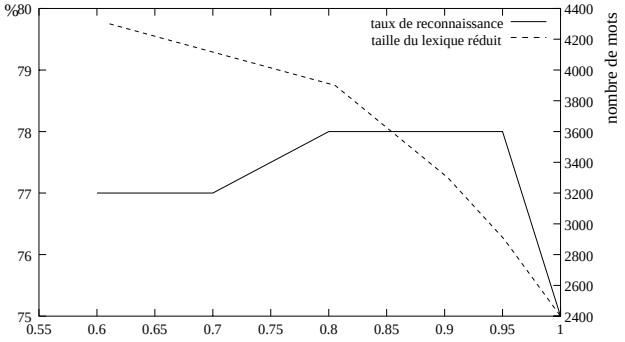

FIG. 11 – *Influence du seuil ε sur les taux de reconnaissance et la taille moyenne des sous-lexiques*

5.4 Comparaison des deux approches

Nous comparons les deux approches d'après les objectifs de départ : encombrement mémoire, taux de reconnaissance et temps de calcul. Pour l'approche par classification des vecteurs de mots, nous avons choisi le nombre de regroupements et le seuil donnant les meilleurs résultats en terme de taux et de temps de calcul (*250* regroupements et ε = 0.95)

La table 4 permet de comparer les tailles mémoires (en nombre de caractères) des lexiques en fonction de leur organisation et du nombre de mots. Le terme *liste* désigne le lexique sans organisation spécifique qui sert de référence. Pour les deux approches, on note une augmentation de la place mémoire occupée par le lexique, due au stockage des index et à l'introduction de redondance dans les regroupements. Cependant cette augmentation est plus importante pour l'approche utilisant les silhouettes génériques car plus de redondance est introduite pendant la phase d'apprentissage des sous-lexiques. Dans le cas de l'approche avec silhouettes génériques, le taux moyen de redondance est de 1.9, alors qu'il est de 1.3 pour la deuxième approche.Dans les deux cas l'augmentation relative est plus faible avec un grand qu'avec un petit lexique : la représentation sous forme d'arbre lexical des sous-lexiques génère une compression d'autant plus intéressante que le lexique est grand.

TAB. 4 – *Encombrement mémoire (nombre de caractères) suivant la taille du lexique (nombre de mots) et l'organisation*

	1 000	7 000	25 000
liste	7 310	51 828	207 732
sil. génériques	22 034	139 419	451 238
classif. vecteurs	10 060	73 368	276 418

Le taux de sélection correcte du sous-lexique est un critère important permettant d'évaluer la qualité d'une organisation : pour l'approche avec silhouette générique il s'élève à 98% et pour l'approche par classification de vecteurs à 96%. Cette différence explique les écarts que l'on observe dans les taux de reconnaissance : 80% pour la première et 78% pour la deuxième (pour un post-traitement avec un lexique de 25 000 mots).

Le dernier point de comparaison concerne le temps de post-traitement qui est proportionnel à la taille des sous-lexiques sélectionnés lors de l'étape de réduction. Pour l'approche avec silhouettes génériques et toujours dans le cadre d'un post-traitement avec un lexique de 25 000 mots, la taille moyenne est de 1 000 mots alors qu'elle est de 3 000 mots pour l'autre approche.

L'approche par silhouettes génériques est plus performante en terme de temps de post-traitement et de taux de reconnaissance, mais cela au détriment de l'encombrement mémoire.

L'approche par classification de vecteurs de mots permet d'avoir un encombrement mémoire réduit mais engendre une augmentation des temps de calcul ainsi qu'une légère baisse des taux de sélection. Cependant les possibilités d'optimisations de cette méthode de réduction sont importantes et nous laissent penser qu'elle constitue une approche particulièrement intéressante pour arriver à un bon compromis en terme de : temps de calcul, occupation mémoire et taux de reconnaissance.

6 Conclusion

Nous avons présenté dans cet article deux approches de post-traitement lexical pour la reconnaissance de mots isolés manuscrits se basant sur deux organisations différentes du lexique pour la réduction. Les résultats expérimentaux ont illustré l'intérêt de l'étude de l'optimisation du post-traitement lexical notamment lors du traitement de lexiques de grandes tailles. Deux approches d'organisation lexicale ont été présentées, ce qui a permis de mettre en relief pour la première (silhouettes génériques) un gain important sur les temps de calcul et pour la seconde (classification non supervisée de mots) un encombrement mémoire très réduit. La seconde approche offre des possibilités d'optimisation qui nous incitent à poursuivre son étude, notamment en travaillant sur le choix d'un meilleur espace de représentation des mots.

La prochaine étape de ce travail consistera à finaliser RESIFMot ce qui nous permettra d'évaluer le système dans

sa globalité par rapport à d'autres approches de reconnaissance.
À plus long terme nous prolongerons ces travaux de recherche en étudiant la combinaison d'un post-traitement lexical basé sur un lexique avec des informations de nature statistique (fréquence d'apparition de n-grammes, de mots), voire des informations de niveau linguistique plus élevé.

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

[1] E Anquetil and H Bouchereau. Integration of an online handwritting recognition system in a smart phone device. In *International Conference on Partern Recognition*, volume 3, pages 192–196, Quebec, Canada, August 2002.

[2] E. Anquetil and G. Lorette. On-line handwriting character recognition system based on hierarchical qualitative fuzzy modeling. In *Fifth International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, pages 47–52, Colchester, Angleterre, September 1996.

[3] E. Anquetil and G. Lorette. Perceptual model of handwritting drawing application to the handwriting segmentation problem. In *International Conference on Document Analysis and Recognition*, volume 1, pages 112–117, Ulm, Germany, August 1997.

[4] James C. Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Plenum Press, 1981.

[5] V.J. Brown, V.J. Della Pietra, R.L. Mercer, S.A. Della Pietra, and J.C. Lai. An estimate of an upper bound for the entropy of english. *Computational Linguistics*, 1992.

[6] F. Damereau. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 7:659–664, 1964.

[7] A. Dengel, R. Hoch, F. Hönes, T. Jäger, M. Malburg, and A. Weigel. Techniques for improving OCR results. In *Handbook on Character Recognition and Document Image Analysis*, chapter 8. World Scientific Publishing Company, 1997.

[8] J C. Dunn. A fuzzy relative of the isodata process and its use in detecting compact, well separated clusters. *Journal of Cybernetics*, 3:32–57, 1974.

[9] M. Gilloux. Réduction dynamique du lexique par la méthode tabou. In *CIFED*, pages 24–31, Quebec, Canada, May 1998.

[10] I. Guyon and F. Pereira. Design of a linguistic postprocessor using variable memory length markov models. In *International Conference on Document Analysis and Recognition*, pages 454–457, Montreal, Canada, August 1995.

[11] D.E. Knuth. The art of computer programming, 1968. Addison-Wesley Publishing Company.

[12] Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 10:707–710, 1966.

[13] S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 23(2):149–164, February 2001.

[14] G. Seni, V. Kripasundar, and R.K. Srihari. Generalizing edit distance for handwritten text recognition. In *Proceedings of SPIE/IS&T Conference on Document Recognition*, pages 54–65, San Jose, CA, February 1995.

[15] G. Seni, R.K. Srihari, and N. Nasrabadi. Large vocabulary recognition of on-line handwritten cursive words. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 18(7), July 1996.