Reconnaissance de courriers manuscrits par HMMs contextuels et modèle de langage

Olivier Morillot, Emmanuèle Grosicki, Laurence Likforman-Sulem

Reconnaissance de courriers manuscrits par HMMs contextuels et modèle de langage

Olivier Morillot, Emmanuèle Grosicki, Laurence Likforman-Sulem

LTCI Télécom ParisTech
37-39, rue Dareau
75014 Paris
morillot@telecom-paristech.fr
likforman@telecom-paristech.fr

DGA Ingénierie des Projets
7-9, rue des Mathurins
92220 Bagneux
emmanuele.grosicki@dga.defense.gouv.fr

RÉSUMÉ. *Cet article décrit une approche globale de reconnaissance de lignes manuscrites, soumis par les auteurs lors de la compétition ICDAR 2011 sur la reconnaissance de courriers en Français. L'intérêt de l'approche proposée est qu'elle permet d'étendre un système de reconnaissance de mots isolés à base de HMM contextuels à un système de reconnaissance de lignes sans avoir à réaliser une segmentation explicite des lignes en mots. Les auteurs présentent ainsi une solution complète, adaptée à un système de reconnaissance de courriers qui comprend un prétraitement original des lignes, un modèle de Markov caché contextuel ainsi qu'un modèle de langage optimisé fondé sur des bigrammes de mots. Les performances obtenues lors de la compétition (73,2% de mots correctement reconnus) sont très encourageantes car comparables aux performances de l'état de l'art d'un système de reconnaissance de mots isolés avec dictionnaire basé sur des HMMs.*

ABSTRACT. *This article describes a global approach for the recognition of handwritten lines which was submitted at the ICDAR 2011 French handwritten mail recognition competition. Its main contribution is to extend the context-dependent HMMs word recognition system to the line level without segmentation into words. Thus, the authors describes a complete solution for mail recognition: original preprocessing, context-dependent HMMs and an optimized bigram language model. Obtained results during the competition (73,2% of words correctly recognized) are very encouraging since they are similar to state-of-art recognition performances obtained with HMMs on single words.*

MOTS-CLÉS : *Reconnaissance d'écriture manuscrite, courriers, reconnaissance de lignes de texte, HMMs contextuels, prétraitement, modèle de langage, bigramme.*

KEYWORDS: *Handwritten recognition, mail, text lines recognition, context-dependent HMMs, preprocessing, language model, bigram.*

1. Introduction

Historiquement, l'identification des adresses postales fut une des premières tâches de la reconnaissance d'écriture manuscrite. Un des principaux enjeux actuels du traitement automatique du courrier est l'analyse de son contenu. Les administrations comme les entreprises s'intéressent désormais aux tâches de classification, de recherche, voire de réponse aux quantités de courriers qu'elles reçoivent. Un des principaux enjeux actuels de la reconnaissance est donc le traitement automatique des courriers et la reconnaissance de leur contenu.

Une part importante de la recherche dans le domaine a été consacrée à la classification de documents (Rodríguez-Serrano *et al.,* 2009) grâce à des méthodes de type "keyword-spotting". D'autres efforts se sont concentrés sur la reconnaissance de mots isolés et ont atteint des taux de reconnaissance très élevés (Grosicki *et al.,* 2009).

La reconnaissance de courrier est une tâche très spécifique. Elle se caractérise souvent par un vocabulaire spécifique à un domaine et donc de taille intermédiaire ($\sim$7.000 mots). De plus, le caractère officiel du courrier introduit des phrases idiomatiques telles que les formules de politesse. En revanche, des mots inconnus et des codes apparaissent dans presque chaque courrier. Ces caractéristiques se retrouvent dans la base RIMES[1] de courriers non contraints.

Notre travail s'interesse à une modélisation de l'écriture manuscrite par modèles de Markov cachés (HMMs : Hidden Markov Models) (El-Yacoubi *et al.,* 1999, Plamondon *et al.,* 2000). Une des spécificités de notre travail est qu'il propose une système de reconnaissance de lignes de texte sans segmentation explicite en mots. Face à l'irrégularité des espacements, cette approche permet de s'affranchir des incertitudes inhérentes à la segmentation. Nous proposons également un prétraitement adapté aux lignes qui comprend une méthode originale de correction de la pente locale des lignes. Les caractères sont modélisés comme de séquences d'états HMMs. Afin de prendre en compte leur voisinage, les caractères sont modélisés par des HMMs contextuels (Bianne-Bernard *et al.,* 2011).

Un des objectifs de notre système est également de prendre en compte les propriétés du langage en construisant un modèle de langage statistique qui soit adapté à notre tâche de reconnaissance. Les quelques travaux qui se sont intéressés à la reconnaissance des lignes utilisent tous des modèles de langage : Vinciarelli a notamment proposé un système de reconnaissance de lignes à large vocabulaire à base de HMMs et de modèles de langage (Marti *et al.,* 2001, Vinciarelli *et al.,* 2004). D'autres approches à base d'hybrides HMM-RNN (Espana-Boquera *et al.,* 2011) ou de BLSTM (Graves *et al.,* 2009) ont également été évaluées sur les lignes de texte. À la différence de ces travaux qui utilisent de grands vocabulaires (de 10000 à 50000 mots) et donc de grands corpus pour estimer leur modèle de langage, notre système ne repose que sur les transcriptions de la base d'apprentissage ($\sim$7000 mots). Nous démontrons

1. Reconnaissance et Indexation de données Manuscrites et de fac similés - Recognition and Indexing of handwritten documents and faxes (http ://www.rimes-database.fr/)

ici l'efficacité de petits dictionnaires et modèles de langage pour une tâche spécifique comme la reconnaissance du courrier.

L'article est organisé en six parties : la deuxième partie s'intéresse au prétraitement des lignes de textes et à l'extraction des caractéristiques. Ensuite la modélisation HMM est détaillée dans la troisième partie. La quatrième partie présente en détail la modélisation du langage effectuée. Les différentes expérimentations et résultats sont rassemblés dans la cinquième partie. Enfin les conclusions et perpectives sont données dans la sixième partie.

2. Prétraitement et extraction des caractéristiques

Un des objectifs du prétraitement des images est de réduire la variabilité qui peut exister entre les écritures de différents scripteurs mais également au sein de l'écriture d'un même scripteur. Cette variabilité concerne l'inclinaison des traits, la pente des lignes ou des mots et les dimensions, notamment la hauteur, de l'écriture. Le prétraitement a également pour objectif la suppression des bruits dans l'image. Nous présentons ci-dessous l'ensemble des prétraitements réalisés sur les lignes de texte et dans l'ordre suivant : débruitage, correction d'inclinaison et correction de pente.

2.1. *Débruitage*

Les lignes de texte des courriers sont découpées en imagettes de ligne à partir des coordonnées rectangulaires vérité terrain fournies par la base de données Rimes. Les courriers étant écrits sans guide-ligne, nombreuses sont les lignes penchées ou trop serrées les unes aux autres. Cela a pour effet d'introduire des traits, hampes et jambages, provenant de lignes voisines dans les régions hautes et basses des imagettes de ligne. Ces hampes et jambages correspondent à des composantes connexes excentrées qui perturbent la suite du prétraitement, notamment la correction de la pente des lignes ainsi que l'extraction des lignes de base et des caractéristiques (cf. Section 2.5). Le nombre de ces lignes bruitées n'étant pas négligeable, nous proposons l'approche de débruitage suivante basée sur la classification des composantes connexes de l'image de ligne. Les composantes connexes de la ligne sont classées par un ensemble de règles en trois catégories :

– Jambages de la ligne précédente

– Ligne à conserver

– Hampes de la ligne suivante

Les composantes sont classées comme jambage ou hampe si elles sont en contact avec le bord de l'imagette et si la position de leur centre de gravité est excentrée. Les composantes hampes et jambages identifiées comme du bruit sont soustraites à l'imagette de ligne. Ces critères sont issus d'un compromis entre le taux de fausses alarmes et celui de détection. Cette procédure permet d'éliminer la plupart du bruit

et seuls quelques accents sont parfois considérés comme du bruit. Un résultat de ce débruitage sur une ligne de texte extraite d'un courrier original (corps de texte) est illustré en Figure 2a-c.

2.2. *Binarisation de l'arrière plan*

Le processus de numérisation peut induire un bruit dans l'arrière plan des imagettes de ligne. Pour éliminer cs bruit, l'arrière plan est binarisé tout en conservant la dynamique des niveaux de gris des traits d'écriture qui sera utilisée pour l'extraction des caractéristiques (cf. Section 2.5. Le seuil de binarisation est estimé sur chaque imagette par la méthode d'Otsu et les valeurs de pixels inférieures à ce seuil sont ramenées à zéro tandis que les autres conservent leur valeur de gris.

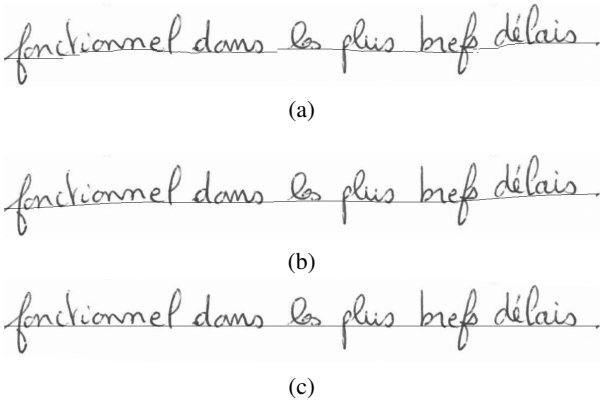
2.3. *Correction de la pente de l'écriture*

La pente des lignes d'écriture (appelée en anglais "slope" ou "skew") n'a pas que pour seul effet de perturber l'extraction des lignes de base. En effet, notre extraction de caractéristiques comprend notamment des caractéristiques de densité estimées dans les trois régions de l'écriture (hampes, zone centrale, jambages). Des lignes d'écriture penchées perturbent donc la définition de ces zones et la valeur des caractéristiques extraites. Cet effet est d'autant plus sensible que l'on travaille directement au niveau des lignes de texte. Cependant une correction globale de la pente est insuffisante puisqu'elle suppose que l'écriture repose sur une droite rectiligne. Pour les systèmes découpant les lignes en mots, une estimation et une correction de la pente par morceaux est possible (Bertolami *et al.,* 2007). Plus récemment, une méthode utilisant une détection de contours puis un classifieur (de type perceptron multicouche) a abordé cette correction locale de pente (Espana-Boquera *et al.,* 2011).

Nous proposons ici une méthode inédite de correction de la pente locale de la ligne d'écriture sans segmentation ni détection des composantes connexes. L'objectif est d'estimer la ligne de base basse, colonne par colonne par une approche à fenêtre glissante afin de la rectifier.

Une fenêtre d'analyse parcourt l'image de ligne de gauche à droite et estime la ligne de base basse. La ligne de base basse est estimée à partir des projections selon la direction horizontale des densités de pixels sur l'axe vertical (Vinciarelli *et al.,* 2001). L'utilisation d'une fenêtre d'analyse suffisament large permet d'avoir un estimateur assez robuste à la présence des jambages ou des silences (Figure 1a).

La courbe représentant les positions de la ligne de base basse sont ensuite lissées par un filtrage gaussien pour en éliminer les discontinuités (Figure 1b). Sans ce lissage, l'image corrigée peut présenter des artefacts de cisaillement vertical. Les tailles des fenêtres d'analyse et de lissage sont proportionnelles à la longueur de l'image afin que



(a)

(b)

(c)

Figure 1 – Processus de correction de la pente des lignes de texte. (a) Estimation locale de la pente. (b) Lissage gaussien des valeurs de la pente. (c) Correction de la pente.

l'approche soit indépendante de la taille de l'écriture et s'adapte aux lignes de texte très courtes.

Nous avons constaté des gains de reconnaissance importants grâce à cette correction de la pente (cf. Section 5).

2.4. *Correction de l'inclinaison de l'écriture*

L'approche HMM par fenêtre glissante est sensible à l'inclinaison de l'écriture. En effet une écriture inclinée peut induire une superposition de traits appartenant à des caractères différents à l'intérieur de la fenêtre verticale. Plutôt que d'extraire les caractéristiques avec une fenêtre penchée (Al-Hajj-Mohamad *et al.,* 2009), nous avons préféré redresser l'écriture. La valeur de l'inclinaison ("slant") est déterminée globalement sur la ligne en cherchant l'orientation maximisant une mesure liée aux densités de pixels noirs dans l'ensemble des colonnes de la ligne (Vinciarelli *et al.,* 2001). L'image est ensuite redressée par une transformation affine.

Néanmoins nous avons constaté que cette correction pouvait être insuffisante pour certaines écritures. L'inclinaison, au même titre que la pente, est une propriété locale et non globale. En effet, l'inclinaison peut varier au sein même d'un mot et corriger une tendance globale peut être néfaste localement. Nous portons donc actuellement nos efforts sur l'estimation et la correction locale de l'inclinaison pour raffiner notre correction.

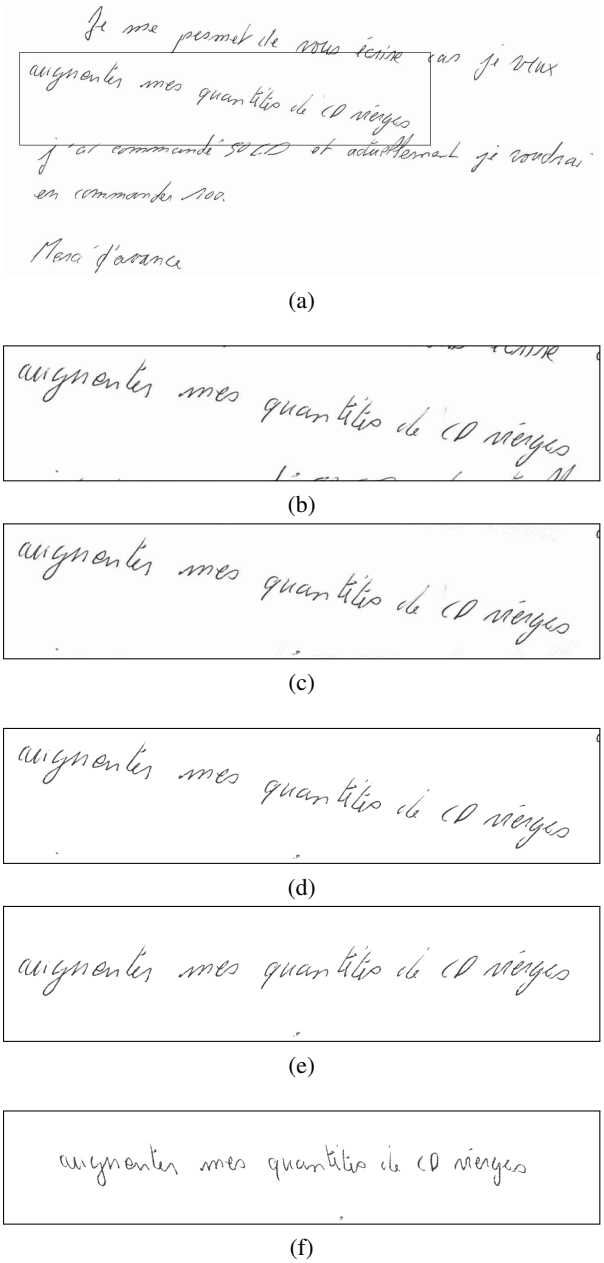


Figure 2 – Principales étapes du prétraitement. (a) Courrier original. (b) Découpage de la ligne. (c) Suppression du bruit périphérique. (d) Binarisation de l'arrière plan. (e) Correction de la pente. (f) Correction de l'inclinaison.

2.5. *Extraction des caractéristiques*

L'extraction des caractéristiques se fait sur les images de lignes prétraitées. Nous utilisons les caractéristiques définies par Al-Hajj et al (Al-Hajj-Mohamad *et al.,* 2005, Al-Hajj-Mohamad *et al.,* 2009)] qui ont montré leur efficacité sur les écritures arabes, françaises et anglaises (Bianne-Bernard *et al.,* 2011). Il s'agit de caractéristiques à la fois statistiques (densités, transitions fond-écriture) et géométriques (convexité locale, positions relatives du centre de gravité et des lignes de bases). L'extraction est réalisée à partir d'une fenêtre glissante avec recouvrement qui s'affranchit des différentes hauteurs d'images en utilisant un nombre fixe de cellules d'analyse. De plus, pour tenir compte de la dynamique des fenêtres glissantes entourant la fenêtre courante, les caractéristiques extraites sont également dérivées. Nous obtenons ainsi un vecteur de caractéristiques de 56 composantes.

3. Modélisation HMM en contexte

3.1. *Modélisation HMM des lignes*

Les lignes sont modélisées à partir de la concaténation des modèles de mots séparés par des espaces (modèle du silence). Les modèles de mots sont quant à eux obtenus par la concaténation des modèles de caractères le composant. Pour disposer d'un modèle de caractères précis, nous avons utilisé une modélisation de ces derniers tenant compte de leur contexte dans le mot. La stratégie d'analyse par fenêtre glissante retenue n'utilise pas de segmentation explicite des lignes en mots. Cette dernière est obtenue implicitement lors du décodage.

Les modèles de caractères sont représentés par une succession d'états émetteurs pour lesquels les transitions s'effectuent de gauche à droite et le saut d'état est autorisé. La densité de probabilité des observations attachée à chaque état est un mélange de N_G distributions gaussiennes obtenues par incrémentations successives grâce à l'algorithme de Baum-Welch. Les paramètres importants à estimer sont le nombre de gaussiennes N_G, le mombre d'états par modèles de caractères ainsi que le nombre de ré-estimations.

3.2. *Modèles contextuels des caractères*

Pour tenir compte des déformations possibles d'un caractère liées à son contexte c'est-à-dire aux lettres qui l'entourent, nous avons opté pour une modélisation contextuelle des caractères. Le mot n'est plus vu comme une succession de caractères indépendants, mais comme une succession de caractères en contexte. Les monographes sont ainsi remplacés par des trigraphes où au caractère central est rajouté son caractère de gauche et de droite (contextes gauche et droite). La figure 3 montre l'exemple de deux trigraphes "t-e+r" et "v-e+n partageant la même lettre centrale. Il n'est cependant pas envisageable de considérer tous les trigraphes possibles car cela augmenterait

Figure 3 – Différentes ligatures du 'e' selon le contexte

considérablement le nombre de paramètres à estimer. On passerait ainsi de 91 monographes (lettres majuscules et minuscules, lettres accentuées, chiffres et caractères spéciaux) à 91^3 trigraphes. Heureusement, tous les trigraphes ne sont pas observables. Par exemple, l'analyse d'un dictionnaire de 11000 mots ne fait apparaître que 9400 trigraphes différents. Il est également possible de réduire le nombre de paramètres à estimer en regroupant certains trigraphes (Bianne-Bernard *et al.,* 2011) .

4. Modèle de langage

Le passage de la reconnaissance du niveau mot au niveau ligne offre la possibilité de modéliser les successions de mots avec un modèle de langage. L'intérêt pour notre système est d'apprendre quelles sont les successions de mots les plus probables afin d'améliorer la reconnaissance faite par le modèle optique. Ainsi, un mot mal reconnu par la seule modélisation HMM pourra être ainsi corrigé grâce à son contexte. Cette approche est utilisée dans la reconnaissance de la parole depuis les années 1980 (Bahl *et al.,* 1983). Ce n'est que plus récemment qu'elle a été transposée à la reconnaissance d'écriture (Vinciarelli *et al.,* 2004). L'estimation de la séquence optimale de mots $\hat{W}$ est donnée :

$$\hat{W} = \arg\max_{W} P(W|X) = \arg\max_{W} P_{optique}(X|W) P_{grammaire}(W)$$

où la vraisemblance $P_{optique}(X|W)$ de la séquence d'observation X pour la suite de mots $W = (w_1, w_2, ..., w_n)$ est calculée par le modèle HMM décrit précédemment et la probabilité $P_{grammaire}(W)$ de la séquence de mots W est calculée grâce au modèle de langage. Celui-ci permet de calculer la probabilité d'obtenir un mot en fonction de son contexte.

4.1. *Modélisation par n-grammes*

Les modèles de langage statistiques par n-grammes (n-grams) sont les plus fréquement utilisés (Katz, 1987) ainsi que leurs variantes telles que les n-grammes par classes (Brown *et al.,* 1992). D'autres approchent récentes utilisent les réseaux de neurones NNLM (Bengio *et al.,* 2001). L'approche par n-grammes réduit le contexte d'un mot aux $n-1$ mots précédents. Ainsi la probabilité d'un mot est définie en fonction de

ces $n-1$ prédécesseurs $w_{i-n+1}, ..., w_{i-1}$: $P(w_i|w_{i-n+1}, ..., w_{i-1})$ Cette probabilité est calculée sur un corpus de textes en dénombrant les successions de mots :

$$\hat{P}(w_i|w_{i-n+1}, ..., w_{i-1}) = \frac{C(w_{i-n+1}, ..., w_i)}{C(w_{i-n+1}, ..., w_{i-1})}$$

avec C(.) la fonction de dénombrement. Le choix de n, la taille du contexte pris en compte, est limité par la taille du corpus d'apprentissage. En effet, plus les séquences de mots considérées sont longues (n grand), plus rares sont leurs apparitions. Pour que leurs estimations de probabilité d'occurence soient significatives, il est donc nécessaire de disposer de très grands corpus. Pour cette raison, la plupart des modèles de langage n-grammes choisissent $n <= 4$. Par exemple, pour un modèle de langage trigramme ($n = 3$) contenant 10000 mots, il est nécessaire d'estimer $10.000^3 = 10^{12}$ probabilités différentes. En pratique, seule une part infime de ces trigrammes apparaissent dans le langage courant et a fortiori dans le corpus. Pour prendre en compte les séquences inédites qui peuvent apparaître dans le décodage, il est possible d'augmenter la taille du corpus d'apprentissage ou bien de limiter la valeur de n. Cependant ces pistes n'assurent pas la modélisation de tous les n-grammes possibles. Une solution plus générale est une stratégie de back-off : si un n-gramme n'est pas observé dans le corpus, on lui attribue une valeur grâce à la probabilité de son préfixe de taille $n-1$, plus susceptible d'apparaître. La masse de probabilité est redistribuée depuis les événements observés vers ceux qui ne l'ont pas été. Cette méthode est appelée lissage des probabilités ("smoothing probabilities" ou "discounting").

4.2. *Construction du modèle de langage*

Nous proposons ici le détail de notre approche pour construire un modèle de langage adapté à la reconnaissance des courriers en Français.

Tout d'abord, l'apprentissage d'un modèle de langage doit être rattachée à la tâche de reconnaissance désirée : plusieurs corpus de grande taille sont disponibles pour la construction de modèles de langage généraux, mais ne correspondent pas forcément au langage utilisé dans un courrier. Par exemple, les phrases commençant par "Je", très présentes dans les courriers, apparaissent très rarement dans les journaux. Aucun des corpus librement disponibles n'était adapté à notre tâche. Nous avons donc choisi de construire notre modèle de langage uniquement à partir des transcriptions des courriers de la base RIMES. La quantité restreinte de textes disponibles pour l'apprentissage nous a donc orientés vers un modèle de bigramme que nous avons construit grâce à la librairie "SRILM Toolkit" (Stolcke, 2002).

L'objectif étant la reconnaissance de lignes, il était préférable d'effectuer l'apprentissage sur les transcriptions des lignes séparément et non de l'intégralité du texte. Cependant pour prendre en compte les bigrammes qui sont situés à la jonction entre de lignes, nous avons effectué un redécoupage du corpus en plaçant les retours à la ligne à des endroits différents (Espana-Boquera *et al.,* 2011). Cette approche permet

également de modéliser un plus grand nombre de débuts de lignes arbitraires. Bien qu'elle ne soit pas prise en compte dans l'évaluation des taux d'erreurs, la ponctuation est modélisée car porteuse d'un sens syntaxique fort.

Écrits sans contrainte, les courriers présentent la particularité d'être assez riches en fautes d'orthographe. Dans cet article, nous nous intéressons uniquement aux erreurs syntaxiques. Dans la base Rimes, ce type d'erreur représente environ 3% des mots du dictionnaire. Dans certains rares cas, ces erreurs peuvent même être aussi fréquentes que la véritable orthographe. Ces erreurs de syntaxe peuvent provoquer deux types d'erreurs de décodage : Si l'erreur apparaît dans le texte d'apprentissage, elle est ajoutée au dictionnaire et peut conduire à ajouter des erreurs aux mots décodés. Si l'erreur apparaît dans le texte à décoder et n'est pas dans le dictionnaire, une erreur de décodage peut intervenir. Nous avons tenté d'autoriser plusieurs orthographes pour un mot en les regroupant sous la même étiquette dans le dictionnaire. Le processus de reconnaissance pourra ainsi sortir le mot correctement orthographié (l'étiquette), voire même corriger des erreurs. Pour ce faire, nous avons ajouté les orthographes correctes au dictionnaire lorsqu'elles manquaient. Le corpus d'apprentissage du LM a aussi été modifié de manière à ce que son vocabulaire corresponde à celui du dictionnaire. Cette procédure a pour effet d'augmenter le nombre d'orthographies possibles (taille du dictionnaire) mais de limiter le nombre de sorties possibles (nombre de monogrammes).

Pour modéliser les bigrammes inédits, nous avons utilisé la méthode de discounting de Good-Turing (Good, 1953). Les probabilités de ces événements inédits $\hat{P}(w_i|w_{i-1})$ sont calculées à partir de la probabilité du monogramme $\hat{P}(w_i)$:

$$\hat{P}(w_i|w_{i-1}) = \alpha(w_{i-1})\hat{P}(w_i) \text{ si } C(w_{i-1}, w_i) = 0$$

avec $\alpha()$ le poids de "back-off". La probabilité $P_{grammaire}$ d'une séquence de mots $w_1, .., w_n$ se décompose comme le produit des probabilités de transition : $P_{grammaire}(w_1, .., w_n) = \prod_{i=1}^{n} P(w_i|w_{i-1})$. La reconnaissance est donc une combinaison du modèle optique HMM et du modèle de langage. Pour pondérer leurs effets, il est possible d'affecter un poids au modèle de langage, le "Grammar Scale Factor" (GSF). Nous en avons optimisé la valeur sur la base de validation (Sec. 5). En résumé, pour une séquence de vecteurs d'observations X donnée, notre modèle calcule la séquence de mots la plus probable $\hat{W}$ ainsi :

$$\hat{W} = \arg\max_{W} P_{optique}(X|W) P_{grammaire}(W)^{GSF}$$

Pour évaluer l'adéquation entre un modèle de langage et le texte à reconnaître, une estimation de perplexité (PP) est classiquement effectuée. Cette mesure peut être vue comme une moyenne du nombre de mots possibles pouvant suivre n'importe quel mot. La perplexité est donc liée à la taille du vocabulaire et son interprétation est limitée : une diminution de sa valeur n'est pas toujours signe d'une meilleure reconnaissance. Dans le cas d'un modèle de bigramme, l'évaluation de la perplexité sur un texte de n mots $(w_1, w_2, ...w_m)$ s'exprime sous la forme :

$$\hat{PP} = 2^{\hat{H}} \text{ où } \hat{H} = \frac{1}{m} \sum_{i=1}^{m} \log P(w_i|w_{i-1})$$

5. Résultats expérimentaux

L'ensemble de nos expériences a été réalisé sur la base française RIMES. Elle fut créée sous l'impulsion des Ministères de la Défense et de la Recherche pour évaluer les systèmes de reconnaissance automatique et d'indexation de courriers manuscrits. Elle a été composée par des volontaires à qui il a été demandé d'écrire des lettres suivant un des neuf scenarii suivants : changement d'information personnelle, demande d'information, ouverture ou fermeture de compte, modification de contrat, commande, plainte, difficultés de paiement, lettre de rappel ou déclaration de sinistre. Depuis sa création en 2006, plusieurs compétitions de reconnaissance de mots ont été organisées grâce à cette base et en 2011 a eu lieu la première compétition de reconnaissance de blocs de mots. La base d'apprentissage comprenait 1500 courriers, soit un total de 11329 lignes. Pour nos expériences, nous avons partagé cette base en une base d'apprentissage de 10318 lignes (1500 courriers) et de validation de 1011 lignes (130 courriers). Nous avons évalué sur cette base de validation l'intérêt de la correction de la pente locale des lignes de texte ainsi que des étapes de construction du modèle de langage. Cette base nous a également permis d'optimiser la valeur du GSF.

Les performances de reconnaissance sont données sous la forme de taux d'erreurs au niveau mot (WER : Word Error Rate) et de taux de mots correctement reconnus (WCR : Word Correct Rate) :

$$WER = \frac{substitutions \ + \ insertions \ + \ suppressions}{nombre\,total\,de\,mots}$$

$$WCR = 1 - \frac{substitutions \ + \ suppressions}{nombre\,total\,de\,mots}$$

L'opération de correction de la pente locale se révèle très utile puisqu'elle conduit à une diminution jusqu'à 5% du taux d'erreurs WER, sur la base de validation, dans le cas d'une reconnaissance sans modèle de langage (voir Tab.1). Ce gain souligne l'intérêt de corriger la pente des lignes. La correction de la pente permet en effet une meilleure extraction des lignes de base et des caractéristiques associées.

Tableau 1 – Impact de la correction de la pente sur la reconnaissance (Décodage de la base de validation sans LM)

Dictionnaire	WER sans deskew	WER avec deskew
apprentissage	51.6%	46.6%
apprentissage et validation	49.6%	44.3%

Concernant la construction du LM, l'expérience montre que l'ajout de lignes redécoupées au corpus a un effet positif sur le taux de reconnaissance (voir Tab.2). La valeur de la perplexité augmente car la recoupe introduit des nouveaux bigrammes.

Tableau 2 – Augmentation artificielle de la taille du corpus (Décodage de la base de validation avec LM, GSF=1)

Corpus	Nombre de lignes	WER	PP
Transcriptions	15172	45.3%	48.0
Transcriptions+recoupes	43766	44.5%	146.9

Nous avons optimisé le poids de la grammaire (GSF) en évaluant les performances de la reconnaissance sur notre base de validation (voir Fig.4). Les modèles avec et sans lissage des probabilités par discounting sont comparés. Dans une première phase, pour des valeurs de GSF faibles (GSF<10) nous pouvons observer que la stratégie avec discounting dégrade les performances. Cette tendance s'explique par le fait que le discounting affaiblit la probabilité des bigrammes correctement appris pour la redistribuer sur les bigrammes inédits. Ainsi la probabilité des bigrammes est trop faible pour avoir un impact suffisant sur la reconnaissance.

Passé une certaine valeur du GSF (GSF>10), le poids donné au LM est suffisant pour que le modèle avec discounting obtienne de meilleurs résultats. Une trop grande valeur de GSF (GSF>25) conduit à une diminution des performances. En effet, en donnant un poids trop important au modèle de langage, le module de reconnaissance finit par imposer les successions de mots les plus probables en dépit de la réalité optique. La valeur optimale choisie sur la base de validation pour le poids de la grammaire est de 25 : elle permet de diminuer le taux WER de 14.2% en valeur absolue par rapport à une reconnaissance sans modèle de langage.

Le raffinement consistant à corriger les erreurs syntaxiques conduit à un gain de 0,3%. Ce gain doit être mis en regard du taux de mots incorrects dans le dictionnaire ($\sim$3%). Les corrections conduisent également à une diminution de la perplexité normalisée (perplexité divisée par le nombre de monogrammes) de 0.029 à 0.022.

Le système évalué lors de la compétition ICDAR a obtenu un taux WER de 31,2% ainsi qu'un taux WRC de 73,2% (Grosicki *et al.,* 2011). Or ce système ne comprenait pas de correction de la ligne de base. Notre nouveau système, évalué sur la base de test ICDAR 2011, atteint un taux WER de 26.2% ainsi qu'un taux WRC de 77.7%. Le gain en valeur absolue du WER est donc de 5%.

6. Conclusions et perspectives

Dans cet article, nous avons présenté le développement d'un système complet de reconnaissance de lignes de textes issues de courriers manuscrits. Celui-ci comprend une modélisation par HMMs contextuels à l'état de l'art. De plus, nous avons développé une nouvelle approche pour la correction de la pente locale des lignes qui

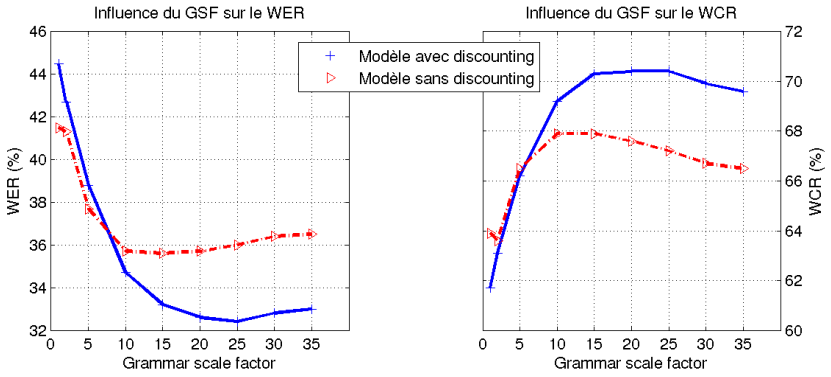


Figure 4 – Optimisation du GSF sur la base de validation

conduit à un gain de 5% en valeur absolue sur le taux WER. Notre travail a mis l'accent sur l'apport des modèles de langage adaptés à des tâches de reconnaissance spécifiques. En effet, notre modèle de langage uniquement construit sur les transcriptions permet d'atteindre un gain de 16,3% en valeur absolue sur le taux WER.

Les développements futurs pourront s'intéresser à définir un traitement dédié pour les champs spéciaux (codes, numéros de téléphones) car ils ne peuvent pas être reconnus avec un dictionnaire à vocabulaire fermé. Concernant le prétraitement, des méthodes d'estimation et de correction de l'inclinaison locale de l'écriture pourraient être mis en œuvre afin d'améliorer le prétraitement des lignes. Enfin, actuellement le modèle de langage ne prend pas encore en compte les accents. Or ceci ont un sens grammatical décisif en Français et seront pris en compte dans notre travail futur.

7. Bibliographie

Al-Hajj-Mohamad R., Likforman-Sulem L., Mokbel C., « Arabic Handwriting Recognition Using Baseline Dependant Features and Hidden Markov Modeling », *Proceedings of the Eighth International Conference on Document Analysis and Recognition - ICDAR05*, p. 893-897, 2005.

Al-Hajj-Mohamad R., Likforman-Sulem L., Mokbel C., « Combining Slanted-Frame Classifiers for Improved HMM-Based Arabic Handwriting Recognition », *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, n° 7, p. 1165-1177, 2009.

Bahl L., Jelinek F., Mercer R., « A Statistical Approach to Continuous Speech Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 5, n° 2, p. 179-190, March, 1983.

Bengio Y., Ducharme R., Vincent P., « A neural probabilistic language model », *Journal of Machine Learning Research*, vol. 3, n° 2, p. 1137-1155, 2001.

Bertolami R., Uchida S., Zimmermann M., Bunke H., « Non-Uniform Slant Correction for Handwritten Text Line Recognition », *Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on*, vol. 1, p. 18 -22, sept., 2007.

Bianne-Bernard A.-L., Menasri F., El-Hajj R., Mokbel C., Kermorvant C., Likforman-Sulem L., « Dynamic and Contextual Information in HMM modeling for Handwritten Word Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

Brown P., DeSouza P., Mercer R., Della-Pietra V., Lai J., « Class-based n-gram models of natural language », *Computational Linguistic*, vol. 18, n° 4, p. 467-479, 1992.

El-Yacoubi A., Gilloux M., Sabourin R., Suen C.-Y., « An HMM-Based approach for off-line unconstrained handwritten modeling and recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, n° 8, p. 752-760, 1999.

Espana-Boquera S., Castro-Bleda M. J., Gorbe-Moya J., Zamora-Martinez F., « Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, p. 767-779, 2011.

Good I., « The Population Frequencies of Species and the Estimation of population parameters », *Biometrika*, vol. 40, p. 237-264, 1953.

Graves A., Liwicki M., Fernandez S., Bertolami R., Bunke H., Schmidhuber J., « A Novel Connectionist System for Unconstrained Handwriting Recognition », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, May, 2009.

Grosicki E., El-Abed H., « ICDAR 2009 Handwriting Recognition Competition », *ICDAR*, p. 1398-1402, 2009.

Grosicki E., El-Abed H., « ICDAR 2011 : French Handwriting Recognition Competition », *ICDAR*, 2011.

Katz S., « Estimation of probabilities from sparse data for the language model component of a speech recognizer », *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 35, n° 3, p. 400-401, 1987.

Marti U.-V., Bunke H., « Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System », *IJPRAI*, vol. 15, n° 1, p. 65-90, 2001.

Plamondon R., Srihari S., « Online and Offline Handwriting recognition : A Comprehensive survey », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n° 1, p. 63-84, January, 2000.

Rodríguez-Serrano J., Perronnin F., « Handwritten word-spotting using hidden Markov models and universal vocabularies », *Pattern Recognition*, vol. 42, n° 9, p. 2106-2116, 2009.

Stolcke A., « SRILM : An Extensible Language Modeling Toolkit », *Proc. International Conference on Spoken Language Processing*, p. 901-904, 2002.

Vinciarelli A., Bengio S., Bunke H., « Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models », *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, p. 709-720, June, 2004.

Vinciarelli A., Luettin J., « A new normalization technique for cursive handwritten words », *Pattern Recognition Letters*, vol. 22, n° 9, p. 1043-1050, 2001.