HAL
open science

# Détection et correction d'erreurs utilisant les probabilités a posteriori dans un système de reconnaissance de phrases manuscrites en-ligne

Solen Quiniou, Eric Anquetil

▶ **To cite this version:**

Solen Quiniou, Eric Anquetil. Détection et correction d'erreurs utilisant les probabilités a posteriori dans un système de reconnaissance de phrases manuscrites en-ligne. Revue I3 - Information Interaction Intelligence, 2009, 8 (2). hal-00582418

**HAL Id: hal-00582418**

**https://hal.science/hal-00582418**

Submitted on 1 Apr 2011

**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.

# Détection et correction d'erreurs utilisant les probabilités *a posteriori* dans un système de reconnaissance de phrases manuscrites en-ligne

Solen Quiniou, Eric Anquetil

IRISA - INSA, Campus Universitaire de Beaulieu
35042 Rennes Cedex, France
{Solen.Quiniou,Eric.Anquetil}@irisa.fr

***Résumé***

*Dans cet article, nous présentons un système complet de reconnaissance de phrases manuscrites en-ligne. Nous nous intéressons plus particulièrement à la détection d'erreurs potentielles sur les phrases issues d'une reconnaissance avec une approche au Maximum A Posteriori. Les probabilités a posteriori des mots, obtenues à partir d'une représentation sous la forme d'un réseau de confusion, sont ainsi utilisées comme indices de confiance. Des classifieurs dédiés (ici, des SVM) sont ensuite appris afin de corriger ces erreurs, en combinant ces probabilités a posteriori à d'autres sources de connaissance. Un mécanisme de rejet est également introduit afin de distinguer les hypothèses d'erreur qui ne pourront être corrigées par l'approche proposée. Des expérimentations ont été menées sur une base de 425 phrases manuscrites écrites par 17 scripteurs. Elles ont mis en évidence une réduction relative du taux d'erreur sur les mots de 14,6 %.*

***Mots-clés*** *: Reconnaissance de phrases manuscrites, réseaux de confusion, probabilités a posteriori, indices de confiance, rejet*

***Abstract***

*In this paper, we present a whole on-line sentence recognition system. We focus on the detection of potential errors on sentences obtained by a recognition using a Maximum A Posteriori approach. Word a posteriori probabilities are computed on confusion networks and are thus used as confidence indexes. Dedicated classifiers (here, SVMs) are then trained to correct these errors and combine the word a posteriori probabilities with other sources of knowledge. A rejection mechanism is also introduced to discriminate error hypotheses that won't be corrected by the proposed approach. Experiments were carried out on 425 handwritten sentences by 17 writers. They showed a 14.6 % relative reduction of the word error rate.*

***Key-words****: Handwritten sentence recognition, confusion networks, a posteriori probabilities, confidence indexes, rejection mechanism*

# 1 INTRODUCTION

Alors que les recherches en reconnaissance d'écriture manuscrite sont bien avancées en ce qui concerne la reconnaissance de caractères isolés et de mots [16, 20], les travaux concernant la reconnaissance de phrases ou de textes manuscrits sont encore relativement récents [2, 11, 14, 22]. Cette tâche est en effet plus complexe et apporte de nouveaux défis. L'intégration de connaissances linguistiques à l'aide de modèles de langage a tout d'abord permis aux systèmes de reconnaissance de phrases hors-ligne ou en-ligne de considérablement améliorer leurs performances : en moyenne, le taux d'erreur sur les mots a ainsi été diminué de 50 %. Malgré cet ajout de connaissances linguistiques, les systèmes de reconnaissance d'écriture manuscrite ne sont pas parfaits et il reste des erreurs dans la *phrase résultat*, c'est-à-dire la phrase en sortie du système de reconnaissance. Il peut alors être intéressant de mesurer la confiance que l'on a en chacun des mots résultats, pour pouvoir identifier ceux sur lesquels il subsiste un doute. Cela permet alors de mettre ces mots en évidence, pour un utilisateur (afin de faciliter sa tâche de relecture), ou encore d'essayer de corriger les erreurs potentielles détectées.

Des indices de confiance sont utilisés en reconnaissance d'écriture [15] mais la majorité d'entre eux concernent des caractères ou des mots isolés. Dans [2], une mesure de confiance reposant sur l'intégration d'un modèle de langage est utilisée pour rejeter les mots de phrases manuscrites hors-ligne, en comparant cet indice de confiance à un seuil fixé *a priori*. Les autres travaux associant des indices de confiance aux mots de phrases résultats d'une reconnaissance se trouvent en reconnaissance automatique de la parole. En effet, dans ce domaine, beaucoup de systèmes de reconnaissance associent une mesure de confiance aux mots de la phrase résultat. Parmi les plus courantes, celles utilisant les probabilités *a posteriori* des mots donnent les meilleurs résultats [9]. Ces probabilités peuvent également être combinées avec d'autres sources d'informations à l'aide de classifieurs [10, 8].

Dans cet article, nous proposons un système complet de reconnaissance de phrases manuscrites en-ligne, qui est une extension de celui présenté dans [18]. Nous ajoutons ici une étape permettant de détecter des erreurs potentielles sur la phrase résultat d'une reconnaissance utilisant l'approche au *Maximum A Posteriori* (MAP). Elles sont détectées en utilisant les probabilités *a posteriori* des mots résultats en tant qu'indices de confiance. Pour calculer ces probabilités *a posteriori*, nous ajoutons une représentation des hypothèses de phrases sous la forme d'un *réseau de confusion*. Cette représentation a été introduite par [12] pour la reconnaissance de la parole et c'est la première fois, à notre connaissance, qu'elle est adaptée au cadre de la reconnaissance d'écriture manuscrite. Une fois ces erreurs potentielles détectées, une étape de correction utilisant des classifieurs intervient. Un mécanisme de rejet est également ajouté avant l'étape de correction afin d'identifier les hypothèses d'erreur qui ne pourront pas être corrigées par l'approche proposée. Cela permet de mettre en évidence les mots non reconnus dans

une interface de saisie (afin de faciliter leur correction par l'utilisateur) voire de les soumettre à une nouvelle étape de reconnaissance utilisant de nouvelles informations. Nous avons présenté une première version de notre approche de détection, rejet et correction des erreurs potentielles dans [17]. Nous l'étendons ici en détaillant plus particulièrement les extensions apportées à la construction et à l'utilisation des réseaux de confusion, pour prendre en compte les spécificités de la reconnaissance d'écriture manuscrite. Ces modifications concernent notamment la gestion de plusieurs hypothèses de segmentation des phrases ainsi que la prise en compte de leur longueur. Nous détaillons également la stratégie complète de détection et correction des erreurs potentielles, incluant le mécanisme de rejet étendu à la segmentation automatique des phrases.

La suite de cet article est décomposée de la façon suivante. Le système de reconnaissance de phrases manuscrites est décrit dans la section 2. La construction et l'utilisation des réseaux de confusion est ensuite détaillée dans la section 3. La stratégie proposée pour détecter, rejeter et corriger les erreurs potentielles est présentée dans la section 4. Les résultats expérimentaux sont enfin discutés dans la section 5 et la section 6 tire les conclusions.

# 2 SYSTÈME DE RECONNAISSANCE DE PHRASES MANUSCRITES EN-LIGNE

Le système de reconnaissance de phrases (voir figure 1) se compose de quatre modules principaux, présentés dans les sous-sections suivantes. L'objet de cet article étant la détection, le rejet et la correction des erreurs potentielles, les deux derniers modules seront détaillés dans les sections 3 et 4.

## 2.1 Extraction automatique des mots

Le premier module (voir figure 1, partie (a)) permet l'extraction des mots de la phrase manuscrite présentée en entrée du système de reconnaissance. Comme une phrase manuscrite en-ligne est représentée par une liste de traces [1] ordonnées chronologiquement, cette tâche consiste à extraire les sous-listes de traces correspondant aux mots de la phrase. Notre approche [18] se base pour cela sur une caractérisation des espaces entre deux traces manuscrites consécutives. Pour ce faire, un classifieur de type RBFN (réseau de neurones à fonction à base radiale) est appris et permet d'attribuer, à chaque espace, une classe parmi les suivantes : espace intra-mot, espace inter-mot et espace inter-ligne. Quatre caractéristiques sont données en entrée du RBFN et correspondent à la distance entre les deux traces ainsi qu'à des informations sur la taille des espaces dans un historique local de traces.

1. Une *trace* est une liste de points ordonnés chronologiquement et capturés entre un posé et un levé de stylo.

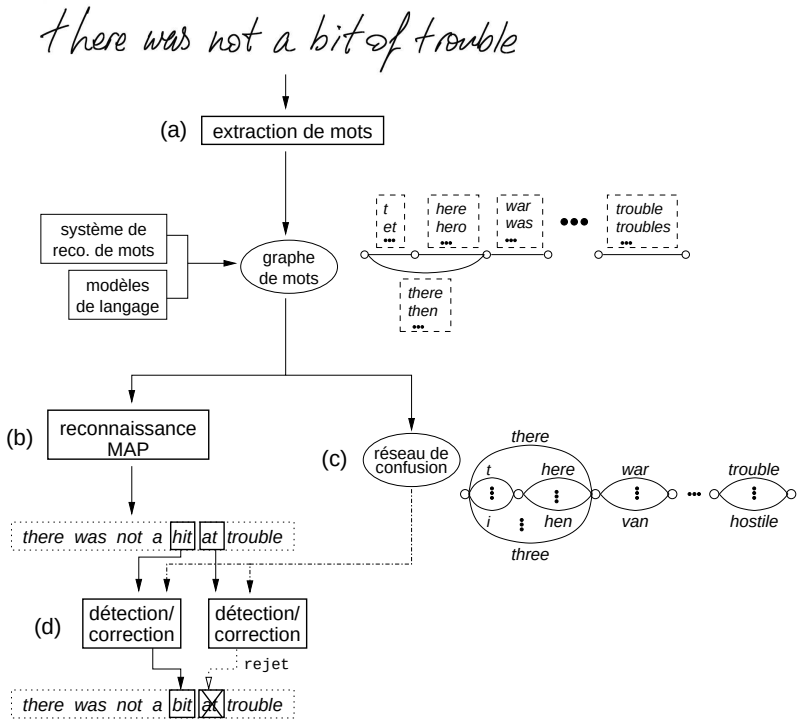



FIGURE 1 – Système de reconnaissance de phrases manuscrites.

Pour absorber les variations de style d'écriture lors du calcul de distance, la distance entre traces est calculée dans la zone comprise entre les lignes de base ; ces lignes sont estimées par rapport à un historique local de traces (voir figure 2). De plus, les caractéristiques utilisées s'appuient uniquement sur cet historique local de traces (correspondant à une fenêtre glissante de traces précédemment écrites) et sont normalisées par rapport à une estimation locale de la hauteur du corps du mot dans cet historique.

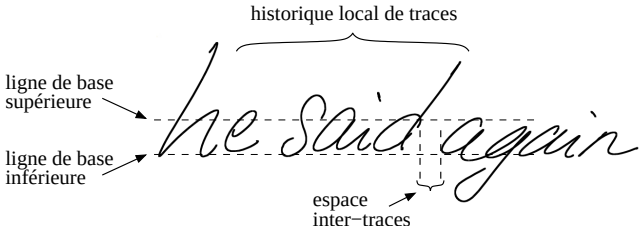



FIGURE 2 – Caractérisation des espaces inter-traces de la phrase.

Afin de gérer d'éventuels problèmes de sur- ou sous-segmentation, un indice de confiance est associé au résultat du RBFN pour évaluer la fiabilité de la première réponse. Il est ensuite comparé à un seuil appris automatiquement et permet d'ajouter des hypothèses supplémentaires de segmentation tout en contrôlant leur nombre total, pour limiter la combinatoire.

À l'issue de cette étape de segmentation, un graphe de mots contenant les hypothèses de mots extraites est construit. Chaque nœud représente une frontière entre deux mots consécutifs et les arcs représentent les mots extraits.

## 2.2 Reconnaissance des phrases par décodage MAP

La reconnaissance de phrases par décodage MAP (*Maximum A Posteriori*) consiste à trouver la phrase $\hat{W}$ qui maximise la probabilité *a posteriori* parmi toutes les phrases $W_k = w_{k,1} \dots w_{k,n_k}$, étant donné un signal $S$ (correspondant à la phrase manuscrite à reconnaître). La phrase $\hat{W}$ est définie par :

$$\hat{W} = \arg\max_{W_k} \; P(S|W_k) + \gamma \, \log\left[P(W_k)\right] + \delta \, n_k, \tag{1}$$

avec $P(S|W_k)$ le score du signal $S$ étant donnée la phrase $W_k$, estimé par le système de reconnaissance et $P(W_k)$ la probabilité *a priori* de la phrase $W_k$, donnée par un modèle de langage. Pour intégrer efficacement les informations apportées par le modèle de langage et pour prendre en compte les éventuels problèmes de segmentation, deux pondérations sont utilisées. Le *poids linguistique* $\gamma$ ($GSF$, pour *Grammar Scale Factor*) sert à compenser l'impact du modèle de langage vis-à-vis du système de reconnaissance et le *poids de segmentation* $\delta$ ($WIP$, pour *Word Insertion Penalty*) permet de gérer les problèmes de segmentation, en pondérant le nombre de mots $n_k$ de la phrase. Lorsque $\delta < 0$, les phrases courtes sont favorisées (compensation des sur-segmentations) et lorsque $\delta > 0$, ce sont les phrases longues qui sont préférées (compensation des sous-segmentations). L'optimisation des poids $\gamma$ et $\delta$ permet de réduire considérablement le taux d'erreur sur les mots.

En réalité, notre système de reconnaissance n'est pas probabiliste mais le score donné par celui-ci peut être combiné aux log-probabilités du modèle de langage car ce score est dans le domaine logarithmique et dans les mêmes plages de valeurs que ces log-probabilités. L'équation 1 devient alors :

$$\hat{W} = \arg\max_{W_k} \; sc(S|W_k) + \gamma \, \log\left[P(W_k)\right] + \delta \, n_k, \tag{2}$$

avec $sc(S|W_k)$ le score estimé par notre système de reconnaissance de mots RESIFMot [3] sur chacun des mots $w_{k,i}$ de la phrase $W_k$ (ce score combine des mesures d'adéquation entre les caractères et leur modèles de lettres ainsi que des informations statistiques et spatiales entre ces caractères et dépend également des opérations d'édition effectuées durant le post-traitement lexical). Le modèle de langage utilisé pour estimer $P(W_k)$ est un modèle bigramme, c'est-à-dire qu'il donne la probabilité de chaque mot $w_{k,i}$ en fonction du mot $w_{k,j}$ qui le précède (cette probabilité est notée $P(w_{k,i}|w_{k,j})$).

Le décodage MAP est réalisé sur le graphe de mots à l'aide de l'algorithme de Viterbi [5] (voir figure 1, partie (b)) et permet d'identifier efficacement la meilleure phrase (correspondant au meilleur chemin dans le graphe), en combinant les informations graphiques et linguistiques comme donné par l'équation 2. Pour chaque mot manuscrit extrait (représenté par un arc dans le graphe), le système de reconnaissance de mots RESIFMot donne une liste de 20 mots candidats, avec un score associé à chacun d'eux.

### 2.3 Construction du réseau de confusion

Pour calculer les probabilités *a posteriori* des mots, utilisées ensuite comme indices de confiance pour détecter les erreurs potentielles de reconnaissance, un réseau de confusion est construit à partir du graphe de mots (voir figure 1, partie (c)). Cette étape est détaillée dans la section 3.

### 2.4 Détection, correction et rejet des erreurs

Le dernier module du système de reconnaissance s'occupe enfin de détecter les erreurs potentielles de reconnaissance sur la meilleure hypothèse de phrase obtenue par le décodage MAP (voir figure 1, partie (d)). Ces hypothèses d'erreur sont ensuite soit corrigées, soit rejetées (si elles ne peuvent pas être corrigées), en utilisant pour cela des classifieurs. Cette dernière étape du système de reconnaissance de phrases est présentée dans la section 4.

## 3 RÉSEAU DE CONFUSION ET CONSENSUS

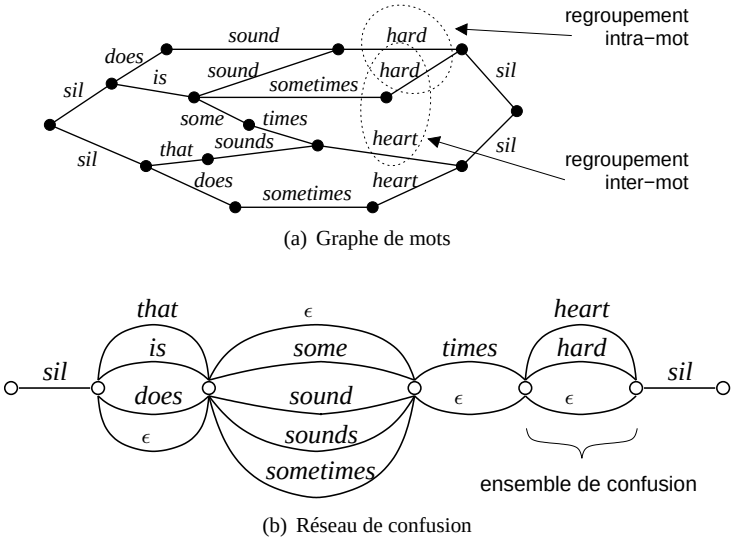
Dans cette section, nous décrivons tout d'abord le principe général des réseaux de confusion puis nous présentons les différentes étapes de la construction d'un réseau de confusion ainsi que le choix de la meilleure phrase, lorsque ces réseaux sont utilisés en reconnaissance de la parole. Dans la dernière sous-section, nous détaillons enfin les modifications que nous avons apportées pour tenir compte des spécificités de la reconnaissance d'écriture.

### 3.1 Principe général

Dans [12], une nouvelle approche pour la reconnaissance de phrases a été proposée en reconnaissance automatique de la parole. Contrairement à l'approche MAP qui vise à maximiser la probabilité de la phrase résultat, cette nouvelle approche est repose sur la minimisation du taux d'erreur sur les mots. Elle est alors plus proche du but de la reconnaissance de phrases qui n'est pas d'obtenir la phrase ayant la probabilité la plus élevée mais plutôt celle qui contient le plus de mots correctement reconnus.

Dans cette approche, l'ensemble des hypothèses de phrases est représenté par un *réseau de confusion* (voir figure 3(b)). Les nœuds d'un tel réseau

représentent des classes d'équivalence, appelées *ensembles de confusion*, qui correspondent à des confusions entre des hypothèses de mots situées à une même position donnée de la phrase à reconnaître. Deux nœuds adjacents sont alors reliés par autant d'arcs que d'hypothèses de mots. Par exemple, dans la figure 3(b), les mots *that*, *is*, *does* et $\epsilon$ (correspondant à une suppression) appartiennent au même ensemble de confusion.




(a) Graphe de mots

(b) Réseau de confusion

FIGURE 3 – Graphe de mots et réseau de confusion correspondant.

À chacun des mots du réseau de confusion est associée une *probabilité a posteriori*, qui correspond à la somme des probabilités de tous les chemins du graphe de mots initial auxquels ce mot appartient. Ces probabilités *a posteriori* peuvent ensuite être utilisées soit directement pour effectuer la reconnaissance en retrouvant la meilleure hypothèse de phrase, soit en tant qu'indices de confiance sur les mots.

## 3.2 Construction du réseau de confusion

L'algorithme 1 donne les différentes étapes pour construire un réseau de confusion (voir figure 3(b)) à partir d'un graphe de mots (voir figure 3(a)) et pour retrouver la phrase résultat qui minimise le taux d'erreur sur les mots. Cet algorithme est celui utilisé en reconnaissance de la parole [12] et ses étapes sont détaillées dans les paragraphes suivants. En ce qui concerne les modifications apportées pour la reconnaissance d'écriture, nous les présenterons dans la sous-section 3.3.

**Algorithme 1** : Algorithme de construction du réseau de confusion.

**Données** :
$G$ : le graphe de mots ;

**Résultat** :
$C_G$ : le réseau de confusion construit à partir du graphe $G$ ;

**début**
§ 3.2.1 calcul des probabilités *a posteriori* des arcs du graphe $G$ ;
§ 3.2.2 création des ensembles de confusion ;
§ 3.2.3 calcul des probabilités *a posteriori* ;
§ 3.2.4 choix de l'hypothèse consensus $W_{consensus}$ ;
**fin**

### 3.2.1 Calcul des probabilités *a posteriori* des arcs

Les probabilités *a posteriori* de chacun des arcs du graphe de mots sont tout d'abord calculées. La probabilité du $k^e$ arc associé à l'étiquette de mot $w_i$ correspond à la somme des probabilités des chemins qui passent par cet arc. Elle peut être calculée efficacement en utilisant l'algorithme *forward-backward* [1] et est donnée par l'équation suivante :

$$P_{post}(w_i^{(k)}) = \sum_j \frac{\alpha(n_{t-1})\ P(w_i|w_j)^\gamma\ P(s_i|w_i)^\zeta\ \beta(n_t)}{\alpha(n_T)} \tag{3}$$

avec $w_i^{(k)}$ le $k^e$ arc correspondant au mot $w_i$, $\alpha(n_{t-1})$ la probabilité *forward* du nœud $n_{t-1}$, $\beta(n_t)$ la probabilité *backward* du nœud $n_t$, $P(w_i|w_j)$ la probabilité donnée par le modèle de langage au bigramme $w_j w_i$ (pondérée par $\gamma$) et $P(s_i|w_i)$ la probabilité acoustique donnée par le système de reconnaissance de parole au mot $w_i$, à partir du signal acoustique $s_i$ (pondérée par $\zeta$). La probabilité *forward* d'un nœud correspond à la somme des probabilités de tous les chemins entre le nœud initial et ce nœud, alors que la probabilité *backward* correspond à la somme des probabilités de tous les chemins entre ce nœud et le nœud final. La probabilité $P_{post}(w_i^{(k)})$ est normalisée par la somme des probabilités de tous les chemins du graphe de mots, représentée par $\alpha(n_T)$ (où $T$ est le nombre de nœuds du graphe).

### 3.2.2 Création des ensembles de confusion

La création des ensembles de confusion est réalisée en alignant les hypothèses du graphe de mots initial.

Les ensembles de confusion sont tout d'abord initialisés de telle sorte que chacun ne contienne que les arcs ayant des temps de début et de fin ainsi que des étiquettes de mots identiques (une étape d'élagage peut être utilisée au préalable, pour supprimer les arcs de probabilité *a posteriori* trop faible).

L'étape de regroupement intra-mot intervient ensuite pour fusionner les ensembles de confusion qui ont la même étiquette de mot mais des temps de début et/ou de fin qui se chevauchent. La mesure de similarité utilisée pour ce regroupement s'appuie alors sur le chevauchement temporel des arcs.

L'étape de regroupement inter-mot permet enfin de fusionner les ensembles de confusion correspondant à des mots différents, c'est-à-dire qui sont des hypothèses de reconnaissance de mot différentes, pour la même position donnée de la phrase à reconnaître. Pour cela, le calcul de similarité entre arcs se base également sur un chevauchement temporel ainsi que sur une similarité phonétique entre les mots qu'ils représentent.

### 3.2.3 Calcul des probabilités *a posteriori* des mots

Une fois les ensembles de confusion créés, la probabilité *a posteriori* de chacun des mots $w_i$ peut être calculée, à partir de celles des arcs dont l'étiquette correspond au mot $w_i$ :

$$P_{post}(w_i) = \sum_{k=1}^{K} P_{post}(w_i^{(k)}) \tag{4}$$

avec $P_{post}(w_i^{(k)})$ la probabilité *a posteriori* du $k^e$ arc correspondant au mot $w_i$ (calculée par l'équation 3) et $K$ le nombre total de ces arcs.

### 3.2.4 Choix de l'hypothèse consensus

Le réseau de confusion est maintenant entièrement construit. L'hypothèse de phrase $W_{consensus}$ qui maximise le nombre de mots correctement reconnus peut alors être choisie : elle est appelée *hypothèse consensus*. Il a été montré dans [12] que l'hypothèse de phrase qui minimise le taux d'erreur sur les mots peut être obtenue en choisissant, dans chaque ensemble de confusion, le mot dont la probabilité *a posteriori* est la plus élevée :

$$W_{consensus} = w^1_{consensus}\, w^2_{consensus} \cdots w^L_{consensus} \tag{5}$$

avec $L$ le nombre d'ensembles de confusion et $w^l_{consensus}$ le mot défini par :

$$w^l_{consensus} = \arg\max_{w_i \in C_l} P_{post}(w_i) \tag{6}$$

avec $C_l$ le $l^e$ ensemble de confusion.

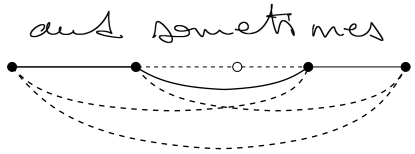
## 3.3 Extension pour la reconnaissance de phrases manuscrites en-ligne

### 3.3.1 Limitations de l'approche précédente

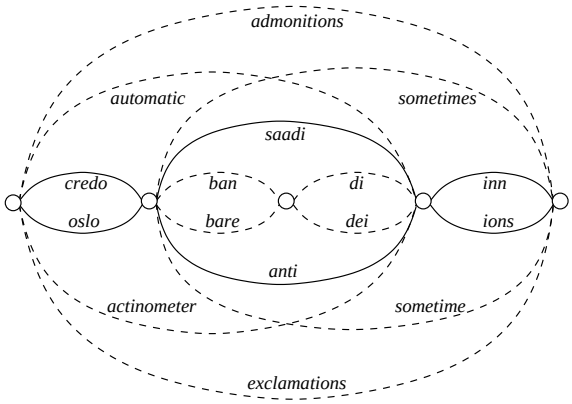
Étant donnée la structure de notre graphe de mots, l'étape de création des ensembles de confusion est immédiate. En effet, les listes de mots candi-

dats associées à chacun des arcs (et données par notre système de reconnaissance de mots) correspondent aux ensembles de confusion. En réalité, c'est le cas lorsqu'il n'y a qu'une seule hypothèse de segmentation de la phrase : il n'existe alors qu'un seul arc entre deux nœuds consécutifs du graphe.

Lorsque plusieurs hypothèses de segmentation sont présentes dans le graphe de mots, nous obtenons un graphe comme celui illustré par la figure 4(a). Nous souhaitons alors obtenir le sous-réseau de confusion correspondant à la figure 4(b). Dans ce sous-réseau, chacun des mots associés aux arcs correspond à un des mots candidats donnés par le système de reconnaissance de mots, à partir d'un mot manuscrit préalablement extrait.



(a) Partie d'un graphe de mots




(b) Sous-réseau de confusion correspondant

FIGURE 4 – Exemple de partie d'un graphe de mots, obtenu dans le cas de la reconnaissance d'écriture, et le sous-réseau de confusion associé.

Pour pouvoir considérer des sous-réseaux de confusion comme celui de la figure 4(b), au lieu des précédents ensembles de confusion, il est nécessaire de pouvoir aligner un arc avec plusieurs arcs. Une première approche a été présentée dans [21] mais elle se limitait à l'alignement d'un arc avec deux autres arcs. Nous étendons cette approche afin de pouvoir aligner un arc avec plus de deux arcs. Cela entraîne une modification du calcul des probabilités *a posteriori* des mots puisque ce calcul se base sur le regroupement des mots

dans les ensembles de confusion. De plus, un mot pouvant maintenant être aligné avec plusieurs autres mots, les phrases représentées par le réseau de confusion peuvent ne pas toutes avoir la même longueur. Nous prenons également en compte la longueur des chemins, lors du nouveau calcul des probabilités *a posteriori* des mots. La modification des ensembles de confusion engendre aussi des modifications lors du choix de l'hypothèse consensus.

Dans les deux paragraphes suivants, nous détaillons les modification effectuées au niveau du calcul des probabilités *a posteriori* des mots ainsi qu'au niveau du choix de l'hypothèse consensus.

### 3.3.2 Nouveau calcul des probabilités *a posteriori* des mots

Comme le réseau de confusion correspond au graphe de mots, les sous-réseaux de confusion s'obtiennent immédiatement et les probabilités *a posteriori* sont directement associées aux mots du graphe. Comme précédemment, le calcul des probabilités *a posteriori* s'appuie sur l'algorithme *forward-backward*. L'équation 7 donne alors la somme des probabilités des chemins contenant le mot $w_i$ qui appartient à la liste de mots candidats associée à un arc $a_{t-y}^{t}$ du graphe de mots ; ce nouveau calcul permet de prendre en compte la nouvelle structure des sous-réseaux de confusion ainsi que la longueur des chemins auxquels appartient ce mot :

$$\sigma_{post}(w_i\langle_{t-y}^{t}\rangle) = \sum_{x,j,k,l} \alpha_k(w_j\langle_{t-y-x}^{t-y}\rangle)P(w_i|w_j)^{\gamma}\widetilde{sc}(s_i|w_i)^{\zeta}\beta_l(w_i\langle_{t-y}^{t}\rangle)(k+l)^{\delta}, \tag{7}$$

avec $\alpha_k(w_j\langle_{t-y-x}^{t-y}\rangle)$ la probabilité *forward* du mot $w_j$ appartenant à l'arc $a_{t-y-x}^{t-y}$ (prédécesseur de l'arc $a_{t-y}^{t}$ considéré) pour les chemins de longueur $k$, $P(w_i|w_j)^{\gamma}$ la probabilité donnée par le modèle de langage au bigramme $w_jw_i$ et pondérée par $\gamma$, $\widetilde{sc}(s_i|w_i)^{\zeta}$ le score normalisé du mot $w_i$ et pondéré par $\zeta$ (la normalisation est effectuée par rapport aux scores minimaux et maximaux des mots de longueur comparable), $\beta_l(w_i\langle_{t-y}^{t}\rangle)$ la probabilité *backward* du mot $w_i$ (qui appartient à l'arc $a_{t-y}^{t}$) pour les chemins de longueur $l$ et $(k+l)^{\delta}$ la longueur du chemin considéré et pondérée par $\delta$.

La *probabilité a posteriori* du mot $w_i$ est finalement obtenue en normalisant la somme des probabilités des chemins contenant le mot $w_i$ (donnée par l'équation 7), par rapport à la somme des probabilités de tous les chemins du graphe de mots (représentée par la variable $denom$) :

$$P_{post}(w_i) = \frac{\sigma_{post}(w_i)}{denom}. \tag{8}$$

### 3.3.3 Nouveau calcul de l'hypothèse consensus

Les ensembles de confusion correspondant maintenant à des sous-réseaux de confusion, il n'est plus toujours possible de choisir, dans chacun d'eux, le

mot ayant la probabilité *a posteriori* la plus élevée. En effet, comme montré par la figure 4(b), ils représentent, non plus des confusions sur des mots, mais des confusions sur une séquence de mots, sous-partie de la phrase à reconnaître. Il faut alors trouver le meilleur chemin dans chacun des sous-réseaux, en s'appuyant sur les probabilités *a posteriori* des mots qui le composent. Pour cela, nous utilisons l'algorithme de Viterbi qui génère la liste des $M$ meilleurs chemins ; nous prenons alors, comme score d'un chemin, le produit des probabilités *a posteriori* des mots qui le compose.

L'hypothèse consensus est alors obtenue en choisissant, dans chaque sous-réseau, le chemin dont le score est le plus élevé. L'équation 5 devient alors :

$$W_{consensus} = W^1_{consensus} W^2_{consensus} \dots W^L_{consensus} \tag{9}$$

avec $L$ le nombre de sous-réseaux de confusion et $W^l_{consensus}$ la séquence de mots définie par :

$$W^l_{consensus} = \arg \max_{W^l \in C_l} \prod_{j=1}^{N} P_{post}(w_j^{(l)}) \tag{10}$$

avec $C_l$ le $l^e$ sous-réseau de confusion et $W^l = w_1^{(l)} \dots w_N^{(l)}$ un chemin (composé de $N$ mots) de ce sous-réseau $C_l$.

# 4 DÉTECTION ET CORRECTION D'ERREURS UTILISANT LES PROBABILITÉS *a posteriori*

Dans cette section, nous utilisons les probabilités *a posteriori* des mots, calculées sur le réseau de confusion, pour détecter des erreurs potentielles de reconnaissance sur la phrase résultat obtenue par l'approche MAP. Nous présentons tout d'abord le principe général de notre approche pour détecter et corriger ou rejeter ces erreurs potentielles puis, dans les sous-sections suivantes, nous détaillons chacune des étapes de détection, correction et rejet.

## 4.1 Principe général

La détection des erreurs potentielles repose, d'une part, sur le calcul de la meilleure phrase en utilisant l'approche MAP (voir sous-section 2.2) et, d'autre part, sur l'utilisation des probabilités *a posteriori* des mots de cette phrase (le calcul de ces probabilités est donné par les équations 7 et 8).

L'algorithme 2 détaille notre approche de détection et de correction ou rejet des hypothèses d'erreur identifiées. La probabilité *a posteriori* de chacun des mots de la phrase résultat est utilisée, en tant qu'indice de confiance, pour détecter si ce mot est potentiellement une erreur de reconnaissance (appelé également *hypothèse d'erreur*). Nous considérons deux cas d'erreurs potentielles pour le mot $w_i$ :

– un autre mot de la liste de mots candidats $l_i$ (à laquelle il appartient) a une probabilité *a posteriori* plus élevée que celle de $w_i$ (§ 4.2) ;
– la probabilité *a posteriori* de $w_i$ est considérée comme trop faible (§ 4.3).

**Algorithme 2** : Algorithme de détection et de correction des erreurs potentielles de reconnaissance, incluant un mécanisme de rejet.

**Données** :
$\widehat{W}_{MAP}$ : la phrase résultat en utilisant l'approche MAP ;
$P_{post}(w_i)$ : la probabilité *a posteriori* du mot $w_i$ ;
$\theta_{post}$ : le seuil sur les probabilités *a posteriori* des mots ;
$w_{max}$ : le mot avec la probabilité maximale dans la liste $l_i$ ;
$w_{max_2}$ : le mot avec la deuxième probabilité la plus élevée, dans $l_i$ ;
$correction_{1,2}(w_i, w_k)$ : les fonctions de correction pour choisir entre les mots $w_i$ et $w_k$ ;
$rejet_{1,2}(w_i, w_k)$ : les fonctions pour rejeter éventuellement le mot $w_i$, à partir des informations sur les mots $w_i$ et $w_k$ ;

**Résultat** :
$\widehat{W}$ : la phrase résultat corrigée et pouvant comporter des mots rejetés ;

**début**
  **pour chaque** *mot* $w_i$ *de la phrase* $\widehat{W}_{MAP}$ **faire**
§ 4.2  **si** $P_{post}(w_i) \neq P_{post}(w_{max})$ **alors**
§ 4.4  **si** $rejet_1(w_i, w_{max})$ **alors**
    $\widehat{w_i} = REJET$ ;
    **sinon**
    $\widehat{w_i} = correction_1(w_i, w_{max})$ ;
§ 4.3  **sinon si** $P_{post}(w_i) < \theta_{post}$ **alors**
§ 4.4  **si** $rejet_2(w_i, w_{max2})$ **alors**
    $\widehat{w_i} = REJET$ ;
    **sinon**
    $\widehat{w_i} = correction_2(w_i, w_{max2})$ ;
  $\widehat{W} = \widehat{w_1}\,\widehat{w_2} \ldots \widehat{w_N}$ ;
**fin**

Nous associons alors une *fonction de correction*, à chaque type d'hypothèse d'erreur : elle vise à choisir entre le mot résultat et un des autres mots de la liste $l_i$ à laquelle il appartient. Certaines erreurs ne peuvent pas être corrigées par l'étape de correction mais il est intéressant de pouvoir tout de même les identifier. Nous ajoutons alors un mécanisme de rejet qui permet de détecter ces erreurs restantes (§ 4.4). La phrase corrigée est finalement obtenue à partir de chacun des mots résultats qui peuvent avoir été corrigés ou rejetés.

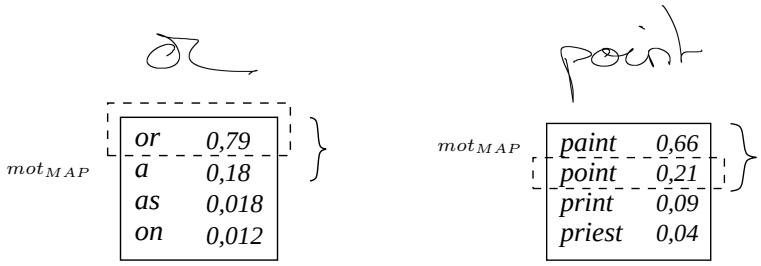
## 4.2 Mots de probabilité *a posteriori* non-maximale

### 4.2.1 Détection

Dans ce premier cas d'erreur potentielle, nous considérons qu'il peut y avoir une erreur de reconnaissance si la probabilité *a posteriori* du mot résultat $w_i$ n'est pas la plus élevée, parmi celles des mots appartenant à la liste de mots candidats $l_i$. Cela signifie qu'il existe au moins un autre mot de la liste $l_i$ dont l'indice de confiance est plus élevé que celui de $w_i$.

Pour détecter ce type d'erreur, nous comparons la probabilité *a posteriori* du mot résultat $w_i$ avec celle du mot $w_{max}$, qui a la probabilité la plus élevée.

La figure 5(a) donne un exemple de ce type d'hypothèse d'erreur. Dans cet exemple, le mot *a* est le mot résultat (appelé aussi $mot_{MAP}$, sur la figure) alors que le mot correct est *or* (encadré en pointillé, sur la figure), ce dernier ayant une probabilité *a posteriori* plus élevée.




(a) Mot *a* de probabilité non-maximale (b) Mot *paint* de probabilité faible

FIGURE 5 – Exemples d'hypothèses d'erreur pouvant être corrigées.

### 4.2.2 Correction

Pour proposer une correction de l'hypothèse d'erreur, nous considérons que le mot correct est soit le mot résultat $w_i$, soit le mot $w_{max}$ qui a la probabilité *a posteriori* maximale (voir figure 5(a)). Les expérimentations montrent que, dans environ 85 % des cas, lorsque le mot correct est présent dans la liste, il correspond à l'un de ces deux mots.

Un classifieur, dédié à la correction de ce type d'erreur, est alors appris pour faire un choix entre ces deux mots. Il comporte 6 entrées (3 caractéristiques pour chacun des deux mots) et 2 sorties (chacune représente un des mots). Les 3 caractéristiques utilisées pour chaque mot sont les suivantes :

- sa probabilité *a posteriori* $P_{post}(w_i)$ ;
- son score lexical normalisé $\widetilde{sc}_{lex}(w_i)$ ;
- son score graphique normalisé $\widetilde{sc}_{graph}(w_i)$.

Le *score lexical* (dépendant des opérations d'édition durant l'étape de post-traitement lexical [3]) et le *score graphique* (combinant des mesures d'adéquation entre chacun des caractères et son modèle de lettre ainsi que des

informations spatiales et statistiques entre les caractères du mot) sont donnés par le système de reconnaissance de mots. Ils sont normalisés de la même façon que le score présenté dans le paragraphe 3.3.2. Ces caractéristiques permettent de considérer à la fois des informations *globales* sur le graphe de mots (les probabilités *a posteriori* des mots, qui correspondent à la somme des probabilités des chemins auxquels appartiennent les mots) et des informations *locales* sur les mots (leurs scores lexicaux et graphiques).

## 4.3 Mots de probabilité *a posteriori* faible

### 4.3.1 Détection

Dans ce deuxième cas, le mot résultat $w_i$ est considéré comme hypothèse d'erreur si sa probabilité *a posteriori* est jugée trop faible (sa probabilité est la probabilité maximale de sa liste de mots $l_i$). Nous considérons ainsi que la confiance dans le résultat de la reconnaissance n'est pas assez fiable.

Pour détecter ce type d'erreur potentielle, nous comparons la probabilité *a posteriori* du mot $w_i$ par rapport à un seuil $\theta_{post}$. Si la probabilité est inférieure à ce seuil, le mot est alors considéré comme une hypothèse d'erreur.

La figure 5(b) donne un exemple de ce type d'hypothèse d'erreur. Dans cet exemple, le mot *paint* est le mot résultat mais sa probabilité *a posteriori* est en-dessous du seuil $\theta_{post}$, fixé empiriquement à 0,8.

### 4.3.2 Correction

La proposition de correction de l'hypothèse d'erreur est réalisée de la même façon que dans le cas précédent, en utilisant un nouveau classifieur, dédié à la correction de ce deuxième type d'erreur et permettant de choisir entre deux mots.

Le choix se porte alors soit sur le mot résultat $w_i$, soit sur le mot $w_{max2}$ ayant la deuxième probabilité *a posteriori* la plus élevée (voir figure 5(b)). De même que précédemment, les expérimentations ont montré que, lorsque le mot correct se trouve dans la liste, il correspond à un de ces deux mots dans à peu près 83 % des cas.

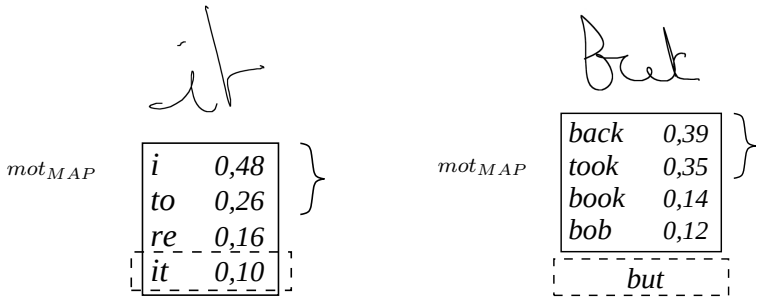
Les caractéristiques utilisées pour chacun de ces mots sont les mêmes que précédemment ; le classifieur appris possède également 6 entrées et 2 sorties.

## 4.4 Rejet des erreurs restantes

### 4.4.1 Limitations de la correction présentée

Dans les deux stratégies de correction présentées, nous supposons que le mot correct se trouve parmi les deux mots donnés en entrée de chacun des classifieurs dédiés. En fait, le mot correct peut soit être parmi les autres mots de la liste $l_i$, soit ne pas être dans cette liste. Dans l'exemple donné par la figure 6(a), *it* est le mot correct mais les deux mots utilisés pour la correction

sont *i* et *to*. De même, dans l'exemple donné par la figure 6(b), le mot correct *but* n'apparaît même pas parmi les mots de la liste (*took* et *back* étant utilisés pour la correction). Ainsi, au vu des expérimentations, lorsque le mot correct n'est pas un des deux mots utilisés pour la correction, il correspond à un des autres mots de la liste dans 40 % des cas et est absent de celle-ci dans les 60 % des cas restants.




(a) Mot *it* présent dans la liste (b) Mot *but* absent de la liste

FIGURE 6 – Exemples de mots à rejeter.

Cette simplification a été faite car la majorité des mots corrects se trouve parmi les deux mots utilisés pour la correction. Elle est aussi effectuée pour pallier un problème de manque de données puisqu'en considérant plus de mots en entrée des classifieurs dédiés, il est nécessaire d'avoir plus de données d'apprentissage afin d'obtenir des classifieurs fiables.

Un *mécanisme de rejet* est alors utilisé pour mesurer la confiance, non plus sur les mots résultats, mais sur le classifieur utilisé pour la correction. En effet, s'il est jugé que ce classifieur ne pourra pas corriger le mot résultat, le mot est alors rejeté.

### 4.4.2 Principe du mécanisme de rejet

Avec l'ajout du mécanisme de rejet, nous utilisons la fonction de rejet pour estimer si le mot pourra être corrigé. Pour définir cette notion de rejet, nous construisons deux *classifieurs de rejet*, un pour chaque type d'hypothèses d'erreur considérés. Cela correspond à l'architecture SC (classifieur spécialisé), présentée dans [13]. Un classifieur de rejet à deux classes (rejet et acceptation) est alors estimé à partir d'un classifieur principal qui réalise la classification des exemples acceptés (ici, ces classifieurs principaux correspondent aux classifieurs chargés de la correction des hypothèses d'erreur).

Chacun des deux classifieurs de rejet comporte 6 entrées, correspondant aux 3 caractéristiques de chacun des deux mots considérés pour la correction ($w_i$ et $w_{max}$ pour le premier cas d'erreur, et $w_i$ et $w_{max2}$ pour le second cas) ; ce sont donc les mêmes entrées que pour les classifieurs de correction.

# 5 EXPÉRIMENTATIONS ET RÉSULTATS

Dans cette section, après avoir présenté le cadre de nos expérimentations (incluant le paramétrage de nos méthodes ainsi que les classifieurs utilisés dans nos approches), nous évaluons la stratégie de détection et de correction ou rejet des erreurs potentielles de reconnaissance. Nous présentons également les résultats obtenus lorsque les réseaux de confusion sont directement utilisés pour effectuer la reconnaissance des phrases.

## 5.1 Cadre expérimental

### 5.1.1 Données manuscrites

Les données manuscrites sont des phrases saisies à partir de 2 598 phrases du corpus Brown. L'ensemble d'apprentissage inclut 517 phrases (soit 8 047 mots) écrites par 25 scripteurs. Cet ensemble d'apprentissage est utilisé pour régler les paramètres des algorithmes de Viterbi et du Consensus ainsi que pour apprendre les classifieurs utilisés pour la correction et le rejet des hypothèses d'erreur. L'ensemble de test, quant à lui, inclut 425 phrases (soit 6 362 mots) écrites par 17 scripteurs. Les scripteurs de l'ensemble de test sont différents de ceux de l'ensemble d'apprentissage. De plus, un ensemble d'apprentissage différent, composé de mots manuscrits isolés, a été utilisé pour apprendre le système de reconnaissance de mots RESIFMot [3].

### 5.1.2 Données linguistiques

Le modèle bigramme utilisé dans nos expérimentations est construit sur le corpus Brown [6], avec l'outil SRILM [19]. Ce corpus est composé de 52 954 phrases anglaises (soit 1 002 675 mots), parmi lesquelles 46 836 phrases (soit 900 108 mots) ont été utilisées pour l'apprentissage du modèle de langage (les 2 598 phrases restantes sont celles utilisées pour la saisie des phrases manuscrites). Le lexique associé est constitué de 13 748 mots et est utilisé lors de l'étape de post-traitement lexical du système de reconnaissance de mots RESIFMot.

### 5.1.3 Classifieurs utilisés

Pour la correction comme pour le rejet des hypothèses d'erreur, les classifieurs utilisés sont des *systèmes à vaste marge* (SVM) [4] avec des noyaux gaussiens. Nous avons choisi ces classifieurs pour leur efficacité et leur fiabilité ainsi que pour leur capacité d'apprentissage lorsque les classes sont déséquilibrées (en termes de nombre d'exemples d'apprentissage).

## 5.2 Utilisation des réseaux de confusion pour la reconnaissance de phrases

Bien que les réseaux de confusion servent ici à calculer les probabilités *a posteriori* des mots, utilisées ensuite en tant qu'indices de confiance, ces réseaux peuvent également être utilisés pour la reconnaissance de phrases. Nous comparons ainsi tout d'abord les résultats obtenus en utilisant l'approche par consensus (sur les réseaux de confusion) à ceux obtenus avec l'approche standard MAP (sur les graphes de mots), pour la reconnaissance de phrases. Ces résultats sont présentés dans la table 1, qui donne les taux de reconnaissance obtenus sur la base de test (les différents paramètres des algorithmes de reconnaissance ont été préalablement optimisés sur la base d'apprentissage, indépendamment pour chacune des approches). Le taux de reconnaissance correspondant au nombre de mots correctement reconnus, divisés par le nombre total de mots des phrases. Nous rappelons également le taux de reconnaissance obtenu sans utiliser de modèle de langage lors de la reconnaissance avec l'approche MAP (correspondant au système de base).

TABLE 1 – Comparaison des deux approches de reconnaissance.

| Approche de reconnaissance | Taux de reconnaissance | Baisse relative du taux d'erreur |
|---|---|---|
| Base | 71,4 % | - |
| MAP | 86,5 % | 51,7 % |
| Consensus ($\delta = 0$) | 83,9 % | 42,7 % |
| Consensus ($\delta_{opt}$) | 85,4 % | 47,9 % |

Nous considérons ici deux variantes de l'approche consensus. Dans le premier cas, le nombre de mots dans les phrases n'est pas considéré, ce qui correspond à un poids $\delta = 0$. Dans le second cas, la prise en compte du nombre de mots dans les phrases (pour le calcul des probabilités *a posteriori* des mots) est optimisée, ce qui correspond à $\delta_{opt}$ (la figure 7 montre l'évolution du taux de reconnaissance, sur la base d'apprentissage, en fonction des paramètres $\delta$ et $\zeta$).

Les taux de reconnaissance obtenus avec les deux approches utilisant le consensus sont inférieurs à celui obtenu en utilisant l'approche MAP. Cette différence pourrait être expliquée par la normalisation du score $\widetilde{sc}(s_i|w_i)$ qui n'est peut-être pas optimale. De plus, ce score n'étant pas de nature probabiliste, sa combinaison avec les probabilités données par le modèle de langage n'est peut-être pas non plus optimale. Nous remarquons néanmoins que la prise en compte du nombre de mots dans les phrases permet d'améliorer le taux de reconnaissance sur les mots de 1,5 %, en absolu.

Un test de significativité comme le *paired t-test* [7] permet de comparer deux algorithmes en mesurant la probabilité que la différence de performances mesurée ne soit pas due au hasard. La différence entre les résultats

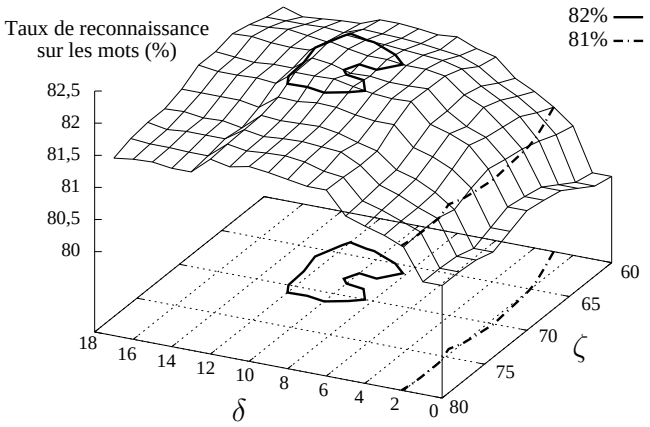



FIGURE 7 – Influence de la prise en compte du nombre de mots des phrases, dans le calcul des probabilités *a posteriori* des mots.

obtenus par une reconnaissance avec le Consensus et une reconnaissance avec l'approche MAP est significative à 99,99 % : cela signifie que les erreurs commises par ces algorithmes sont différentes. Cela confirme notre choix d'utiliser les probabilités *a posteriori* des mots en tant qu'indices de confiance sur la phrase résultat de la reconnaissance par l'approche MAP.

## 5.3 Détection, correction et rejet des erreurs

Dans les expérimentations présentées dans cette section, les probabilités *a posteriori* des mots sont utilisées pour évaluer la confiance sur les mots de la phrase résultat obtenue par l'approche MAP. Nous présentons tout d'abord les résultats obtenus lorsque seule l'étape de correction est utilisée sur les hypothèses d'erreur puis nous y ajoutons le mécanisme de rejet.

### 5.3.1 Correction des mots de probabilité *a posteriori* non-maximale

Dans ce premier cas d'erreur considéré, 3,1 % des mots de la base de test sont détectés comme hypothèses d'erreur par cette approche. Cela correspond à la détection de 14,9 % des erreurs sur les phrases résultats de la reconnaissance par l'approche MAP.

Dans la table 2, nous comparons le taux de reconnaissance obtenu en utilisant l'approche MAP seule à celui obtenu en ajoutant l'approche de détection et de correction des hypothèses d'erreur. Les taux de reconnaissance sont donnés en considérant uniquement, d'une part, les mots sélectionnés (c'est-à-dire détectés comme hypothèses d'erreur) et, d'autre part, les mots présents (c'est-à-dire les hypothèses d'erreur telles que le mot à reconnaître est l'un des deux mots en entrée du classifieur utilisé pour la correction).

TABLE 2 – Taux de reconnaissance des mots non-maximaux.

| Approche de reco. | Mots sélectionnés | Mots présents |
|---|---|---|
| MAP | 35,7 % | 60,2 % |
| Détection/correction | 42,7 % | 72,0 % |

L'utilisation de cette approche de correction permet de diminuer de 29,8 % le taux d'erreur parmi les mots présents.

### 5.3.2 Correction des mots de probabilité *a posteriori* faible

Dans ce deuxième cas d'erreur considéré, 11,8 % des mots de la base de test sont détectés comme hypothèses d'erreur. Cela correspond à la détection de 47,2 % des erreurs.

Dans la table 3, nous présentons les taux de reconnaissance obtenus, pour la correction des hypothèses d'erreur lorsque la confiance sur les mots résultats est considérée comme faible.

TABLE 3 – Taux de reconnaissance pour la correction des mots faibles.

| Approche de reco. | Mots sélectionnés | Mots présents |
|---|---|---|
| MAP | 46,1 % | 83,0 % |
| Détection/correction | 47,8 % | 86,1 % |

La réduction du taux d'erreur sur les mots présents est alors de 18,3 %. La réduction plus faible du taux d'erreur, par rapport au premier type d'erreur, vient de la segmentation de la phrase. En effet, pour prendre en compte les différentes hypothèses de segmentation des phrases, les ensembles de confusion sont maintenant des sous-réseaux. Comme la probabilité *a posteriori* d'un mot correspond à la somme des probabilités des chemins auxquels il appartient, normalisé par la somme des probabilités de tous les chemins, la probabilité *a posteriori* des mots sera généralement plus faible dans ces sous-réseaux de confusion (voir la figure 4(b)). En conséquence, même si les mots des sous-réseaux de confusion sont correctement reconnus, leur probabilité *a posteriori* sera généralement inférieure au seuil $\theta_{post}$ et ces mots seront détectés comme hypothèses d'erreur. Ainsi, un grand nombre des mots détectés sont déjà correctement reconnus, d'où une amélioration plus faible des performances en utilisant l'approche de correction.

### 5.3.3 Rejet des erreurs restantes

La stratégie de rejet permet de distinguer les hypothèses d'erreur qui pourront être corrigées de celles qui ne pourront pas l'être. Afin d'utiliser ce mécanisme utilisant des classifieurs supplémentaires (voir sous-section 4.4), il est tout d'abord nécessaire de choisir ces classifieurs.

### *Choix des classifieurs de rejet*

L'objectif est ici d'obtenir un classifieur qui accepterait toutes les hypothèses d'erreur pouvant être corrigées et qui rejetterait toutes les autres. En pratique, il est rarement possible de trouver un tel classifieur : il faut alors trouver un compromis, en permettant le rejet de certaines hypothèses d'erreur qui auraient pu être corrigées, et ce afin d'augmenter le nombre d'hypothèses d'erreur rejetées et qui sont réellement à rejeter. Les performances du classifieur de correction (en termes de taux de reconnaissance sur les mots) seront généralement diminuées mais nous cherchons également à limiter cette baisse. Nous choisissons alors un classifieur de rejet afin que les performances du système de reconnaissance utilisant ce mécanisme se situent entre celles obtenues avec le classifieur de correction (sans rejet) et celles obtenues sans étape de correction (approche MAP simplement).

Pour cela, nous utilisons, d'une part, une courbe *ROC* qui donne le *TAR* (taux de vraies acceptations) en fonction du *FAR* (taux de fausses acceptations) : cela permet de trouver le compromis entre les vraies et les fausses acceptations, en cherchant à minimiser le *FAR* (c'est-à-dire les hypothèses d'erreur à rejeter mais qui sont acceptées) et à maximiser le *TAR* (c'est-à-dire les hypothèses d'erreur pouvant être corrigées et qui sont bien acceptées). D'autre part, pour contrôler l'impact sur la baisse des performances du classifieur de correction, nous utilisons une courbe *Perf/FAR*, comme proposé dans [13]. Les performances du classifieur de correction sont alors mesurées par le taux de reconnaissance sur les mots présents (voir sous-section 5.3.1). Sur les courbes utilisées par la suite, seuls les meilleurs classifieurs sont représentés (parmi plus de 1 000 classifieurs appris en faisant varier les valeurs des paramètres d'apprentissage).

### *Option de rejet pour les mots de probabilité a posteriori non-maximale*

La figure 8 donne les courbes *ROC* et *Perf*/*FAR*, utilisées pour choisir le classifieur associé au rejet des hypothèses d'erreur du premier type (nous rappelons également les courbes de performance pour l'approche MAP et pour l'approche de correction sans rejet).

Sur chacune des courbes, le classifieur de rejet choisi est encadré. Le *FAR* obtenu par ce classifieur est de 61,5 % et le *TAR* de 90,7 %. Nous remarquons ici l'intérêt d'utiliser conjointement les courbes *ROC* et *Perf*/*FAR* pour choisir le classifieur de rejet. En effet, pour une même valeur de *TAR*, il existe plusieurs classifieurs obtenant des valeurs de *FAR* différentes. Néanmoins, ce n'est pas celui correspondant au *FAR* le plus faible (la valeur du *FAR* étant à minimiser) qui permet d'obtenir les performances les plus élevées.

### *Option de rejet pour les mots de probabilité a posteriori faible*

Les courbes *ROC* et *Perf*/*FAR*, utilisées pour choisir le classifieur de rejet des hypothèses d'erreur du deuxième type, sont données par la figure 9.

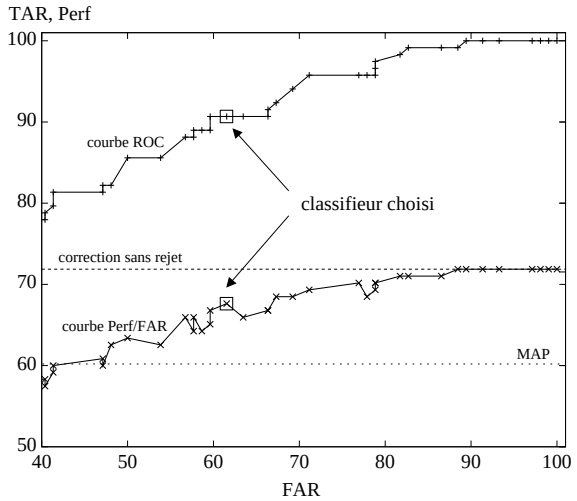



FIGURE 8 – Rejet sur les mots de probabilité non-maximale.

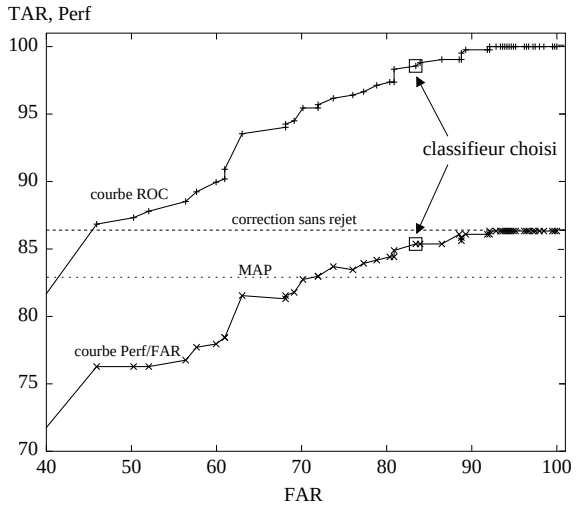



FIGURE 9 – Rejet sur les mots de probabilité faible.

Comme précédemment, le classifieur choisi est encadré. Le *TAR* obtenu par ce classifieur est plus élevé que précédemment (98,6 %) mais le *FAR* est moins bon (83,4 %). Cela vient du fait que la majorité des mots détectés sont déjà correctement reconnus (voir sous-section 5.3.2).

### 5.3.4 Bilan : combinaison des corrections avec le mécanisme de rejet

Nous présentons maintenant les résultats obtenus en combinant les deux stratégies de détection et de correction des hypothèses d'erreur et en y associant ou non leur stratégie de rejet. La détection cumulée des deux types d'hypothèses d'erreur permet de sélectionner 62,1 % des erreurs sur les phrases résultats de la reconnaissance en utilisant l'approche MAP.

La table 4 regroupe les différents taux obtenus. L'augmentation absolue du *taux de reconnaissance*, en utilisant la détection et la correction des hypothèses d'erreur, est de 0,4 % : cela représente une baisse relative du taux d'erreur sur les mots de 3,2 %.

TABLE 4 – Taux pour la correction combinée des erreurs.

| Approche de reconnaissance | Taux de reco. | Taux de rejet | Taux d'erreur |
|---|---|---|---|
| MAP | 86,5 % | 0,00 % | 13,5 % |
| Détect./correct. | 86,9 % | 0,00 % | 13,1 % |
| Détect./correct. + rejet | 86,8 % | 1,7 % | **11,5 %** |

Le taux de reconnaissance atteignable correspond à une bonne correction de toutes les hypothèses d'erreur détectées, en ne considérant que celles qui peuvent être corrigées. Il est ainsi de 88,4 %. Comparativement à ce taux de reconnaissance atteignable, le taux de reconnaissance sur les mots est alors amélioré d'environ 25 % (le rejet n'est pas considéré ici).

Lorsque les *stratégies de rejet* sont prises en compte, l'amélioration du taux de reconnaissance est plus faible (de 0,3 %, en absolu). Cependant, les mots rejetés sont distingués des mots en erreur, le taux de rejet étant de 1,7 %. Le taux d'erreur est ainsi diminué : la baisse relative est alors plus importante, puisqu'elle est de 14,6 %. En utilisant le *paired t-test*, cette amélioration est significative à 99,95 %, par rapport à la reconnaissance MAP.

## 6 CONCLUSION ET PERSPECTIVES

Dans cet article, nous avons présenté un système complet de reconnaissance de phrases manuscrites, en nous focalisant sur l'intégration d'une représentation des hypothèses de reconnaissance de phrases par un réseau de confusion. Ce réseau de confusion a été adapté afin de pouvoir prendre en compte plusieurs hypothèses de segmentation des mots ainsi que le nombre de mots des phrases, lors du calcul des probabilités *a posteriori* des mots. Les probabilités *a posteriori* calculées sur ce réseau de confusion ont été utilisées comme indices de confiance sur les mots des phrases résultats d'une reconnaissance par un décodage MAP. Deux types d'erreurs potentielles ont alors été mis en évidence. Afin de corriger chacun de ces types d'erreurs, un mécanisme de correction utilisant des SVM a été mis en place. Cette étape de correction a permis de diminuer le nombre d'erreurs sur les mots de la

phrase obtenue par l'approche MAP. Enfin, une étape de rejet a été intégrée afin d'identifier les mots détectés comme hypothèses d'erreur mais ne pouvant être corrigés lors de l'étape de correction. Cela a permis ainsi de réduire le taux d'erreur sur les mots de 14,6 %. Ces mots pourront ensuite être mis en évidence ou être soumis à un nouveau traitement.

Dans nos prochains travaux, nous étudierons la détection des mauvaises segmentations, soit directement lors de l'étape de détection des erreurs potentielles de reconnaissance, soit en utilisant les classifieurs de rejet. En effet, les erreurs de segmentation correspondent typiquement aux cas où le mot correct ne se trouve pas dans la liste de mots candidats. De plus, pour la détection des hypothèses d'erreur liées à une probabilité *a posteriori* faible, nous avons vu qu'un seuil fixe n'était pas complètement adapté. Il faudrait alors affiner ce seuil, afin de ne pas détecter trop de fausses erreurs. Enfin, lors de la phase de correction, il pourrait être intéressant de considérer plus de deux mots en entrée des classifieurs dédiés. Comme nous l'avons déjà mentionné, cela nécessitera l'augmentation des bases d'exemples, afin de pouvoir apprendre les classifieurs dédiés de manière fiable.

## RÉFÉRENCES


[1] L.E. Baum, T. Petrie, G. Soules et N. Weiss. A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Function of Markov Chains. *The Annals of Mathematical Statistics*, 41(1) :164–171, 1970.

[2] R. Bertolami, M. Zimmermann et H. Bunke. Rejection Strategies for Offline Handwritten Text Recognition. *Pattern Recognition Letters*, 27 :2005–2012, 2006.

[3] S. Carbonnel et E. Anquetil. Lexicon Organization and String Edit Distance Learning for Lexical Post-Processing in Handwriting Recognition. In *Int. Work. on Frontiers in Handwriting Recognition*, pages 462–467, 2004.

[4] C. Cortes et V. Vapnik. Support-Vector Networks. *Machine Learning*, 20(3) :273–297, 1995.

[5] G.D. Forney. The Viterbi Algorithm. *Proceedings of the IEEE*, 61(3) :268–278, 1973.

[6] W.N. Francis et H. Kucera. *Brown Corpus Manual*. Brown University, 1979.

[7] L. Gillick et S. Cox. Some Statistical Issues in the Comparison of Speech Recognition Algorithms. In *Int. Conf. Acoustics, Signal and Speech Processing*, pages 532–535, 1989.

[8] D. Hillard et M. Ostendorf. Compensating for Word Posterior Estimation Bias in Confusion Networks. In *Int. Conf. Acoustics, Signal and Speech Processing*, pages 1153–1156, 2006.

[9] H. Jiang. Confidence Measures for Speech Recognition : a Survey. *Speech Communication*, 45(4) :455–470, 2005.

[10] T. Kemp et T. Schaaf. Estimating Confidence Using Word Lattices. In *Eurospeech*, pages 827–830, 1997.

[11] M. Liwicki et H. Bunke. Handwriting Recognition of Whiteboard Notes – Studying the Influence of Training Set Size and Type. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 21(1) :83–98, 2007.

[12] L.L. Mangu. *Finding Consensus in Speech Recognition*. PhD thesis, Johns Hopkins University, 2000.

[13] H. Mouchère. *Étude des mécanismes d'adaptation et de rejet pour l'optimisation de classifieurs : application à la reconnaissance de l'écriture manuscrite en-ligne*. PhD thesis, INSA de Rennes, Décembre 2007.

[14] F. Perraud, C. Viard-Gaudin, E. Morin et P.-M. Lallican. Statistical Language Models for On-Line Handwriting Recognition. *IEICE Transactions on Information and Systems*, E88-D(8) :1807–1814, 2005.

[15] J.F. Pitrelli, J. Subrahmonia et M.P. Perrone. Confidence Modeling for Handwriting Recognition : Algorithms and Applications. *Int. Journal on Document Analysis and Recognition*, 8(1) :35–46, 2006.

[16] R. Plamondon et S.N. Srihari. On-Line and Off-Line Handwriting Recognition : A Comprehensive Survey. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(1) :63–84, 2000.

[17] S. Quiniou et E. Anquetil. Utilisation de réseaux de confusion pour la reconnaissance de phrases manuscrites en-ligne. In *Congrès Francophone de Reconnaissance des Formes et Intelligence Artificielle*, pages 101–108, 2008.

[18] S. Quiniou, F. Bouteruche et E. Anquetil. Word Extraction Associated with a Confidence Index for On-Line Handwritten Sentence Recognition. *Int. Journal on Pattern Recognition and Artificial Intelligence*. à paraître.

[19] A. Stolcke. SRILM - An Extensible Language Modeling Toolkit. In *Int. Conf. on Spoken Language Processing*, pages 901–904, 2002.

[20] A. Vinciarelli. A Survey on Off-Line Cursive Word Recognition. *Pattern Recognition*, 35(7) :1433–1446, 2002.

[21] J. Xue et Y. Zhao. Improved Confusion Network Algorithm and Shortest Path Search from Word Lattice. In *Int. Conf. Acoustics, Signal and Speech Processing*, pages 853–856, 2005.

[22] M. Zimmermann et H. Bunke. Optimizing the Integration of a Statistical Language Model in HMM based Offline Handwritten Text Recognition. In *Int. Conf. on Pattern Recognition*, pages 541–544, 2004.