Nouvelle méthode de fusion de données pour l'apprentissage des systèmes hybrides MMC/RNA

Application pour la reconnaissance automatique de la parole

Lilia Lazli et Mohamed Tayeb Laskri

Laboratoire de Recherche en Informatique (LRI)
Groupe de Recherche en Intelligence Artificielle (GRIA)
Département d'Informatique
Faculté des Sciences de l'Ingénieur
Université Badji Mokhtar d'Annaba
B.P.12 Sidi Amar 23000 Annaba – Algérie

L_Lazli@Yahoo.fr laskri@univ-annaba.org

RÉSUMÉ. De nombreuses expériences ont déjà montré qu'une forte amélioration du taux de reconnaissance des systèmes MMC (Modèles de Markov Cachés) traditionnels est observée lorsque plus de données d'apprentissage sont utilisées. En revanche, l'augmentation du nombre de données d'apprentissage pour les modèles hybrides MMC/RNA (Modèles de Markov cachés/Réseaux de Neurones Artificiels) s'accompagne d'une forte augmentation du temps nécessaire à l'apprentissage des modèles, mais pas ou peu des performances du système. Pour pallier cette limitation, nous rapportons dans ce papier les résultats obtenus avec une nouvelle méthode d'apprentissage basée sur la fusion de données. Cette méthode a été appliquée dans un système de reconnaissance de la parole arabe. Ce dernier est basé d'une part, sur une segmentation floue (application de l'algorithme c-moyennes floues) et d'une autre part, sur une segmentation à base des algorithmes génétiques.

MOTS-CLEFS : Reconnaissance de la parole arabe, segmentation floue, algorithmes génétiques, modèles de Markov cachés, réseaux de neurones artificiels, méthode de fusion de données.

Novembre 2005, Numéro spécial CARI'04
Revue ARIMA **pages 125 à 170**

ABSTRACT. It is well known that traditional Hidden Markov Models (HMM) systems lead to a considerable improvement when more training data or more parameters are used. However, using more data with hybrid Hidden Markov Models and Artificial Neural Networks (HMM/ANN) models results in increased training times without improvements in performance. We developed in this work a new method based on automatically separating data into several sets and training several neural networks of Multi-Layer Perceptrons (MLP) type on each set. During the recognition phase, models are combined using several criteria (based on data fusion techniques) to provide the recognized word. We showed in this paper that this method significantly improved the recognition accuracy. This method was applied in an Arabic speech recognition system. This last is based on the one hand, on a fuzzy clustering (application of the fuzzy c-means algorithm) and of another share, on a segmentation at base of the genetic algorithms.

KEY WORDS: Arabic speech recognition, fuzzy clustering, genetic algorithms, hidden Markov models, artificial neural networks, data fusion method.

1. Introduction

La plupart des applications en RAP (Reconnaissance Automatique de la Parole) utilisent la technologie basée sur les modèles statistiques MMC (Modèles de Markov Cachés) qui sont capables de modéliser simultanément les caractéristiques fréquentielles et temporelles du signal vocal. Même si les progrès réalisés par ces modèles sont énormes, ces derniers pèchent par leur manque de capacité discriminante : plus exactement, approcher les vraisemblances conditionnelles avec précision requiert en retour un nombre d'exemples déraisonnables pour réaliser l'apprentissage du système. Or une imprécision sur les vraisemblances se traduit par un mauvais comportement du système, au voisinage des frontières Baysiennes de classification. De plus, la mise en œuvre de ces modèles nécessite des hypothèses contraignantes, ainsi que le coût en temps de calcul et en mémoire.

Le connexionisme a naturellement été appliqué à la reconnaissance de la parole au vu des bonnes capacités en classification de forme. Les Réseaux de Neurones Artificiels (RNA) présentent beaucoup d'avantages potentiels mais sont cependant mal adaptés à traiter les signaux séquentiels. Il a également montré que ces réseaux de neurones (y compris des réseaux récurrents complexes) ne sont pas capables de modéliser les dépendances à long terme, ce que fait par contre très bien un MMC par l'intermédiaire de ces contraintes topologiques (traitant les contraintes phonologiques, lexicales et syntaxiques). Les modèles connexionnistes n'ont pour l'instant pas surpassé les MMC. Ceci est en grande partie due à leur relative inadéquation au problème du traitement séquentiel de l'information. C'est pourquoi une grande tendance dans le domaine du neuromimétisme consiste à étendre les systèmes classiques ou à développer de nouveaux modèles pour prendre en compte les variabilités inhérentes à la parole.

C'est dernières années, il y a eu d'événement fondamental fonçant le connexionisme sur une nouvelle voie. On constate une volonté de dépasser les limitations actuelles du connexionisme. Ainsi, de nouveaux types de systèmes voient le jour, inspirés de la neurobiologie, de la psychologie ou mixant des techniques connexionnistes avec d'autres symboliques [LAZa02], [OSO98], [TOW94] ou stochastiques [BOU89], [OLS02], [BOU90], ces modèles sont couramment appelés *modèles hybrides*. Dans ce sens, l'ère du Perceptron Multi-Couches (PMC) devra évaluer pour dépasser la simple classification de forme.

Plusieurs résultats récents (obtenus sur différentes bases de données allant des petits lexiques aux très grands lexiques) ont montré que les systèmes MMC/RNA conduisent généralement à des performances de reconnaissance équivalentes ou meilleures que celles des systèmes MMC utilisés dans les mêmes conditions, avec cependant plusieurs avantages supplémentaires au niveau des besoins en CPU et mémoire [BOU93], [BOU94], [LAZb03], [OLS02]. En effet, des modèles hybrides MMC\RNA ont été

conçus ces dernières années pour la parole : pour l'Anglais [RII97] et pour le Français [DER97] afin d'additionner les qualités de chacun des modèles fusionnés mais sans réellement homogénéiser l'architecture.

Le traitement de la parole arabe est encore à ses débuts, la raison pour laquelle, nous avons pensé à l'application des modèles hybrides MMC\RNA pour des bases de données allant des petits lexiques aux moyens lexiques, ayant comme objectif la reconnaissance de la parole indépendante du locuteur [LAZb02], [LAZc02]. Nous avons ainsi utilisé dans nos expériences un PMC pour estimer les probabilités a posteriori utilisées pour chaque état du MMC. Pour augmenter le taux de reconnaissance, et pour des fins d'une segmentation acoustique, nous avons proposé deux nouveaux algorithmes : (1) le premier repose sur des concepts de la logique floue : l'algorithme C-Moyennes Floues (CMF) [LAZc03], [LAZd03], [LAZe03]. Nous avons pensé à cet algorithme, vu que l'algorithme classique C-Moyennes (CM) fournit des distributions discrètes assez dures, non probabilisées, qui ne transmet pas assez d'informations sur les observations discrètes. En revanche, l'algorithme CMF proposé permet de classer les données acoustiques en diverses classes selon un degré d'appartenance floue. (2) concernant le deuxième algorithme, nous nous sommes intéressés plus particulièrement, aux méthodes impliquant une classification supervisée par partition et nous avons retenu une comme base pour notre travail. Cette solution consiste à faire le choix d'une mesure que nous utilisons dans notre application. Cet algorithme cherche une "bonne" partition relativement à un critère qui mesure la qualité d'une partition. Nous sommes donc ramenés à un problème d'optimisation. Les propriétés de cet algorithme ne garantissent pas la convergence vers un optimum global, c'est pourquoi nous nous sommes intéressés à une heuristique de type Algorithmes Génétiques (AG), moins susceptibles d'être piégés par les minima locaux et désormais largement employés dans les problèmes d'optimisation.. Si sur un plan théorique, aucun résultat général ne prouve que cette méthode conduise à une solution optimale, en pratique la convergence globale est souvent constatée.

Des expériences préliminaires au niveau du mot, utilisant des vocabulaires de tailles 1200 et 3900 mots sont rapportées. Nous comparons les résultats du système proposé avec ceux d'un système classique utilisant les MMC standards.

Nous avons constaté que ces modèles hybrides MMC\PMC souffrent de nombreux défauts parmi lesquelles le fait que le nombre de paramètres est en quelque sorte borné. En effet, aucune amélioration n'est généralement observée (comme habituellement pour les MMC continus) lorsque le nombre des données d'apprentissage et/ou de paramètres est fortement augmenté.

Pour pallier cette limitation des systèmes hybrides MMC/PMC, nous proposons dans ce papier une nouvelle méthode visant à explorer ce problème. Cette méthode est basée sur des expériences qui ont déjà montré qu'il est possible d'améliorer

sensiblement les performances des systèmes hybrides en combinant plusieurs modèles [LAZa04], [LAZb04]. A la base, l'hypothèse est que, si les modèles sont entraînés sur différentes parties du fichier d'apprentissage, ils vont sélectionner des propriétés différentes des données, permettant ainsi une amélioration des résultats lorsque les sorties sont combinées. D'ailleurs c'est un peu dans cette optique que certains laboratoires travaillant en reconnaissance de la parole entraînent des modèles pour les hommes et pour les femmes. Lors de la reconnaissance, les modèles sont tous les deux utilisés et la sortie correspondant au meilleur score est sélectionnée [GAU94]. Ces modèles sont combinés selon plusieurs critères pour fournir le mot le plus probable.

La figure 1 présente les différentes étapes dans les processus d'apprentissage et de reconnaissance du système proposé. Chacun des éléments présents sur cette figure sera décrit dans ce papier.

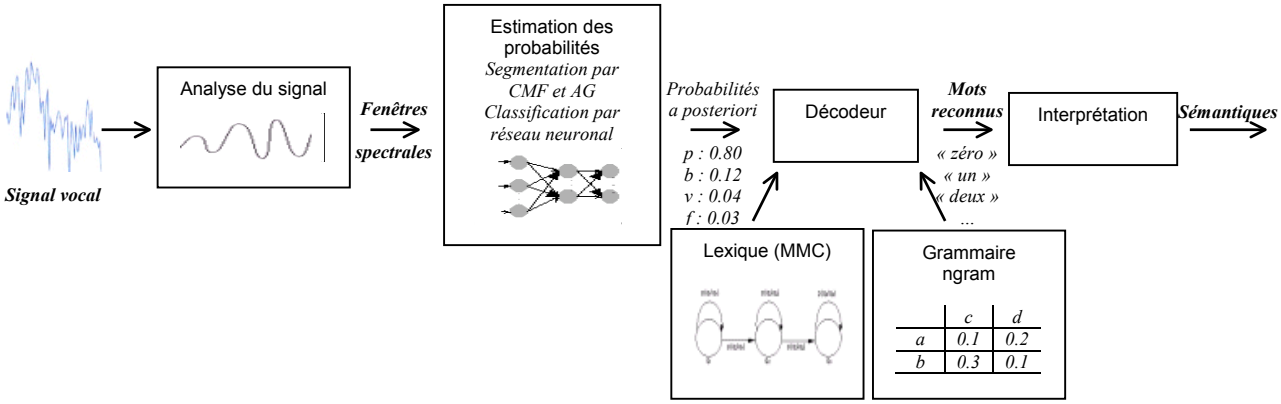

Figure 1. *Une architecture typique du système complet de reconnaissance de parole.*

Le premier bloc (*analyse du signal*) fait l'analyse du signal, dans toutes les expériences décrites dans ce papier, nous avons appliqué l'analyse log RASTA-PLP pour l'extraction de paramètres. Le second bloc (*estimation des probabilités*) calcule les probabilités locales associées à chaque tranche acoustique de 10 ms, l'application de l'algorithme CMF et des AG pour la segmentation des fenêtres d'analyse et l'utilisation des réseaux de neurones de type PMC pour l'estimation des probabilités a posteriori, étant la base de ce deuxième bloc. Ces probabilités sont ensuite envoyées au décodeur qui les intégrera dans le temps (MMC) en utilisant l'information lexicale et les contraintes phonologiques, ainsi que les contraintes grammaticales dans le cas de la reconnaissance de la parole continue. Dans ce cas là, les hypothèses de sons résultants peuvent alors éventuellement être envoyées à un module d'analyse sémantique pour leur interprétation.

Ce travail s'intègre dans l'un des thèmes de recherche menés au sein de GRIA/LRI (Groupe de Recherche en Intelligence Artificielle / Laboratoire de Recherche en Informatique), au département d'Informatique de l'université d'Annaba.

La contribution principale de cet article concerne en premier lieu, la mise en œuvre de nouvelles approches pour la segmentation acoustique de la parole, vu que cette dernière est une phase primordiale pour tout système de RAP. En deuxième lieu, nous propositons d'une nouvelle méthode basée sur la fusion de données pour l'apprentissage du système hybride MMC/PMC proposé.

De ce fait, dans cet article, nous rappelons tout d'abord (section 2) quelques éléments de base des modèles statistiques les plus utilisés en RAP: les MMC. La mise en œuvre de ces derniers nécessite des hypothèses contraignantes qui peuvent pénaliser la performance des MMC. La raison pour laquelle, nous présentons dans la section 3 une définition des modèles hybrides MMC/ANN qui combinent les avantages des modèles de Markov cachés (alignement temporel statistique) et des réseaux de neurones (estimation de la probabilité d'émission d'une observation par un état et entraînement au niveau de la trame). Nous évoquons dans cette section, l'idée de base de l'apprentissage et reconnaissance par les modèles MMC/ANN, après avoir rappelé les principes de la paramétrisation et l'estimation des probabilités a posteriori par le biais d'un réseau neuronal. Nous achevons cette section, en présentant l'apport de ces modèles hybrides. Dans les sections 4 et 5, nous nous intéressons plus particulièrement, aux méthodes impliquant une classification supervisée par partition et nous en retenons deux comme base de notre travail : la première méthode est basée sur les concepts de la logique floue, l'algorithme CMF (présenté dans la section 4), le principe de l'approche des AG choisie comme deuxième méthode est expliqué dans la section 5. Dans ces deux dernières sections, nous esquissons le cadre général de notre solution, avant de la spécifier puis d'en donner une traduction algorithmique. La section 6 introduit les concepts d'une nouvelle méthode d'apprentissage des modèles hybrides MMC/ANN, basée sur l'éclatement des données en plusieurs parties pour entraîner plusieurs réseaux et les recombiner lors de la phase de reconnaissance par différentes méthodes de combinaison. La dernière section (section 7) présente les tests pratiques réalisés sur différentes bases de données ainsi que les résultats obtenus.

2. Modèles de Markov cachés

Dans notre travail, nous avons utilisé des modèles statistiques précis : les modèles de Markov cachés (MMC) [BOU93] qui se sont imposés comme la technologie prédominante en reconnaissance de la parole ces dernières années. Ces modèles se sont avérés les mieux adaptés aux problèmes de la reconnaissance de la parole.

2.1. Problèmes à résoudre

Soit M le modèle de Markov caché associé au son X et constitué d'une concaténation de sous-unités lexicales. La reconnaissance de la séquence de vecteurs acoustiques X s'effectue en trouvant le modèle M qui maximise la probabilité $P(M\backslash X, \lambda)$ (probabilité qu'un modèle M génère une séquence de vecteurs acoustiques X étant donné une série de paramètres λ). Cette probabilité est aussi appelée probabilité *a posteriori*. Malheureusement, il n'est pas possible d'accéder directement à cette probabilité par le processus d'apprentissage des modèles de Markov, mais seulement à la probabilité qu'un modèle donné génèrera une certaine séquence de vecteurs acoustiques $P(X\backslash M)$.

En utilisant la loi de Bayes (1), il est possible de lier ces deux probabilités selon :

$$P(M \mid X) = \frac{P(X \mid M).P(M)}{P(X)} \tag{1}$$

où

- $P(X\backslash M)$ est la vraisemblance de la séquence d'observations X étant le modèle M.
- $P(M)$ est la probabilité a priori du modèle.
- $P(X)$ la probabilité a priori de la séquence de vecteurs acoustiques.

Nous verrons un peu plus tard qu'il est nécessaire de choisir un critère pour l'apprentissage des paramètres $\lambda = \{A, B, \pi\}$:

- Critère de MP: Maximum a Posteriori.
- Critère MV: Maximum de Vraisemblance.

Hypothèses

- **H1 :** On suppose que $P(M)$ peut être calculé indépendamment des observations. Cette probabilité est en effet indépendante de X et peut être estimée à partir du modèle de langage.
- **H2 :** Pour une séquence d'observations connue, $P(X)$ peut être considéré constant, puisqu'il est indépendant du modèle, si les paramètres de ces modèles sont fixés. Ainsi maximiser $P(M \mid X) = \frac{P(X \mid M)P(M)}{P(X)}$ revient à maximiser $P(X\backslash M).P(M)$.

Il faut alors résoudre 3 problèmes liés à ces modèles.

- L'estimation des probabilités : comment calculer $P(X/M)$ et quelles sont les hypothèses nécessaires à propos du modèle pour se définir une série de paramètres utiles pour la reconnaissance ?
- L'apprentissage : étant donné une séquence d'observation X_j associée à leurs modèles de Markov respectifs, comment déterminer les paramètres des modèles afin que chacun ait la probabilité la plus grande possible de générer les séquences d'observations associées ? Comment trouver l'ensemble des paramètres λ qui maximisant $P(M\backslash X, \lambda)$ pour l'ensemble des séquences de vecteurs acoustiques X associé au modèle M[1]. Cette probabilité n'étant pas directement accessible, on préfère maximiser $P(X\backslash M, \lambda)$ (soit utiliser le critère MV plutôt que MP)[2].
- Le décodage : étant donné une séquence de MMC avec leurs paramètres entraînés et une séquence d'observation X, comment trouver la meilleure séquence M_k de modèles de Markov élémentaires pour maximiser la probabilité que M_k génère les observations ?

2.2. Limites des MMC

Malgré que les MMC bénéficient d'algorithmes d'apprentissage et de décodage performants (Algorithmes de *Baum*-Welch, de *Viterbi*)[3] néanmoins, les hypothèses nécessaires à la mise en œuvre de ces algorithmes peuvent pénaliser les performances de ces modèles.

Les principales hypothèses les plus contraignantes sont [BOUb90]:

- Pas de contexte acoustique pris en compte[4]. Aucune corrélation entre les vecteurs acoustiques n'est directement modélisable.

---

[1] Ce critère est discriminant car il minimise le taux d'erreur.

[2] Critère MV n'est plus discriminant ce qui constitue une des premières limitations des MMC.

[3] Les approches les plus utilisées pour les MMC sont basées sur des adaptations de l'algorithme EM (Expectation-Maximisation) appelées :

- Algorithme de *Baum-Welch* : $P(X\backslash M)$ est estimée en tenant compte de tous les chemins possibles.
- Algorithme de *Viterbi* : $P(X\backslash M)$ est estimée en tenant compte du meilleur chemin uniquement.

[4] Une solution à ce problème proposée par *Furui* puis par *Lee* consiste à utiliser les dérivées des vecteurs acoustiques. Une amélioration sensible du taux de reconnaissance est observée, mais ce n'est pas la solution à ce problème.

- Le formalisme est rigide, l'intégration d'autres sources de connaissance (syntaxique, sémantique, etc.) est difficile.
- Apprentissage non discriminant (maximisation de la vraisemblance au lieu des probabilités *a posteriori*).
- Les composantes des vecteurs acoustiques sont supposées non corrélées.
- La séquence des états est un processus de Markov du premier ordre.
- Forme des densités de probabilité fixée (multi-gaussiennes ou discrète).

Notons tout de même que la plupart des systèmes de reconnaissance proposés sur le marché actuellement sont basés sur ce type de technique [BOI99], [RAB89][5].

Certaines de ces hypothèses peuvent être supprimées (ou adoucies) en utilisant conjointement les modèles de Markov et un réseau de neurones. Ces modèles sont appelés modèles hybrides.

3. Modèle hybride MMC\PMC

Ces dernières années, les réseaux de neurones ont pris une part de plus en plus importante dans le monde de la recherche notamment depuis que *Rumelhart* [BOI94] a montré les multiples possibilités des réseaux de neurones à couches multiples. Ils sont en particulier utilisés comme estimateur statistique.

Les différents types des réseaux de neurones : les perceptrons multi-couches, les réseaux récurrent, les réseaux à délais temporels ainsi que les réseaux prédictifs (voir [BOI94] pour plus de détails sur leur fonctionnement) ont été appliqués avec succès à une multitude de problèmes de classification. Les réseaux les plus utilisés en reconnaissance de la parole sont les PMC [BOI94]. Une seule couche cachée est généralement utilisée[6]. l'addition de cette couche cachée permet au réseau de modéliser des fonctions de décision complexes et non linéaires entre n'importe quel espace d'entrée et de sortie. Nous n'utiliserons que ce type de réseau dans les expériences décrites dans ce travail. Nous allons décrire par la suite, comment ils peuvent être utilisés pour la RAP.

[5] Watson d'AT&T, VoiceType d'IBM et Easy Speaking de Dragon Dictate…

[6] En effet, il a été démontré qu'un réseau à plusieurs couches cachées est équivalent à un réseau à une couche cachée de taille plus importante.

3.1. Paramétrisation et estimation des probabilités

Les réseaux de neurones estiment des probabilités a posteriori (et non des vraisemblances comme dans le cas de distributions gaussiennes), il est cependant nécessaire d'adapter la théorie de base des MMC de façon à pouvoir traiter ces probabilités a posteriori. On pourra ainsi inclure proprement le formalisme des RNA dans les algorithmes puissants d'apprentissage et de décodage propres aux MMC. L'apprentissage des RNA étant souvent supervisé, les algorithmes d'apprentissage MMC fourniront alors un contexte général pour l'apprentissage des RNA en nous générant les sorties cibles pour leur apprentissage avec simple supervision au niveau global (séquence de mots).

Le critère optimal pour l'apprentissage et la reconnaissance MMC est basé sur les probabilités a posteriori $P(M_j \mid \mathrm{X},\Theta)$de modèles M_j étant donné une séquence acoustique X et un ensemble de paramètres Θ. Cependant, de façon à faire apparaître les probabilités a posteriori locales, la loi de Bayes n'est pas directement appliquée et la décomposition entre modèles acoustiques et modèle de langage est faite différemment. Nous reproduisons ici brièvement la démonstration relative au développement de la probabilité a posteriori globale présentée dans [BOI99]. Sans hypothèses particulières, on peut effectivement développer la probabilité a posteriori globale selon :

$$P(M \mid \mathrm{X}) = \sum_{Q \in M} P(M, Q \mid \mathrm{X})$$

$$= \sum_{l_1=1}^{L} \cdots \sum_{l_N=1}^{L} P\left(q_{l_1}^1, \ldots, q_{l_N}^N, M \mid \mathrm{X}\right) \tag{2}$$

où la somme sur Q représente tous les chemins de longueur N possibles dans M et $q_{l_n}^n$ l'évènement que l'état q_{l_n} est visité à l'instant n. L'équivalent de l'approximation Viterbi prendra uniquement le chemin de probabilité maximale et les sommes seront remplacées par des opérateurs *max*.

Si nous considérons une séquence d'états particulière, la probabilité a posteriori de cette séquence d'états peut alors se décomposer en un produit de contributions acoustiques et de probabilités a priori.

$$P\left(q_{l_1}^1, \ldots, q_{l_N}^N, M \mid X\right) = P\left(q_{l_1}^1, \ldots, q_{l_N}^N \mid X\right) P\left(M \mid X, q_{l_1}^1, \ldots, q_{l_N}^N\right)$$

$$\cong \underbrace{P\left(q_{l_1}^1, \ldots, q_{l_N}^N \mid X\right)}_{\text{acoustique}} \underbrace{P\left(M \mid q_{l_1}^1, \ldots, q_{l_N}^N\right)}_{\text{prob. a priori}} \tag{3}$$

Dans cette dernière expression, les auteurs ont simplement fait l'hypothèse que le deuxième facteur était conditionnellement indépendant de X, ce qui n'est pas une hypothèse trop forte. En faisant encore l'hypothèse habituelle que les modèles de Markov sont d'ordre 1 et que à l'instant n la dépendance sur X est limitée à un certain contexte acoustique $X_{n-c}^{n+c} = \{x_{n-c},...,x_n,...,x_{n+c}\}$ centré sur x_n, le premier facteur de (3) se simplifie en :

$$\begin{aligned} P\left(q_{l_1}^1,...,q_{l_N}^N \mid X\right) &= P\left(q_{l_1}^1 \mid X\right)P\left(q_{l_2}^2 \mid X, q_{l_1}^1\right)...P\left(q_{l_N}^N \mid X, q_{l_1}^1,..., q_{l_{N-1}}^{N-1}\right) \\ &= \prod_{n=1}^{N} P\left(q_{l_n}^n \mid X, Q_1^{n-1}\right) \\ &\cong \prod_{n=1}^{N} P\left(q_{l_n}^n \mid X_{n-c}^{n+c}, q_{l_{n-1}}^{n-1}\right) \end{aligned} \tag{4}$$

où $Q_1^{n-1} = \left\{q^1, q^2,..., q^{n-1}\right\}$ représente la séquence d'états précédents. En utilisant les mêmes hypothèses, le deuxième facteur de (3) devient :

$$P\left(M \mid q_{l_1}^1,..., q_{l_N}^N\right) = \frac{P\left(q_{l_1}^1,..., q_{l_N}^N \mid M\right)P(M)}{P\left(q_{l_1}^1,..., q_{l_N}^N\right)}$$

$$\cong P(M)\left[\prod_{n=1}^{N} \frac{P\left(q_{l_n}^n \mid q_{l_{n-1}}^{n-1}, M\right)}{P\left(q_{l_n}^n \mid q_{l_{n-1}}^{n-1}\right)}\right] \tag{5}$$

Etant donné ces simplifications, nous pouvons alors approximer (5) selon :

$$P(M \mid X) \cong \sum_{l_1,...,l_N}\left[\prod_{n=1}^{N} P\left(q_{l_n}^n \mid X_{n-c}^{n+c}, q_{l_{n-1}}^{n-1}\right)\frac{P\left(q_{l_n}^n \mid q_{l_{n-1}}^{n-1}, M\right)}{P\left(q_{l_n}^n \mid q_{l_{n-1}}^{n-1}\right)}\right]P(M) \tag{6}$$

et l'approximation Viterbi est obtenue en remplaçant la somme sur les états (l_1,…, l_N) par une maximisation. Cette formulation contient donc maintenant trois ensembles de probabilités a priori :

- $P\left(q_l^n \mid q_k^{n-1}\right)$ qui représente l'information a priori présente dans l'ensemble d'apprentissage.
- $P\left(q_l^n \mid q_k^{n-1}, M\right)$ les probabilités a priori relatives à la topologie du modèle M. En fait, ces probabilités sont équivalentes aux probabilités de transition des MMC standards. En supposant que les paramètres des MMC soient invariants dans le temps, ces probabilités s'écrivent alors $P\left(q_l \mid q_k\right)$ et $P\left(q_l \mid q_k, M\right)$.
- $P(M)$ l'information a priori relative au modèle de langage qui, comme dans les MMC classiques, contient l'information syntaxique.

Les probabilités $P\left(q_k^n \mid X_{n-c}^{n+c}, q_l^{n-1}\right)$, appelées *probabilités de transition conditionnelles*, jouent maintenant le rôle de probabilités d'émission et peuvent être estimées par un réseau de neurones comme représenté à la figure 2 mais dans lequel on a ajouté aux entrées des unités supplémentaires représentant la classe précédente (comme imposée par la structure du modèle de Markov considéré). L'algorithme d'apprentissage associé, qui réalise l'estimation récursive et la maximisation des probabilités a posteriori, appelé REMAP sort du cadre de ce travail et nous renvoyons le lecteur à [BOU94] pour plus de détails.

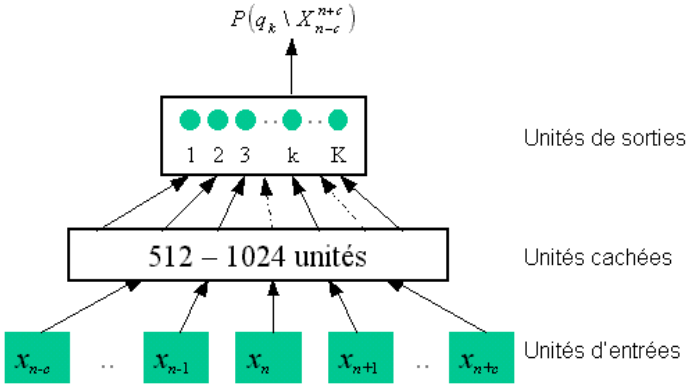

Figure. 2 – *Exemple d'un PMC avec entrée contextuelle et générant des probabilités a posteriori à sa sortie. Dans notre cas, ce type de réseau de neurones est utilisé pour la classification de son entrée (vecteur acoustique à l'instant n dans son contexte) en classes* $q_k\,(k=1,...,K)$. *Par la suite, chacun des états MMC sera associé à une de ces classes.*

En relevant la dépendance par rapport à l'état précédent dans (6), on obtient alors la formulation des systèmes hybrides MMC/RNA la plus couramment utilisée (voir par exemple [BOI99] et [BOUa90]) :

$$P(M \mid X) \cong \sum_{l_1,\ldots,l_N} \left[\prod_{n=1}^{N} P\left(q_{l_n}^n \mid X_{n-c}^{n+c}\right) \frac{P\left(q_{l_n}^n \mid M\right)}{P\left(q_{l_n}^n\right)} \right] P(M) \qquad (7)$$

Grâce à cette équation, nous concluons que les sorties d'un réseau de neurones tel que représenté à la figure 3 et estimant donc les probabilités $P\left(q_l \mid X_{n-c}^{n+c}\right)$ doivent être divisées par une estimation de la probabilité a priori $P(q_l)$ associée à chaque classe q_l. En général, cette probabilité a priori sera simplement estimé sur l'ensemble d'apprentissage comme brièvement discuté dans [BOUa90].

De façon similaire à ce qui a été fait en modèles MMC, nous pouvons définir des récurrences α et β pour l'apprentissage automatique des paramètres du réseau de neurones. Pour simplifier les notations, et de façon à mieux mettre en évidence les relations avec les récurrences MMC, nous négligeons ici le contexte acoustique et supposons que le réseau de neurones ne regarde que le vecteur x_n à l'instant n. Nous avons alors :

$$\alpha_n(l \mid M) = \frac{p\left(X_1^n, q_l^n \mid M\right)}{p\left(X_1^n\right)} = \left[\sum_k \alpha_{n-1}(k \mid M) P(q_l \mid q_k)\right] \frac{P(q_l \mid x_n)}{P(q_l)}$$

$$= \left[\sum_k \alpha_{n-1}(k \mid M) P(q_l \mid q_k)\right] \frac{P(x_n \mid q_l)}{P(x_n)} \qquad (8)$$

$$\beta_n(l \mid M) = \frac{p\left(X_{n+1}^N \mid q_l^n, X_1^n, M\right)}{p\left(X_{n+1}^N\right)} = \left[\sum_k \beta_{n+1}(k \mid M) P(q_k \mid q_l)\right] \frac{P(q_k \mid x_{n+1})}{P(q_k)}$$

$$= \left[\sum_k \beta_{n+1}(k \mid M) P(q_k \mid q_l)\right] \frac{P(x_{n+1} \mid q_k)}{P(x_{n+1})} \qquad (9)$$

Supposant que l'on dispose d'un estimateur de $P(q_k)$, nous pouvons alors calculer les sorties désirées pour l'apprentissage du réseau de neurones :

$$P\left(q_k^n \mid X, M\right) = \gamma_n(k \mid M) = \frac{p\left(q_k^n, X \mid M\right)}{p(X \mid M)}$$

$$= \frac{\alpha_n(k \mid M)\beta_n(k \mid M)}{\sum_l \alpha_n(l \mid M)\beta_n(l \mid M)} \qquad (10)$$

Les sorties du réseau de neurones doivent donc être divisées par une estimation des probabilités a priori $P(q_k)$. Celles-ci peuvent être obtenues soit par simple comptage dans le cas de l'approximation Viterbi soit, comme dans le cas de l'apprentissage Baum-Welch (EM), en les réestimant en fonction de la valeur actuelle des paramètres selon :

$$P(q_k) = \frac{\sum_{n=1}^N p\left(q_k^n \mid X, M\right)}{N} = \frac{\sum_{n=1}^N \gamma_n(k \mid M)}{N} \qquad (11)$$

Comme pour les MMC standard, il est donc possible d'estimer la probabilité $P(M\backslash X)$ à partir de probabilités locales $P(q_k\backslash x_n)$ générées aux sorties d'un réseau de neurones.

3.2. Apprentissage et reconnaissance par modèle MMC/PMC

Le PMC est utilisé comme classificateurs statistiques. Il permet de classifier des vecteurs acoustiques en différentes classes, chaque classe étant associée à un état stationnaire de l'ensemble Q. Le vecteur d'observation x_n est introduit aux entrées du PMC. Si l'ensemble Q contient K états stationnaires, le réseau présentera K nœuds de sortie. Etant donné x_n à l'entrée du PMC, on peut montrer que la sortie k de ce réseau est une estimation de la probabilité locale $P(q_k \backslash x_n)$. En utilisant la loi de Bayes :

$$P(q_k \mid x_n) = \frac{P(x_n \mid q_k) P(q_k)}{P(x_n)} \tag{12}$$

Il suffit de diviser cette probabilité locale par la probabilité a priori $P(q_k)$ pour obtenir un rapport de vraisemblance ("scaled likelihood") $P(x_n \mid q_k) / P(x_n)$. Comme pendant la reconnaissance, $P(x_n)$ est constant et ne modifie en rien la classification, on se ramène au formalisme des MMC présentés précédemment. Ce formalisme permet alors de modéliser le caractère séquentiel du signal de parole. Remarquons que l'architecture du réseau permet aisément d'introduire plusieurs vecteurs acoustiques consécutifs. Il suffit pour cela d'augmenter le nombre d'entrées du PMC. Cela a simplement pour conséquence d'augmenter la dimension des vecteurs d'entrées des perceptrons de la couche cachée.

Durant l'apprentissage, les vecteurs d'observation x_n de l'ensemble d'apprentissage sont consécutivement présentés aux entrées du PMC. L'apprentissage est dit supervisé car on présente également au PMC les sorties désirées de celui-ci. La sortie associée à l'état stationnaire correspondant au vecteur d'entrée est forcée à 1 alors que les autres sorties sont à 0. L'algorithme opère alors par rétro-propagation de l'erreur d'estimation du vecteur de sortie du PMC et utilise la méthode itérative du gradient pour estimer les poids du réseau.

Nous avons montré que la probabilité globale $P(M \backslash X)$ peut s'exprimer en fonction des sorties d'un réseau de neurones estimant $P(q_k \backslash x_n)$. Par conséquent, en plus des avantages caractéristiques des réseaux de neurones, un système MMC/RNA bénéficie aussi de tous les avantages liés aux MMC, à savoir leur capacité à traiter les données séquentielles et leur possibilité d'entraîner l'ensemble des paramètres Θ (les paramètres du réseau de neurones dans le cas MMC/RNA) sans nécessiter la segmentation explicite de la base d'apprentissage en termes de classes de Ω. Comme dans le cas MMC, il est donc possible d'estimer les paramètres Θ par un algorithme de type Baum-Welch ou Viterbi. Nous rapportons ici le mode d'apprentissage à l'aide des fonctions avant-arrière décrit dans [BOI99].

Apprentissage "avant-arrière" (Baum-Welch)

De façon équivalente aux approches MMC, l'apprentissage des modèles a pour but d'estimer les paramètres

$$\Theta^* = \arg\max_{\Theta} \prod_{j=1}^{J} P\left(M_j \mid X_j, \Theta\right) \tag{13}$$

Dans le cas de l'apprentissage "avant-arrière" (Baum-Welch) de systèmes hybrides MMC/RNA, on veut donc estimer l'ensemble des paramètres Θ maximisant (7), c'est à dire :

$$\arg\max_{\Theta} \sum_{l_1,\ldots,l_N} \left[\prod_{n=1}^{N} P\left(q_{l_n}^n \mid X_{n-c}^{n+c}\right) \frac{P\left(q_{l_n}^n \mid M\right)}{P\left(q_{l_n}^n\right)} \right] P(M) \tag{14}$$

Selon le même principe que l'apprentissage des MMC, il est toujours possible de définir une fonction auxiliaire dont la maximisation est équivalente à la maximisation de (13). Nous pouvons alors utiliser une variante de l'algorithme EM pour entraîner un système hybride MMC/RNA :

- ***Initialisation***
  - Choisir un réseau de neurones initial (ensemble de paramètres) et une distribution de probabilités a priori $P(q_k)$ initiale, ou
  - estimer les paramètres initiaux $\Theta^{(0)}$ du réseau RNA et les probabilités a priori à partir d'une segmentation initiale. Dans notre travail et pour ne pas avoir à annoter une base de fenêtres d'analyse, ce qui réduirait l'intérêt des MMC, des MMC discrets ont été appris, permettant d'associer chacun des leurs états à des parties de mots à analyser. Ainsi par programmation dynamique (algorithme de Viterbi) on peut aligner les observations sur les états et annoter une base d'apprentissage pour une première génération du RNA choisi (PMC dans notre cas).
- ***Etape d'estimation*** **(E)** : étant donné les paramètres $\Theta^{(t)}$ d'un réseau de neurones à l'itération t et les estimateurs de probabilités a priori $P(q_k)$, calculer
  - les nouvelles sorties désirées $P\left(q_k^n \mid X, M, \Theta^{(t)}\right)$(10) associées à chaque vecteur d'apprentissage x_n, grâce aux récurrences α (12) et β (9) calculées à partir des sorties du réseau de neurones et des estimateurs courants de probabilités a priori $P(q_k)$;
  - les nouveaux estimateurs des probabilités a priori (11).
- ***Etape de maximisation*** **(M) :** nouvel apprentissage (EBP) du réseau de neurones avec $P\left(q_k^n \mid X, M, \Theta^{(t)}\right)$(14) comme sortie désirée. On peut montrer que cet apprentissage RNA conduit à un nouvel ensemble de paramètres $\Theta^{(t+1)}$ maximisant la fonction auxiliaire et garantissant donc que :

$$\prod_{j=1}^{J} P\left(M_j \mid X_j, \Theta^{(t+1)}\right) \geq \prod_{j=1}^{J} P\left(M_j \mid X_j, \Theta^{(t)}\right)$$

- ***Itérer*** **:** la convergence de ce processus itératif peut être démontrer en prouvant que (1) les nouvelles sorties désirées maximisent $P(M\backslash X)$ pour un ensemble de paramètres donnés et (2) que l'algorithme de gradient EBP, en convergeant vers ces sorties désirées, vont dans la direction d'un accroissement de $P(M\backslash X)$.

Reconnaissance

La reconnaissance de la parole par système hybride MMC/RNA se pratique selon le même principe que la reconnaissance MMC. Comme illustré dans la figure 3, le réseau de neurone est simplement utilisé comme estimateur de probabilités locales pour les MMC. Après division par les estimateurs de probabilités a priori, le réseau de neurones (PMC dans notre cas) fournit donc les vraisemblances normalisées $p(x_n\backslash q_k)/p(x_n)$ qui sont utilisées, soit dans un algorithme Viterbi, soit dans une récurrence "avant" (8), afin d'estimer $P(M_j\backslash X)$ pour tous les modèles M_j possibles et d'assigner la séquence X au modèle M_k conduisant au maximum de probabilités a posteriori.

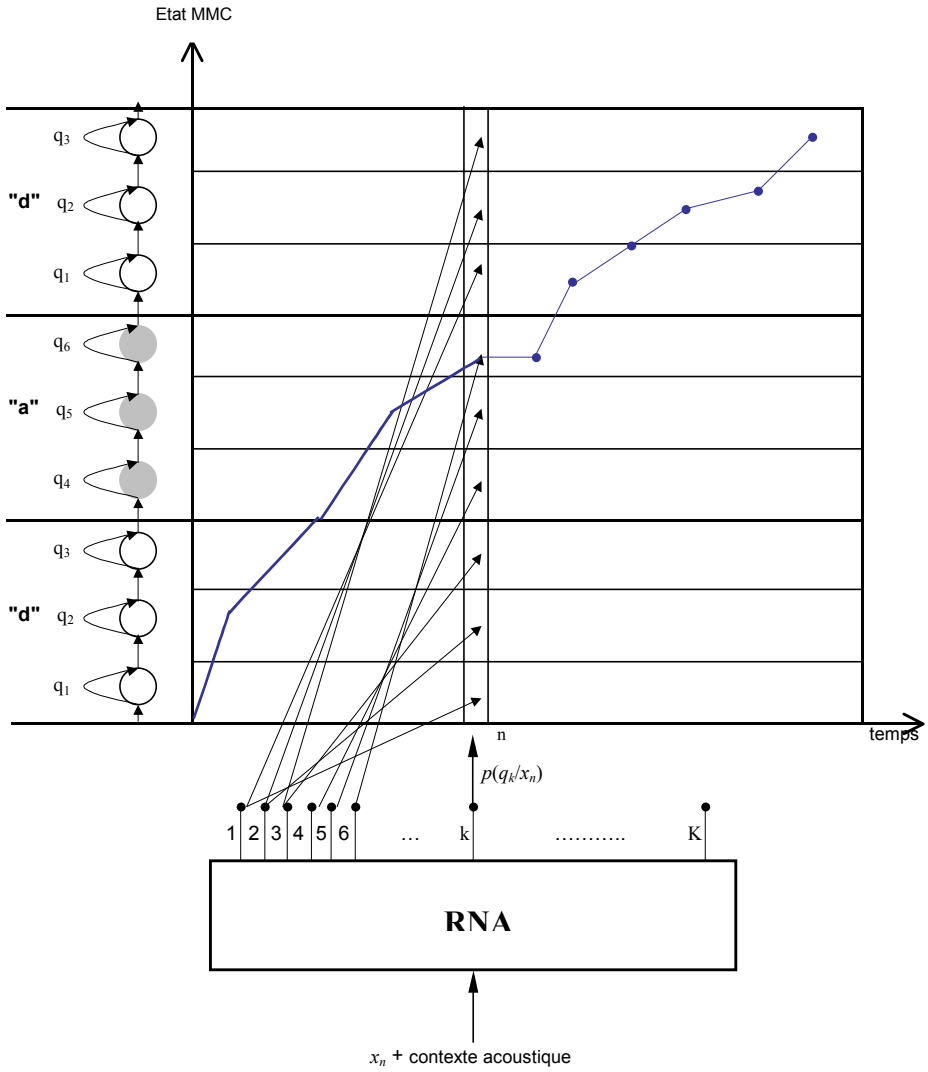

Figure 3. *Schéma de fonctionnement d'un système hybride MMC/RNA : pour chaque vecteur x_n (dans son contexte) présenté à l'entrée du réseau de neurones, celui génère l'ensemble des probabilités $P(q_k \backslash x_n)$ qui après division par les probabilités a priori $P(q_k)$ estimées sur l'ensemble d'apprentissage, donnent les vraisemblances locales requises pour les différents états MMC.*

3.3. Apport de l'hybridation MMC/RNA

Plusieurs résultats récents (obtenus sur différentes bases de données allant des petits vocabulaires aux très grands vocabulaires) ont montré que ces systèmes conduisent généralement à des performances de reconnaissances significativement meilleures que celles des systèmes classiques utilisés dans les mêmes conditions [BOU89], [BOU94], [DER97], [LAZa03], [OLS02], [RII97].

Il faut noter qu'actuellement, il y a une tendance vers l'utilisation des MMC avec l'apprentissage discriminant du type MMI ou MCE, ces derniers contrairement aux MMC usuels, qui estiment les vraisemblances conditionnelles, au moyen du principe du maximum de vraisemblance (MLE), permettent d'estimer les probabilités a posteriori de mots ou de séquences de mots, à l'aide du principe du maximum de probabilité a posteriori (MAP). L'algorithme d'apprentissage associé, sort du cadre de ce travail et nous renvoyons le lecteur à [BOI94] et [BOI99] pour plus de détails.

Les avantages principaux des modèles hybrides MMC/ANN sont les suivants :

- Modèle ne nécessitant pas d'hypothèses sur la forme des distributions (gaussienne ou multi-gaussiennes) statistiques associées à chaque état des MMC. En effet, il a été démontré en théorie et en pratique que l'apprentissage du réseau de neurones permettait d'estimer des distributions statistiques de n'importe quelle forme.

- Du fait de l'apprentissage discriminant des réseaux de neurones (ce qui est une de leurs propriétés majeures), on aboutit à des MMC avec discrimination locale (au niveau de la fenêtre d'analyse).

- D'autre part, l'utilisation de l'information temporelle est plus aisée avec ce type de système : il est facile de fournir plusieurs vecteurs acoustiques à l'entrée du réseau de neurones. Une information contextuelle est donc prise en compte dans les probabilités estimées et la corrélation entre des fenêtres successives n'est pas négligée. Pour diverses raisons, cela n'est pas possible avec des MMC classiques.

4. Distributions discrètes floues

Plus généralement, en classification acoustique de la parole, il est difficile de déterminer nettement les frontières entre unités élémentaires (les phonèmes par exemple) ayant une tendance à appartenir à plusieurs classes à la fois, du fait que la réalité peut ne pas se laisser aussi facilement découper en sous-ensembles dont l'intersection est nécessairement vide. On peut très bien trouver des fenêtres acoustiques dont lesquelles les segments peuvent se recouvrir (cas des fenêtres d'analyse appartenant aux chiffres "*quatre*" et "*sept*" prononcés en arabe), sans avoir rendu

compte des exigences propres à la classification hiérarchique où si deux segments se recouvrent, l'un doit être un sous-ensemble parfait de l'autre.

Ces dernières décennies ont vu le développement des méthodes de segmentation à chevauchement pour répondre à cette limite. Cependant, ces méthodes ont été peu fréquemment utilisées, puisqu'elles ont plusieurs inconvénients pratiques : (1) elles produisent généralement trop de segments avec trop de chevauchements ; (2) les segments sont trop souvent inclus les uns dans les autres et/ou se chevauchent trop de sorte que le résultat final est simplement une liste de segments difficiles à interpréter.

De nouvelles approches de classification ont été proposées pour essayer de surmonter ce problème parmi lesquelles, on peut noter l'approche par la logique floue [BEZ81], [BEZ99] avec l'introduction du concept de degré d'appartenance qui détermine la "force" avec laquelle un individu (fenêtres acoustiques dans notre cas) appartient aux différentes classes. Cela repose sur le fait que le concept de la logique floue ne cherche pas un point de rupture x qui décide de l'appartenance d'un individu à une classe, mais quelle raisonne plutôt sur la base d'un intervalle de valeurs.

Comme évoqué ci-dessus, l'idée qui soutient l'approche par la logique floue est la possibilité d'appartenance à la fois à plusieurs classes pour une fenêtre acoustique. Toutes les méthodes de classification "dure" (parmi lesquelles, la méthode de c-moyennes utilisée pour le calcul des distributions $P(x_n \backslash q_k)$ dans le cas de notre système hybride MMC/PMC) contraignant les fenêtres acoustiques à être membre d'une, et une seule classe, se trouvent ainsi exclues. Bien que la probabilité d'appartenance des objets à plusieurs classes ne soit pas une exclusivité des techniques floues, nous avons choisi de retenir ces dernières car elles fournissent une matrice des degrés d'appartenance de chaque fenêtre acoustique à chaque classe.

Ce qui n'est pas le cas des autres analyses de segmentation avec chevauchement. En effet, ces techniques de segmentation à chevauchement, bien qu'elles soient capables d'apporter une image chevauchée des segments, ne permettent pas d'établir un indice susceptible de révéler les mouvements tendanciels des individus entre classes faute d'informations disponibles. L'approche par la logique floue en segmentation acoustique de la parole, se justifie donc grâce à sa capacité d'engendrer une matrice des degrés d'appartenance.

4.1. Degré d'appartenance

"*Très souvent, les classes d'objet rencontrées dans le monde physique ne possèdent pas de critères d'appartenance bien définis*". Ce constat montre le fossé qui sépare les représentations mentales de la réalité et les modèles mathématiques usuels à base de variables booléennes vrai/faux. En effet, il est difficile de proposer un seuil en deçà duquel l'observation sera affectée entièrement à telle ou telle classe.

Nous avons adapté l'idée de J.*C. Bezdek* [BEZ81], [BEZ99] pour réaliser une classification floue des fenêtres acoustiques. Le résultat de cette classification floue sera utilisé pour calculer les probabilités a posteriori. L'idée était qu'au lieu de chercher à tout prix un seuil unique s décidant l'appartenance à un ensemble dans un contexte donné, il semble plus réaliste de considérer deux seuils $s_1 < s_2$, avec une fonction d'appartenance donnant à chaque fenêtre un degré d'appartenance (compris entre 0 et 1) selon lequel la fenêtre acoustique en question appartient à une classe donnée. En deçà de s_1, la fenêtre acoustique appartient complètement à une classe (degré d'appartenance maximal égal à 1) ; au-delà de s_2, elle n'appartient plus à cette classe (degré d'appartenance minimal, par convention égal à 0). Entre s_1 et s_2, les degrés d'appartenance seront intermédiaires (entre 0 et 1).

Le concept de sous ensemble flou et le degré d'appartenance ont été introduits pour éviter les passages brusques d'une classe à une autre et autoriser les éléments à n'appartenir complètement ni à l'une ni à l'autre ou encore à appartenir partiellement à chacune. Ces notions permettent de traiter : des catégories aux limites mal définies, des situations intermédiaires entre le "tout" et le "rien", le passage progressif d'une propriété à une autre, ou encore des valeurs approximatives exprimées en langage naturel.

Parmi les techniques de la logique floue en classification, l'algorithme C- Moyennes Floues (CMF) a été choisi pour son autonomie due à l'usage d'un classificateur non supervisé. Les autres [PHA99] méthodes comme les k-plus proches voisins flous ou celle fondée sur les relations floues sont tous des algorithmes de classification supervisée réclamant un échantillon d'apprentissage. On va présenter dans ce qui suit le principe de cet algorithme de classification très populaire, basé sur la logique floue, connu pour son efficacité et sa robustesse.

4.2. L'algorithme des c-moyennes floues

L'algorithme des C-Moyennes (CM) est l'une des méthodes les plus connues parmi les techniques de classification non supervisée et qui est fréquemment utilisée pour la quantification vectorielle de la parole. La version c-moyennes floues est une extension directe de cet algorithme, où l'on introduit la notion d'ensemble flou dans la définition des classes. Comme leurs homologues "durs", cet algorithme utilise un critère de minimisation des distances intra-classes et de maximisation des distances inter-classes, mais en tenant compte des degrés d'appartenance des trames acoustiques.

L'algorithme CMF est un algorithme de classification floue fondé sur l'optimisation d'un critère quadratique de classification où chaque classe est représentée par son centre de gravité [BEZ81]. L'algorithme nécessite de connaître le nombre de classes au préalable et génère les classes par un processus itératif en minimisant une fonction objectif. Ainsi, il permet d'obtenir une partition floue de la parole en donnant à chaque fenêtre d'analyse un degré d'appartenance à une région donnée.

Les principales étapes de l'algorithme des c-moyennes floues sont :

1. La fixation arbitraire d'une matrice d'appartenance $[u_{ij}]$ où u_{ij} est le degré d'appartenance de la fenêtre d'analyse *i* à la classe *j*.

2. Le calcul des centroïdes des classes.

3. Le réajustement de la matrice d'appartenance suivant la position des centroïdes.

4. Le calcul du critère d'évaluation de la qualité de la solution, la non convergence de ce critère impliquant le retour à l'étape 2.

La partition floue obtenue est présentée sous forme d'une matrice $N * C$, où N est le nombre des fenêtres acoustiques, C le nombre de classes obtenues. Contrairement aux méthodes de classification dure, la valeur d'appartenance d'un objet à une classe ne prend pas simplement les valeurs 0 ou 1, mais toutes les valeurs possibles dans l'intervalles [0, 1].

Pour avoir une bonne partition, on impose aux éléments de la matrice U (u_{ij}) les contraintes suivantes :

- $u_{ik} \in [0,1]$;
- $\sum_k u_{ik} = 1$; ceci $\forall i$.

L'algorithme du CMF fait évoluer la partition (Matrice U) en minimisant la fonction objectif suivante :

$$J_m(U,C) = \sum_{i=1}^{N} \sum_{k=1}^{C} (u_{ik})^m . \|x_i - c_k\|^2 \qquad (15)$$

où :

- $m > 1$ est un paramètre contrôlant le degré de flou (généralement $m = 2$);
- c_k : le centre de la classe k.

Algorithme CMF

1. Choisir le nombre de classes : C // information a priori, algorithme supervisé.
2. Initialiser la matrice de partition U, ainsi que les centres c_k.
3. Faire évoluer la matrice de partition et les centres suivant les deux équations :

1) $u_{ik} = \dfrac{1}{\sum_{j=1}^{c}\left(\dfrac{d_{ik}}{d_{ij}}\right)^{\frac{2}{m-1}}}$, // mise à jour des degrés d'appartenance où :

$d_{ij} = \left\| x_i - c_j \right\|$,

2) $c_k = \dfrac{\sum_i u_{ik}^m . x_i}{\sum_i u_{ik}^m}$, // mise à jour des centres

4. Test d'arrêt : $|J^{t+1} - J^t| < seuil$.

Le résultat direct fourni par l'algorithme CMF est la matrice des degrés d'appartenance de chaque fenêtre à chaque classe. Cette matrice donne déjà une image graduée de l'appartenance des fenêtres d'analyse aux classes ainsi qu'une image du chevauchement des classes. On peut très bien arrêter ici l'algorithme en affectant la fenêtre à la classe la plus plausible, mais le problème de l'opposition entre la taille des segments et leur degré d'homogénéité reste entier.

5. Algorithmes génétiques en classification supervisée par partition

Le problème du partitionnement d'un ensemble de *n* objets en *k* classes, telle que la distance entre les objets d'une classe et leur centre soit minimale est donc un problème d'optimisation. Celui-ci est difficile, compte tenu de l'existence de nombreux minima locaux. Les méthodes classiques (c-moyennes, nuées dynamiques, etc.) résolvant ce problème sont des méthodes quasi-optimisantes car elles peuvent conduire à ces minima locaux. Nous rappelons ici leurs principaux défauts :

- Nécessité de connaître a priori, le nombre *k* de classes.
- Sensibilité aux conditions initiales : choix de la configuration initiale et ordre de traitement des instances.
- Convergence vers des minima locaux.

La principale limitation de ces méthodes de partitionnement, nous a poussé de s'intéresser à d'autres méthodes. Or il se trouve que les algorithmes génétiques semblent particulièrement aptes à traiter le problème d'optimisation, moins susceptibles d'être piégés par les minima locaux et désormais largement employés dans les problèmes d'optimisation..

Leur principe basé sur une description fidèle d'une évaluation naturelle, assure une recherche efficace dans le monde des solutions d'un problème donné. Pour qu'ils

puissent surpasser leurs cousins plus classiques dans la quête de la robustesse, les AG sont fondamentalement différents selon quatre axes principaux :

1. Les AG utilisent un codage des paramètres, et non les paramètres eux mêmes.
2. Les AG travaillent sur une population de points, au lieu d'un point unique, c'est une des grandes forces des algorithmes génétiques.
3. Les AG n'utilisent que les valeurs de la fonction étudiée, pas sa dérivée, ou une autre connaissance auxiliaire.
4. Les AG utilisent des règles de transition probabilistes, et non déterministes.

5.1. Principes des AG

Les AG permettent de réaliser une recherche multi-dimensionnelle dont le but est de trouver une valeur optimale au sens d'une fonction de *mérite* dans un problème d'optimisation. Ils utilisent un parallélisme implicite en ce sens qu'ils manipulent simultanément de multiples solutions dont ils calculent les valeurs d'*adaptation*.

La résolution d'un problème d'optimisation par un AG s'effectue sur une *population* d'*individus*, chacun représentant une solution potentielle au problème. Ces individus sont encodés par une chaîne binaire, entière ou réelle de longueur finie (mais qui peut être variable dans le temps) appelée *chromosome* ou *génotype*.

Dans la pratique les opérateurs de reproduction de croisement et de mutation sont très variés, aussi la modélisation d'un AG est-elle un problème complexe ? Cependant, il existe quelques résultats théoriques dont le théorème des *schèmes* [BAL99], dont nous donnons juste le résultat. Les hypothèses de cette modélisation sont les suivantes : on se limite à un algorithme génétique *simple* où les chaînes (séquences) sont binaires avec sélection de type *roulette*. Le croisement simple (à un point) et la mutation est uniforme. En substance, ce théorème nous dit que les schèmes de longueur utile courte, d'ordre faible et dont la valeur d'adaptation est supérieure à la moyenne de la population sont favorisés lors de la génération d'une nouvelle population. De fait, cela permet de justifier pourquoi un AG fonctionne.

Une construction d'un algorithme génétique peut être structurée comme suit:

- Une représentation chromosomique des solutions du problème.
- Une méthode de création de la population initiale des solutions.
- Une fonction d'évaluation qui permet d'évaluer les solutions plus ou moins "adaptées" (fitness).
- Des opérateurs génétiques qui modifient la composition des chromosomes des parents au cours de la reproduction.

- Les valeurs de paramètres que les algorithmes génétiques emplois (taille de la population, probabilité d'adaptation des opérateurs génétiques, etc.) qui peuvent influer sur la vitesse de convergence de l'algorithme.

5.2. Algorithmes génétiques pour la segmentation de la parole

Dans le but d'une segmentation automatique de la parole, nous avons appliqué un type particulier des AG afin de réaliser une classification supervisée par partition. Le déroulement typique de cet algorithme est défini comme suit : (voir figure 4)

Algorithme AG en classification supervisée par partition

Entrée

vecteur d'apprentissage x_i /* Après l'analyse CPL(Codage Prédictif Linéaire), nous obtenons des matrices (X) dont chacune d'eux est composée d'un ensemble de fenêtres (L fenêtres), chaque fenêtre est représentée par un vecteur x_i (i représente le rang de la fenêtre) de p (p = 12) paramètres */

Initialisation

Soient :

– Une population de N solutions (notées A_i^k. $\forall\ i \in [1, N]$, k désigne l'étape, N la taille de la population), et les valeurs de la fonction objectif de coût pour les N solutions (notée J_i^k)

– J_p^0 : la valeur de la fonction objectif W pour $p^{ième}$ chromosome de la population initiale.

– t_r : le taux de remplacement de la population.

– P_c : la probabilité de croisement.

– P_m : la probabilité de mutation.

– *nbiter* : le nombre d'itérations.

– $J_h = \min_{1 \le i \le N}\left(J_i^0\right)$ et A_h^0 la solution correspondante.

– $k \leftarrow 1$.

Répéter

Répéter

Sélection de 2 parents.

Générer 2 enfants viables par croisement des 2 parents selon une probabilité P_c.

Obtenir 2 enfants viables par mutation de chacun selon une probabilité P_m.

Jusqu'à taille (nouvelle population) = $N.t_c$

Remplacer les $N.t_c$ **"**pires**"** individus de la génération courante par les $N.t_c$ nouveaux individus.

Evaluer leur adaptation. On a alors **:**

$$\begin{pmatrix} A_1^k \ldots A_N^k \\ J_1^k \ldots J_N^k \end{pmatrix}$$

$J_k \leftarrow min_{1 \le j \le p}(J_j^k)$ et A_j^k c'est la partition associée**.**

$J_h \leftarrow min\ (J_k, J_h)$

$k \leftarrow k+1$.

Jusqu'à (k = *nbiter***)**

Figure 4. *Structure algorithmique de l'AG proposé en classification supervisée par partition*

Dans ce qui suit, et pour des fins d'éclaircissement du fonctionnement des AG, les expériences reportées sont effectuées uniquement sur une partie de la base de test (composée de chiffres arabes).

5.2.1. Codage des individus

L'idée clé est que le codage d'un individu doit permettre un échantillonnage "efficace" de l'espace de recherche. Nous avons retenu une représentation fondée sur l'indexage des classes. Un individu est donc encodé par une chaîne d'entiers dont la valeur ordinale (= indice) représente le numéro de l'objet et la valeur cardinale correspond au numéro de la classe auquel appartient l'objet.

Considération pratique : La partition suivante des objets (vecteurs acoustiques) en 10 classes (10 chiffres arabes représentant une des bases de test) :
*classe*0 = $\{V_1,V_2,\ldots..V_{21},V_{22}\}$, *classe*1 = $\{V_{23},V_{24},\ldots V_{34},V_{35}\}$, *classe*2 = $\{V_{36},V_{37},\ldots V_?\ldots\}$, …, *classe* 9 = {…} sera représentée par le chromosome suivant : (voir figure5)

000000 …	**111111…**	**222222…**	**333333…**	**444444…**	**555555…**	**666666…**	**777777…**	**888888…**	**999999…**

Figure 5. *Exemple de codage des individus*

5.2.2. Taille de la population

Il s'agit là, d'un paramètre fondamental de l'AG, pour lequel il n'existe pas de critères généraux permettant de savoir quelle taille optimale il conviendrait de prendre. Nous avons tenté d'adapter la taille de la population à la taille du problème à traiter.

5.2.3. Fonction de mérite

Un individu ou encore génotype représente, rappelons le, un partitionnement de l'ensemble des observations en k classes. Nous utilisons la fonction de représentation suivante :

$$g_l = \frac{1}{|C_l|} \sum_{x_i \in C_l} x_i \qquad (16)$$

avec g_l le centre de gravité de la classe C_l, $l \in [1, M]$, M le nombre de classes (chiffres dans notre cas), ce qui permet de déduire les M représentants. Le critère d'inertie intra-

classe, mesure la qualité d'une partition dont l'expression est donnée en (17) est notre *fonction objectif* de coût, et nous cherchons à en minimiser la valeur. La transformation (selon critère) de cette fonction objectif conduit à la *fonction de mérite* que nous utilisons dans notre algorithme pour mesurer la qualité d'un individu.

$$w = \sum_{l=1}^{M} \sum_{x_i \in C_l} p_i \, d^2 \, (x_i \, , g_l \,) \tag{17}$$

où p_i désigne le poids du $i^{\text{ième}}$ objet.

5.2.4. Reproduction

Dans cette phase, on crée à chaque itération, une nouvelle population, en appliquant les opérateurs génétiques : sélection, croisement et mutation. Elle consiste à sélectionner, suivant une méthode, les individus les plus adaptés dans la population au sens de la fonction de *mérite*, et à les reproduire tels quels dans la génération suivante.

Dans notre AG nous avons utilisé la variante baptisée *"steady state genetic algorithm"* consistant à ne remplacer qu'un certain pourcentage de la population à chaque génération (la taille de la population reste donc constante au cours du déroulement de l'AG). L'évolution de la population est assurée au moyen des opérateurs de la *sélection*, le *croisement* et la *mutation* qui permettent de combiner et modifier les chromosomes. Nous résumons ici, les opérateurs génétiques utilisés.

- **Sélection :** Il existe de nombreuses stratégies de sélection, nous en présentons juste deux brièvement. Dans la sélection de type *roulette* la probabilité de sélection d'un individu est proportionnelle à sa valeur d'adaptation. La sélection de type *tournoi* consiste à sélectionner un sous-groupe de la population et à conserver le meilleur individu de ce sous-groupe. Vu que dans notre cas, le problème est un problème de minimisation de la fitness. Le meilleur individu est celui qui possède la plus petite valeur de fitness.

Considération pratique : Selon la valeur d'adaptation de chaque chromosome de la population, on sélectionne les meilleurs individus. Dans le tableau 1, la sélection des meilleurs individus est réalisée selon l'ordre ascendant de leurs valeurs de fitness.

Tableau 1. *Sélection des meilleurs individus*

Parents	fitness
1	6.4520
2	6.8843
3	7.3374
4	6.2424
5	6.7076
6	7.1452
7	6.4746
8	6.8331

Parents	fitness
1	6.2424
2	6.4520
3	6.4746
4	6.7076
5	6.8331
6	6.8843
7	7.1452
8	7.3374

- **Croisement :** Il permet de créer deux individus (enfants) en combinant les gènes des deux parents obtenus à l'étape précédente. Cet opérateur permet d'augmenter en moyenne la qualité d'une population. Nous avons utilisé la variante à un point de croisement aléatoirement choisi parmi les points de coupe possibles (pour une chaîne de longueur *l*, il y en a *l* - 1). Cette solution peut toutefois conduire à un enfant non *viable* (voire tous les deux), c'est pourquoi on teste éventuellement un à un les différents points de coupure. On peut éventuellement limiter ce parcours en fixant a priori, un certain nombre d'itérations. Si à la fin de cette vérification, l'enfant reste non viable, on prend l'un des parents. Un enfant est dit non viable, s'il ne possède pas le même nombre de classes que ces parents.

Considération pratique: Soient les deux parents suivants choisis de notre application (voir figure 6).

Parent 1

Point de coupure

0000000000…	1111011111…	2222222222…	3363333333…	4474447744…
5555665555…	6666556666…	7777677777…	8833882288…	9999999099…

Parent 2

0000000000…	1111111551…	2222332222…	3333333333…	4444444444…
5555656556…	6666666666…	7777777777…	8888883388…	9999999999…

Figure 6. *Deux parents*

On obtient le point de coupe à la fenêtre acoustique n° 102. Les deux enfants obtenus sont les suivants (voir figure 7) :

Enfant 1

0000000000…	1111011111…	2222222222…	3363333333…	4444444444…
5555656556…	6666666666…	7777777777…	8888883388…	9999999999…

Enfant 2

0000000000…	1111111551…	2222332222…	3333333333…	4474447744…
5555665555…	6666556666…	7777677777…	8833882288…	9999999099…

Figure7. *Deux enfants*

En ce qui concerne la valeur de la probabilité de croisement, nous avons suivi les recommandations proposée par Goldberg [GOL94], qui citant les travaux de *De Jong*, propose une probabilité de croisement élevée (≥ 0.5).

- **La mutation :** Cette étape permet d'introduire une variation aléatoire dans le génotype de l'individu (ici les enfants). Cet opérateur permet donc d'explorer de nouvelles régions de l'espace de recherche, diminuant ainsi les risques de converger vers des minima locaux.. Chaque *gène* est donc susceptible de changer selon une probabilité donnée, il est aussi possible de ne choisir aléatoirement qu'une partie des

gènes. La mutation peut conduire à un individu non viable, cette éventualité est donc prise en compte au niveau du codage de cet opérateur.

Considération pratique : Soient les gènes de l'enfant1 présenté précédemment (voir figure 8):

Enfant avant mutation

0000000000…	1111011111…	2222222222…	3363333333…	4444444444…
5555656556…	6666666666…	7777777777…	8888883388…	9999999999…

Enfant après mutation

0000000000…	1111011111…	2222222222…	3363333333…	4444444444…
5555656556…	6666666666…	7777777777…	8888883388…	9799999999…

↑Changement de gène

Figure 8. *Exemple de mutation*

Là encore, nous avons suivi les recommandations de *Goldberg* [GOL94] quant à la probabilité de mutation, en adoptant une valeur inversement proportionnelle à la taille de la population.

5.2.5. Remplacement de la nouvelle population

Nous avons procédé à un remplacement des mauvaises solutions. Dans ce but, nous avons classé toutes les solutions (parents et enfants) selon leur adaptation et on ne garde dans notre population que les *N* premiers individus, c'est-à-dire de plus petites valeurs de fitness obtenant ainsi, la nouvelle population qui est constituée des *N* meilleurs adaptations. Si le nombre de générations est atteint, alors on extrait la meilleure solution, et on passe à l'étape suivante ; sinon, on fait une nouvelle itération de reproduction.

Considération pratique : La population des parents et enfants obtenue, est présentée dans le tableau 2 à gauche. Pour faire un remplacement des individus les plus mauvais, nous avons classé tous les parents et leurs enfants selon leurs fitness croissantes. On ne gardera dans la population que les *N* premiers individus(8 dans notre expérience). La nouvelle population est présentée dans le tableau 2 à droite.

Tableau 2. *Meilleure population obtenue*

Enfant	fitness	Parents	fitness
1	4.2582	1	3.1490
2	11.5489	2	3.1803
3	7.2761	3	3.2824
4	9.1245	4	3.3785
	13.5487	5	3.5947
	5.0214	6	3.8692
7	3.1421	7	3.8761
8	3.8692	8	3.8815
9	3.51031		
10	7.6815		
11	7.2584		
12	3.1436		

Nouvelle population	fitness
1 : Enfant 7	3.1421
2 : Enfant 12	3.1436
3 : Parent 1	
4 : Parent 2	
	3.2824
6 : Parent 4	3.3785
7 : Enfant 9	3.5103
8 : Parent 5	3.5947

5.2.6. Critère d'arrêt

Il n'existe pas de critère d'arrêt garantissant la convergence de l'AG vers une solution optimale. Il est d'usage de fixer a priori, le nombre d'itérations. Quand ce nombre est atteint, on prend le meilleur chromosome obtenu comme solution optimale. Nous avons utilisé cette méthode dans nos expériences.

Pour l'algorithme génétique utilisé, nous avons fixé un nombre maximal de 100 itérations représenté par l'axe des abscisses (voir figure 9) pour représenter le test d'arrêt. Il faut noter qu'il n'existe pas de critère d'arrêt garantissant la convergence de l'AG vers une solution optimale. Il est d'usager de fixer a priori, le nombre d'itérations. Quand ce nombre est atteint, on prend le meilleur chromosone obtenu comme solution optimale. Les classes caractérisant les chiffres arabes au nombre de 10 (du 0 à 9) sont représentées par l'axe des colonnes au niveau de la figure 9.

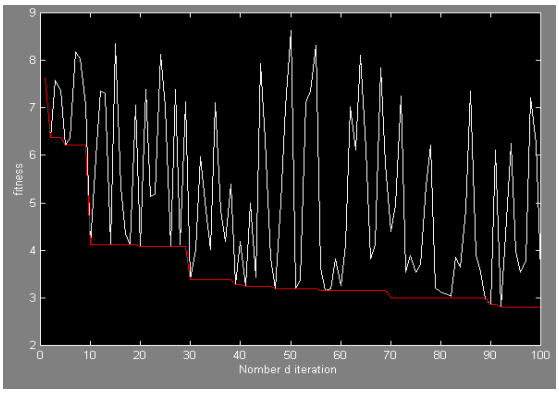

Figure 9. *Convergence du processus de classification pour la base des chiffres arabes (0-9)*

si sur un plan théorique, aucun résultat général ne prouve que cette méthode conduise à une solution optimale, en pratique la convergence globale est souvent constatée. C'est la remarque que nous l'avons constaté dans nos expériences. Nous remarquons une convergence du processus d'apprentissage obtenue lors de la 100ème itération avec une valeur d'adaptation minimale.

6. Méthode de fusion de données

Plusieurs résultats récents en reconnaissance automatique de la parole (obtenus sur différentes bases de données allant des petits lexiques aux très grands lexiques) ont montré que les systèmes hybrides MMC/RNA combinant la technologie des modèles de Markov cachés et des réseaux de neurones artificiels, conduisent généralement à des performances de reconnaissance équivalentes ou meilleures que celles des systèmes MMC utilisés dans les mêmes conditions, avec cependant plusieurs avantages supplémentaires au niveau des besoins en CPU et mémoire [BOI94], [GAU94], [MOR01], [MOR02]. En effet, des modèles hybrides MMC\RNA ont été conçus ces dernières années pour la parole : pour l'Anglais et pour le Français afin

d'additionner les qualités de chacun des modèles fusionnés mais sans réellement homogénéiser l'architecture.

Néanmoins l'un des principaux défauts liés à ces modèles hybrides réside dans le fait que le nombre de paramètres est en quelque sorte borné. En effet, aucune amélioration n'est généralement observée (comme habituellement pour les MMC continus) lorsque le nombre des données d'apprentissage et / ou de paramètres est fortement augmenté. Le tableau 3 reporte les résultats obtenus sur un corpus personnel avec deux réseaux de neurones de type PMC entraînés sur 1000 phrases pour le premier et environ 4000 pour le second.

Tableau 3. *Taux d'erreur au niveau du mot pour les trois dictionnaires de 60, 150, et 700 mots et deux modèles hybrides entraînés avec 1000 et 4000 phrases (paramètres log RASTA-PLP).*

Taille	*PMC (1000 phrases)*	*PMC (4000 phrases)*
60 mots	1.3%	1.5%
150 mots	5.0%	4.8%
700 mots	23.0%	24.2%

Les résultats reportés sur le tableau 3 montrent que l'augmentation du nombre des données d'apprentissage n'est pas réellement utiles pour améliorer le système hybride de base. Ceci est probablement du au faible nombre de paramètres associés au système indépendant du contexte. Ces paramètres sont bien estimés à partir des 1000 phrases et donc l'augmentation du nombre des données d'apprentissage n'a aucun effet bénéfique sur le système. Nous proposons dans ce papier une nouvelle méthode visant à explorer ce problème. Cette méthode est basée sur des expériences qui ont déjà montré qu'il est possible d'améliorer sensiblement les performances des systèmes en combinant plusieurs modèles.

6.1. Description de la procédure de fusion

Cette procédure vise à éclater les données d'apprentissage en plusieurs parties pour entraîner plusieurs réseaux et les recombiner lors de la phase de reconnaissance. Cet éclatement est réalisé par une simple classification des trames acoustiques (celles qui ont été incorrectement classées sont réutilisées pour entraîner un autre réseau). Cette procédure a été testée sur une base de données personnelle pour des vocabulaires de 60, 150 et 700 mots. N'ayant pas observé d'amélioration significative du taux de reconnaissance en utilisant un réseau PMC entraîné sur la totalité des données d'apprentissage (4000 phrases) par rapport à celui entraîné sur un plus petit ensemble d'apprentissage (voir tableau 3), nous avons alors cherché à tirer mieux parti de ces

données supplémentaires dont nous disposions, pour améliorer sensiblement les taux de reconnaissance.

En premier lieu, nous entraînons un réseau PMC classique estimant les probabilités a posteriori de mots sur une petite partie des données d'apprentissage (ce réseau sera par la suite dénommé PMC1). Ce réseau est alors utilisé pour filtrer le reste des données d'apprentissage pour un second réseau. Les données conservées pour entraîner le second réseau sont celles pour lesquelles le premier réseau PMC1 s'est trompé dans la classification. Ainsi pour toutes les données d'apprentissage, nous comparons pour chaque trame acoustique, les sorties du réseau PMC1 (correspondant aux probabilités a posteriori) et nous sélectionnons celles correspondant à la probabilité la plus élevée. Si la sortie sélectionnée est la bonne (celle correspondant à l'alignement Viterbi forcé), la donnée est écartée du nouvel ensemble d'apprentissage, sinon elle est gardée. Dans un esprit de simplification et pour éviter la suppression de trames acoustiques correspondant à un même mot sur une grosse partie de la base d'apprentissage, nous avons décidé, pour chaque mot de la base d'apprentissage, de calculer un pourcentage d'erreur de classification des trames acoustiques (nombre de trames mal classées / nombre de trames). Si ce taux d'erreur est supérieur à un seuil fixé, le mot est gardé pour l'apprentissage du deuxième réseau ; sinon, il n'est plus pris en compte. Il va de soi que plus le seuil fixé est élevé, moins il y aura de mots gardés et donc moins la subdivision des données d'apprentissage sera bonne. Le seuil a été fixé de manière à obtenir un nombre suffisant de trames acoustiques (environ le nombre de trames utilisées pour entraîner PMC1) et pour s'assurer de la validité de l'apprentissage des différents réseaux. Nous avons ainsi filtré une première fois les données d'apprentissage pour créer un nouvel ensemble d'apprentissage qui sera utilisé pour entraîner un second réseau (noté PMC2). Enfin, le reste des données d'apprentissage est passé au travers des deux réseaux PMC1 et PMC2. Si ces deux réseaux sont en désaccord sur la classification des exemples présentés à l'entrée, cet exemple est alors ajouté aux données d'apprentissage du troisième réseau. En revanche, si les deux réseaux sont d'accord, l'exemple est écarté. Là encore, un taux d'erreur par mot est calculé et comparé à un seuil pour décider ou non de l'insertion dans les données d'apprentissage du troisième réseau (noté PMC3). Le processus de division des données d'apprentissage a été stoppé après le troisième réseau, mais il faut noter qu'il est possible de continuer le processus décrit dans ce paragraphe jusqu'à ce que l'ensemble des données d'apprentissage ait été utilisé. L'apprentissage des réseaux est réalisé de manière classique par propagation arrière du gradient de l'erreur quadratique. Ces réseaux sont ensuite combinés par différentes méthodes pour estimer les probabilités utilisées par les MMC.

6.2. Les différentes méthodes de combinaison

En admettant que les trois réseaux aient été entraînés par la méthode décrite dans le paragraphe précédent, il faut pouvoir utiliser ces réseaux de manière efficace pour la reconnaissance. Chacun des trois réseaux PMC1, PMC2, PMC3 est composé de 10 sorties (correspondant aux 10 états stationnaires des MMC), 288 nœuds cachés et 2880 nœuds en entrée correspondant à 9 trames acoustiques de 26 paramètres, quantifiés par l'algorithme CMF [7] (voir [LAZa03], [LAZb03] pour plus de détails).

Le système utilisé lors de la reconnaissance est décrit sur la figure 10.

Ainsi, après extraction des paramètres acoustiques (utilisation de la méthode log RASTA-PLP pour les expériences décrites ci-après, voir [HER94], [LAZc03] pour plus de détails) et segmentation à l'aide de l'algorithme CMF et AG, un passage par chacun des trois réseaux PMC1, PMC2 et PMC3 est effectué, nous disposons donc pour chaque trame acoustique de 3 séries de probabilités qu'il nous faut combiner (voir figure 10). Il existe plusieurs techniques pour combiner les probabilités ; nous allons en décrire quelques-unes qui ont été testées sur la base de données personnelle.

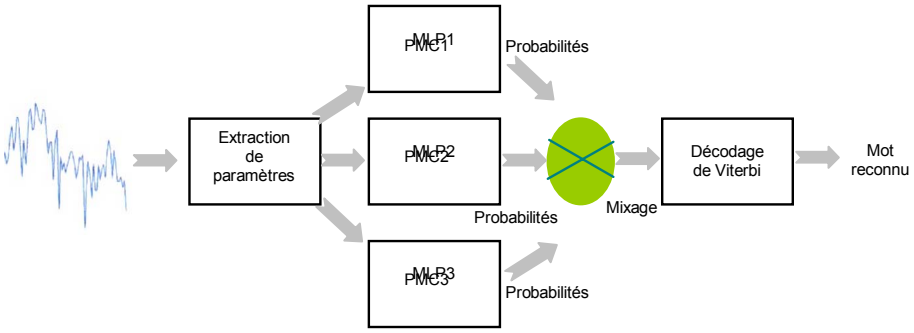

Figure10. *Le processus de reconnaissance pour la méthode de fusion.*

7 Les vecteurs ont été quantifiés en 4 dictionnaires selon le principe de l'algorithme CMF comme suit :
128 prototypes pour les coefficients log RASTA-PLP
128 prototypes pour les Δ log RASTA-PLP
32 prototypes pour la dérivée première de l'énergie ΔE
32 prototypes pour la dérivée seconde de l'énergie ΔΔE

6.2.1. Combinaison linéaire

Il s'agit de la combinaison la plus simple. Chacune des composantes des 3 vecteurs de probabilités pour chaque trame acoustique est « moyennée » selon la formule classique :

$$OUT[i] = \frac{1}{N}\sum_{j=1}^{N} Out_j[i] \tag{18}$$

où :

- N est le nombre d'experts (nombre de réseaux) utilisé (3 dans notre cas).
- $Out_j[i]$ est la composante i du vecteur observé à la sortie de l'expert j.

6.2.2. Combinaison linéaire dans le domaine logarithmique

C'est le même type de combinaison que celle décrite précédemment si ce n'est que nous avons utilisé $Out_j[i] = log[P_{PMCj}[q_i \backslash X]]$. Ce type de combinaison a déjà été utilisé avec succès dans des recombinaisons de systèmes à bandes multiples [DUP96], [MOR01], [MOR02] ou pour le calcul du score de la « poubelle »[8] en reconnaissance de mots clés par modèles hybrides [BOI94]. Le vecteur de sortie fourni aux modèles MMC est donc la moyenne des log probabilités de chacune des sorties des réseaux.

6.2.3. Combinaison par la technique du vote

Ce type de combinaison est un peu plus évolué. Pour chaque vecteur acoustique, le vecteur de sortie est construit de la manière suivante : Si les deux premiers réseaux PMC1 et PMC2 s'accordent sur la classification de la trame acoustique (même sortie ayant la meilleure probabilité), alors le vecteur de sortie du réseau PMC1 est utilisé comme vecteur de sortie fourni aux MMC. Si les deux réseaux sont en désaccord, alors le vecteur de sortie du réseau PMC3 est utilisé comme vecteur de probabilités fournit aux MMC. Cette méthode est appelée méthode du vote et est couramment utilisée pour la combinaison de plusieurs modèles.

6.2.4. Combinaison basée sur le critère entropique

Dans le type de combinaison précédente, le critère de sélection des vecteurs de sortie était basé sur l'accord ou le désaccord entre les deux premiers réseaux PMC1 et PMC2.

[8] La « poubelle » est un modèle qui prend en compte l'ensemble des mots prononcés par un locuteur qui n'appartiennent pas au lexique utilisé.

Ici, nous nous sommes basés sur un critère du type entropique. Pour chaque trame acoustique et pour chaque réseau, nous calculons l'entropie du réseau selon la formule suivante :

$$Entropie \quad = - \sum_{k=1}^{Nsorties} p(q_k \setminus X) \quad * \quad \log(p(q_k \setminus X)) \tag{19}$$

L'entropie est une mesure de la validité de l'information ou de l'incertitude d'une donnée. Le vecteur de sortie fourni au décodage est celui correspondant au réseau pour lequel l'entropie est la plus petite. En effet, dans le cas extrême où le réseau est absolument sûr (1 pour une sortie, 0 pour les autres), l'entropie est alors nulle.

Dans le cas où le réseau n'est pas capable de se décider (même probabilité *1 \ N sorties* pour chacune des sorties), l'entropie vaut alors – log(*1\N sorties*) > 0.

6.2.5. Combinaison par l'intermédiaire d'un PMC

Ce type de combinaison fournit généralement des résultats assez bons. Il suffit d'entraîner un réseau PMC classique avec en entrée, un vecteur composé des probabilités de sortie de chacun des trois PMC et un contexte acoustique quelconque. Ainsi dans les expériences menées sur la base de données personnelle, nous avons entraîné un réseau PMC avec 3*10 composantes en entrée, correspondant aux probabilités des 3 réseaux et un contexte de 3 trames acoustiques, soit 90 nœuds d'entrée, 288 nœuds cachés, 10 nœuds de sorties.

7. Expériences et résultats

7.1. Corpus utilisés

Trois bases de données ont été utilisées dans ce travail :

1. La première base (BD1) est lue par 30 locuteurs, chaque locuteur doit prononcer respectivement son nom & prénom, le nom des villes de naissance et résidence. Chaque son devrait être prononcé 10 fois. Nous avons choisit le vocabulaire d'une façon artificielle afin d'éviter les répétitions dans les noms des locuteurs et des villes, aussi bien pour les prénoms des locuteurs. Ce qui nous a permet d'avoir un vocabulaire de 1200 mots.

2. La deuxième base de données (BD2) est lue par les même locuteurs utilisés dans la première expérience. Ce vocabulaire contient 13 mots de commande (ex. sauvegarder\ sauvegarder sous\ sauvegarder tout\ précédent\ suivant, etc.) de sorte que chaque locuteur prononce chaque mot de commande 10 fois, ce qui donne un vocabulaire de 3900 sons.

3. La troisième base de données (BD3) est lue aussi par les même locuteurs utilisés dans la première expérience. Ce vocabulaire contient 10 chiffres arabes (*0-9*), de sorte que chaque locuteur prononce 10 fois chaque chiffre, ce qui donne un vocabulaire de 3000 sons.

Pour les données de test ont été énoncées par 8 locuteurs (4 hommes et 4 femmes) qui n'ont pas participé à l'apprentissage du système et qui prononcent la séquence "nom – prénom – ville de naissance – ville de résidence" 5 fois concernant le premier corpus, 5 fois les mots de commande sélectionnés au hasard (le nombre des mots de commande prononcés par chaque locuteur est entre 5 à 10 mots), et prononcent aussi 5 fois les chiffres arabes.

Pour des fins de reconnaissance automatique de mots isolés prononcés en Arabe, l'objectif de nos expériences est de faire en premier une étude comparative entre (1) le système MMC discrets (2) le système hybride MMC/PMC utilisant des distributions discrètes (application de l'algorithme c-moyennes pour la segmentation acoustique) (3) le système hybride MMC/PMC utilisant des distributions discrètes floues (application de l'algorithme c-moyennes floues pour la segmentation acoustique) (4) le système hybride MMC/PMC utilisant les AG pour la segmentation acoustique.

Les vecteurs acoustiques utilisés dans cette expérience sont les coefficients log RASTA-PLP. L'analyse est effectuée sur des fenêtres de 30 ms décalées de 10 ms. Ainsi, un vecteur de 13 composantes (énergie+ coefficients) est calculé toutes les 10 ms.

Dans toutes les expériences décrites dans ce travail, la même topologie des MMC a été employée pour les quatre types de modèles définis dans la suite de cette section.

7.2. Modèle 1- MMC discret

Les vecteurs acoustiques ont été quantifiés en 4 dictionnaires selon le principe de l'algorithme c-moyennes comme suit :

- 128 prototypes pour les coefficients log RASTA-PLP
- 128 prototypes pour les Δ log RASTA-PLP
- 32 prototypes pour la dérivée première de l'énergie ΔE
- 32 prototypes pour la dérivée seconde de l'énergie $\Delta\Delta E$

Des modèles de mots à 10 états ont été utilisés pour modéliser chacun des unités élémentaires (mots). Notant seulement que le choix de 10 états par modèle a été choisi d'une façon empirique.

7.3. Modèle 2 - Modèle hybride MMC/PMC avec des entrées fournies par l'algorithme C-Moyennes

Un PMC possédant en entrée 2880 neurones correspondant à 9 trames de contexte[9]. Le vecteur binaire fournit à l'entrée du réseau composé seulement de 36 bits à "1" (obtenu en appliquant les concepts de la quantification vectorielle "C-Moyennes" (voir modèle 1)). Une couche cachée de taille variable, une couche de sortie composée d'autant de neurones qu'il y a d'états MMC. Le nombre de neurones de la couche cachée a été choisit de manière à satisfaire la règle heuristique suivante [JOD94] :

$$\text{Nombre de neurones cachés} = (\text{nombre de neurones d'entrée} * \text{nombre de neurones de sortie})^{1/2}$$

Ainsi un PMC à une seule couche cachée comprenant 2880 neurones à l'entrée, 288 neurones pour la couche cachée et 10 neurones de sortie a été entraîné.

7.4. Modèle 3 - Modèle hybride MMC/PMC avec des entrées fournies par l'algorithme C-Moyennes Floues

Pour ce cas, nous avons essayé de comparer la performance du modèle hybride précédent avec celle d'un modèle hybride MMC/PMC utilisant en entrée du réseau un vecteur acoustique composé de valeurs réelles qui ont été obtenues en appliquant l'algorithme CMF. Nous avons présenté chaque paramètre cepstrale (log RASTA-PLP, Δ log RASTA-PLP, Δ E, $\Delta\Delta$ E) par un vecteur réel dont les composantes définies les degrés d'appartenance du paramètre aux différentes classes des "dictionnaires". La topologie du PMC est similaire au modèle 2, néanmoins que la couche d'entrée est composée d'un vecteur réel avec 2880 composantes réelles correspondant aux différents degrés d'appartenance des vecteurs acoustiques aux classes des "dictionnaires".

7.5. Modèle 4 - Modèle hybride MMC/PMC avec des entrées fournies par les AG

Pour ce dernier cas, les vecteurs acoustiques quantifiés qui sont présentés à l'entrée du PMC, sont fournis par l' AG. Les paramètres de l'AG utilisé sont récapitulés dans le tableau suivant (voir tableau 4):

[9] 9 trames de contexte correspondant à la configuration connue pour donner les meilleures résultats.n'est pas la solution à ce problème.

Tableau 4. *Paramètres de l' AG*

Nombre de générations maximal	100
Taille population	08
Probabilité de croisement	0.7
Probabilité de mutation	0.2
Pourcentage de remplacement de la Ω	0.5

Ces quatre types de modèles ont été comparés dans le cadre d'une reconnaissance de mots arabes isolés. Le tableau 5 résume les différents résultats obtenus des modèles utilisant seulement les paramètres acoustiques fournis par une analyse log RASRA-PLP. Tous les essais d'apprentissage et de test des quatre modèles ont été effectués sur la BD1 en premier puis sur la BD2 et enfin sur la BD3.

Tableau. 5 – *Taux de reconnaissance pour les quatre types de modèles. Coefficients log RASTA-PLP*

Corpus utilisé	*Modèle 1*	*Modèle 2*	*Modèle 3*	*Modèle 4*
BD 1	79.3%	81.9%	84%	82.7%
BD 2	78.7%	83.9%	84.7%	85.2%
BD 3	83.4%	84.2%	85.8%	85%

Les taux de reconnaissance obtenus ainsi que l'étude des erreurs commises lors du processus de reconnaissance montrent que :

1. L'approche du modèle hybride MMC/PMC discret (modèle 2, 3 et 4) est toujours plus performante que celle des MMC discrets.
2. Le modèle hybride MMC/PMC utilisant en entrée du PMC, un vecteur dont les composantes sont obtenues en appliquant l'algorithme CMF et de l'AG, a donné les meilleurs résultats pour les trois corpus utilisés.

La procédure de fusion décrite dans ce papier a été testée sur 3900 sons composant les données d'apprentissage (environ 437 000 de trames acoustiques). Pour la classification des trames acoustiques, nous avons utilisé uniquement l'algorithme CMF et les AG, vu les meilleurs résultats obtenus par rapport à ceux de l'algorithme c-moyennes. Trois réseaux ont été entraînés sur les différentes parties décrites dans le tableau 6.

Tableau 6. *Nombres de trames acoustiques utilisées pour chacun des réseaux*

PMC	*Trames (apprentissage)*	*Trames (validation croisée)*[10]
1	150 000	30 000
2	120 000	15 000
3	167 000	22 000
Réseau de base	437 000	67 000

A titre de comparaison, nous mettrons dans chacun des tableaux de résultats ceux correspondant au modèle hybride MMC/RNA de base dans les mêmes conditions. Les premiers tests effectués sur cette méthode ont été réalisés au niveau de la trame acoustique (tableau 7 et 8) en premier, puis au niveau du mot (tableau 9 et 10).

Tableau. 7 – *Taux de reconnaissance au niveau de la trame acoustique pour les différentes méthodes de combinaisons lors des phases d'apprentissage et de validation croisée : paramètres log RASTA-PLP et distributions discrètes floues.*

	Linéaire	*Log linéaire*	*Vote*	*Entropie*	*PMC*	*PMC de base*
APP	75.8 %	76.0 %	76.2 %	75.1 %	**77.2 %**	75.3 %
V-Crois	74.7 %	74.6 %	**75.9 %**	70.9 %	73.1 %	67.4 %

Tableau. 8 – *Taux de reconnaissance au niveau de la trame acoustique pour les différentes méthodes de combinaisons lors des phases d'apprentissage et de validation croisée : paramètres log RASTA-PLP et distributions obtenues avec les AG.*

	Linéaire	*Log linéaire*	*Vote*	*Entropie*	*PMC*	*PMC de base*
App	71.0 %	71.0 %	70.4 %	96.2 %	**71.5 %**	68.7 %
V-Crois	69.2 %	65.3 %	70.6 %	67.4 %	**72.3 %**	61.9 %

Les premiers résultats au niveau de la trame acoustique sont fortement encourageants puisque nous observons une réduction d'environ 21 % du taux d'erreur par rapport au système hybride MMC/PMC de base, cas des distributions discrètes floues (algorithme CMF), et une réduction de 19 % par rapport au système hybride MMC/PMC de base, cas des distributions obtenues par l'AG.

[10] La phase de « validation croisée » est utilisée pour adapter le taux d'apprentissage du PMC.

Observons maintenant les performances au niveau du mot pour des vocabulaires de 60, 150 et 700 mots sur les tableaux 9 et 10.

Tableau 9. *Taux d'erreur au niveau du mot pour les trois dictionnaires de 60, 150, et 700 mots et les différentes méthodes de combinaisons : paramètres log RASTA-PLP et distributions discrètes floues*

.

Taille	*Linéaire*	*Log linéaire*	*Vote*	*Entropie*	*PMC*	*PMC de base*
60 mots	**0.8 %**	1.1 %	1.4 %	1.5 %	0.9 %	2.1 %
150 mots	4.8 %	5.4 %	5.3 %	5.2 %	**4.3 %**	5.4 %
700 mots	16.7 %	16.8 %	17.5 %	18.5 %	**15.2 %**	18.7 %

Tableau. 10 – *Taux d'erreur au niveau du mot pour les trois dictionnaires de 60, 150, et 700 mots et les différentes méthodes de combinaisons : paramètres log RASTA-PLP et distributions obtenues par les AG*

Taille	*Linéaire*	*Log linéaire*	*Vote*	*Entropie*	*PMC*	*PMC de base*
60 mots	**1.1 %**	**1.2 %**	1.3 %	1.3 %	1.4 %	2.3 %
150 mots	**4.4 %**	4.6 %	4.6 %	**4.4 %**	4.5 %	5.6 %
700 mots	17.6 %	17.5 %	17.8 %	17.7 %	**16.7 %**	20.6 %

Nous observons, là encore, une réduction significative du taux d'erreur pour les trois types de vocabulaire (40 % pour 60 mots, 13 % pour 150 mots et 9% pour 700 mots) par rapport au système hybride MMC/PMC de base, cas des distributions discrètes floues (algorithme CMF), et aussi une réduction de 37 % pour 60 mots, 12 % pour 150 mots et 9.5% pour 700 mots par rapport au système hybride MMC/PMC de base, cas des distributions obtenues par l'AG.

8. Conclusion et perspective

Nous avons défini dans ce papier, une nouvelle méthode de fusion de données qui a été appliquée dans un système de reconnaissance de la parole arabe. Ce dernier est basé d'une part, sur une segmentation floue (application de l'algorithme c-moyennes floues) et d'une autre part, sur une segmentation à base des algorithmes génétiques.

Cette méthode permettant de diviser en plusieurs parties l'ensemble d'apprentissage et d'entraîner plusieurs PMC sur chacune de ces parties. Nous espérons ainsi tirer profit de l'apprentissage des réseaux sur des données filtrées par la procédure de fusion mettant en exergue des propriétés différentes du signal. Différents types de combinaisons des systèmes ont été testés :

-La combinaison linéaire.

-La combinaison linéaire dans le domaine logarithmique.

-La combinaison par la technique du vote.

-La combinaison par le critère entropique.

-La combinaison par un PMC.

Une réduction significative du taux d'erreur a pu être observée en utilisant la méthode de fusion décrite dans ce papier par rapport au système de base (40 % pour 60 mots, 13 % pour 150 mots et 9% pour 700 mots) pour des distributions discrètes floues, et 37 % pour 60 mots, 12 % pour 150 mots et 9.5% pour 700 mots, pour des distributions obtenues par l'AG. Cette procédure nous a permis de tirer au mieux parti des nombreuses données d'apprentissage dont nous disposions. Cette amélioration, obtenue dans le cadre d'une reconnaissance de mots isolés arabe, devrait aussi être constatée pour un système de reconnaissance de la parole continue. Dans cette optique, la même procédure décrite dans ce papier pourra être appliquée et le même système utilisé.

Il semble que la méthode de combinaison des sorties des PMC la plus efficace (du moins pour l'expérience décrite ici), dans les deux cas, consiste à combiner les sorties des trois réseaux de neurones par le biais d'un PMC classique.

Un seul petit inconvénient à la méthode : il est nécessaire de faire tourner en parallèle trois réseaux. Les temps de reconnaissance sont donc plus importants que pour le système de base. Ils restent cependant plus qu'acceptables. De plus il faut pouvoir disposer d'un nombre important de données d'apprentissage, le peu de données que nous possédons risque d'être un facteur limitatif de l'amélioration que nous pourrions observer. Cependant, nous avons détaillé dans ce papier, les résultats obtenus au niveau du mot pour des vocabulaires de 1200, 3000 et 3900 mots. C'est à notre connaissance la

première fois que les modèles hybrides sont utilisés pour effectuer une reconnaissance de la parole grand vocabulaire en arabe.

Nous suggérons comme même quelques axes de recherche en guise de prolongement à ce travail introductif.

- Il serait souhaitable de comparer les résultats du système hybride proposé avec ceux des systèmes utilisant les MMC avec apprentissage discriminant de type MMI ou MCE.

- Il faudrait aussi tester les méthodes proposées avec les bases de données standards pour l'anglais et le français.

- Il faut étendre ce système hybride pour reconnaître la parole continue.

9. Références

[BAL99] L. O. BALL, B. OZYURT, & J. C. Bezdek. "Clustering with a genetically optimized approach". *IEEE Transactions on Evolutionnary Computation*, 3(2) :103–112, 1999.

[BEZ81] J. C. BEZDEK. "Pattern Recognition with Fuzzy Objective Function Algorithms". *Plenum Press*, New York, 1981.

[BEZ99] J.C. BEZDEK. "Fuzzy models and algorithms for pattern recognition and image processing". *Handbooks of fuzzy sets series* ; FSHS 4. Boston: Kluwer Academic. xv, 776, 1999.

[BOI94] J-M. BOITE, H. BOURLARD, B. D'HOORE, S. ACCAINO, J. VANTIEGHEM. "Task independent and dependent training: performance comparison of HMM and hybrid HMM/MLP approaches". *Proceedings of the International Conference on Acoustics, Speech and Signal Processing,* volI, pp.617-620, 1994.

[BOI99] R. BOITE, H. BOURLARD, T. DUTOIT, J. HANCQ, H. LEICH. "Traitement de la parole". *Collection électricité, presses polytechniques et universitaires romandes*. Novembre 1999.

[BOU89] H. BOURLARD, C.J. WELLEKENS. "Links between Markov models and multilayer perceptrons". *In Advances in Neural Information Processing Systems*, ed. by D.J. Touretzky, 502-510, San Mateo. Morgan KaufmRNA, 1989.

[BOUa90] H. BOURLARD, N. MORGAN. "A continuous speech recognition system embedding MLP into HMM". *Advances in Neural Information Processing Systems*, vol. 2, D.S. Touretzky (ed.), pp. 502-510, Morgan Kaufmann, San Mateo, CA, 1990.

[BOUb90] H. BOURLARD, C.J. WELLEKENS. "Links between Markov models and multilayer perceptrons". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 1167-1178, 1990.

[BOU93] H. BOURLARD, N. MORGAN. "Continuous speech recognition by connectionist statistical methods". *IEEE Trans. on Neural Networks*, 1993.

[BOU94] H. BOURLARD, N. MORGAN. "Connectionist Speech Recognition – A Hybrid Approach*". Kluwer Academic Publishers*, 1994.

[DER97] O. DEROO, C. RIS, F. MALFRERE, H. LEICH , S. DUPONT, V. FONTAINE, J.M. BOÎTE. "Hybrid HMM/ANN system for speaker independent continuous speech recognition in French". *Faculté polytechnique de Mons – TCTS ,* Belgium*,* 1997.

[FUR86] S. FURUI. "Speaker independent isolated word recognizer using dynamic features of speech spectrum". *IEEE Trans. on Acoustics, Speech and Signal Processing*, 34.52-59, 1986.

[GAU94] J.L. GAUVAIN et *al*, "Speaker-independent continuous speech dictation". *Proc. Speech Communication*, Novembre 1994.

[GOL94] D. E. GOLDBERG. "Algoritmes génétiques - Exploration, optimisation et apprentissage automatique". *Addison Wesley*, 1994.

[HER90] H. HERMANSKY. "Perceptual Linear Predictive (PLP) Analysis for Speech". *J. Acoust. Soc. Am.*, pp. 1738-1752, 1990.

[HER94] H. HERMANSKY. "RASTA Processing of speech". *IEEE Trans. On Speech and Audio Processing,* vol.2, no.4, pp. 578-589, 1994.

[JOD94] J.F. JODOUIN. "Les réseaux de neurones: Principes & Définitions". *Edition Hermes*, Paris, France, 1994.

[LAZa02] L. LAZLI, M. SELLAMI. "Système Neuro-symbolique pour la Reconnaissance de la Parole Arabe". *CGE'02 : Conférence nationale sur le génie électrique, pp. 64 (résumé), EMP : "Ecole Militaire Polytechnique"*, 17-18 décembre, Alger, Algérie, 2002.

[LAZb02] L. LAZLI, M. SELLAMI. "Reconnaissance de la parole arabe par système hybride MMC/PMC". *CGE'02 : Conférence nationale sur le génie électrique, pp. 64 (résumé), EMP : "Ecole Militaire Polytechnique"*, 17-18 décembre, Alger, Algérie, 2002.

[LAZc02] L. LAZLI, M. SELLAMI. "Proposition d'une Architecture d'un Système Hybride MMC - PMC pour la Reconnaissance de la Parole Arabe". *MCSEAI 2002: the 7th Magrebian Conference on Computer Sciences,* vol I, pp. 101-109*,* 6-8 Mai, Annaba, Algérie, 2002.

[LAZa03] L. LAZLI. "Discriminant learning for hybrid HMM-ANN system using a fuzzy clustering for Arabic speech recognition". *JIEEE 2003: the 5th Jordanian International Electrical & Electronics Engineering conference*, pp?, 14-16 Octobre, Amman, Jordan, 2003.

[LAZb03] L. LAZLI, M. SELLAMI. "Speaker independent isolated speech recognition for Arabic language using hybrid HMM-MLP-FCM system". *AICCSA 2003 (ACS/IEEE): international conference on Applications & Computer Systems,* pp. 108 (Abstract), 14-18 Juillet, Tunis, Tunisie, 2003.

[LAZc03] L. LAZLI, M. SELLAMI. "Connectionist Probability Estimators in HMM Arabic Speech Recognition using Fuzzy Logic". *MLDM 2003: the 3rd international conference on Machine Learning & Data Mining in pattern recognition, LNAI 2734, Springer-verlag,* pp.379-388, 5-7 Juillet, Leipzig, Allemagne, 2003.

[LAZd03] L. LAZLI. "Système de reconnaissance Neuro-Markovienne pour la parole arabe isolée basé sur une segmentation floue". *Mémoire de MAGISTER en INFORMATIQUE*. Département d'Informatique, Université Badji Mokhtar, Annaba, Algérie, Juin

2003.

[LAZe03] L. LAZLI, M. SELLAMI. "Hybrid HMM-MLP system based on fuzzy logic for Arabic speech recognition". *PRIS 2003: the third international workshop on Pattern Recognition in Information Systems with ICEIS 2003: the 5th International Conference on Entreprise Information Systems, Springer-verlag,* pp. 150-155, 22-23 Avril, Angers, France, 2003.

[LAZa04] L. LAZLI, M.T. LASKRI. "Nouvelle méthode d'entraînement des systèmes hybrides HMM/ANN à base d'une segmentation floue. Application pour la reconnaissance automatique de la parole". *CARI 2004: the 7ème Colloque Africain sur la Recherche en Informatique,* pp. 331-338. Novembre 22-25, Hammamet, Tunisie, 2004.

[LAZb04] L. LAZLI, M.T. LASKRI. "Application de la méthode de fusion de données pour la reconnaissance automatique de la parole indépendante de locuteur". *MCSEAI 2004: the 8th Magrebian Conference on Computer Sciences,* pp. 523-533, 9-12 Mai, Sousse, Tunisie, 2004.

[MOR01] A. MORRIS et *al*, "MAP combination of multi-stream HMM or HMM/ANN experts". *In Eurospeech, Special Event Noise Robust Recognition*, Aalborg, Denmark, 2001.

[MOR02] A. MORRIS , "Same applications of priori knowledge in multi-stream HMM and HMM/ANN based ASR". Dalle Molle Institute for Perceptual artificial Intelligence (IDIAP), 2002

[OLS02] J. OLSSON, "Text Dependent Speaker Verification with a Hybrid HMM/ANN System". MASTER These Signal processing group, Uppsala University, Nov. 2002.

[OSO98] F.S. OSORIO. "INSS, un système hybride neuro-symbolique pour l'apprentissage automatique constructif". Thèse de Doctorat, institut polytechnique national de Grenoble - INPG, laboratoire LEIBNIZ - IMAG, Février, 1998.

[PHA99] D.L. PHAM, J.L. PRINCE. "An Adaptive Fuzzy C-means algorithm for Image Segmentation in the presence of Intensity Inhomogeneities". *Pattern Recognition Letters*, 20(1): p. 57-68, 1999.

[RAB89] L.R. RABINER. "A tutorial on hidden Markov models and selected applications in speech recognition". *Proceedings of the IEEE*, vol.77, no. 2, pp.257-285, 1989.

[RII97] S.K. RIIS, A. ROGH. "Hidden Neural Networks : A framework for HMM/ANN hybrids". *IEEE 1997, Proc. ICASSP-97,* Apr 21-24, Munich, Germanie, 1997.

[TOW94] G-G. TOWELL, J-W SHAVLIC. "Knowledge based artificial neural networks". *Journal of Artificial intelligence,* vol 70, pp.119-165, 1994.