Faculté Polytechnique de Mons

Laboratoire TCTS, Mons

Modèles dépendants du contexte et méthodes de fusion de données appliqués à la reconnaissance de la parole par modèles hybrides HMM/MLP

Dissertation originale présentée pour l'obtention du grade de Docteur en Sciences Appliquées par

Olivier Deroo

Le 22 décembre 1998

Membres du jury :

Professeur J. TRECAT, FPMs, Président

Professeur H. BOURLARD, EPFL, IDIAP
Professeur J-P. MARTENS, RUG

Dr. T. DUTOIT FPMs
Professeur C. CONTI, FPMs, Doyen

Professeur H. LEICH, FPMs, Promoteur

A Corinne et Solène ...

Résumé

Le problème de la reconnaissance automatique de la parole est un domaine d'études actif depuis le début des années 50. Les modèles les plus utilisés en reconnaissance de la parole sont les modèles de Markov cachés (HMMs). Dernièrement, une extension de ces modèles a été mise au point donnant naissance aux modèles hybrides. Les modèles hybrides combinent la technologie des modèles de Markov cachés et des réseaux de neurones artificiels. De nombreuses publications ont déjà montré l'efficacité de ces modèles en reconnaissance de la parole (continue ou isolée) indépendamment du locuteur pour de petits et grands vocabulaires. Ils sont couramment utilisés avec des modèles de phonèmes indépendants du contexte. Nous avons proposé dans ce travail d'étendre les modèles hybrides pour tenir compte d'un contexte acoustique plus important. Les modèles dépendants du contexte (diphones ou triphones), qui ont déjà prouvé leur efficacité par rapport aux modèles de phonèmes classiques pour les HMMs, sont plus adaptés parce que les effets de coarticulation sont explicitement modélisés. Leur utilisation avec les modèles hybrides est néanmoins beaucoup plus complexe. Nous avons développé dans cette thèse plusieurs nouvelles méthodes pour combiner efficacement les modèles de phonèmes dépendants du contexte et les modèles hybrides. Ces nouvelles méthodes, qui ont été testées et comparées à celles développées par d'autres laboratoires sur plusieurs tâches de reconnaissance, améliorent fortement les taux de reconnaissance.

Nous avons développé dans cette thèse un système de reconnaissance de la parole continue grand vocabulaire en français utilisant les modèles hybrides. Nous avons utilisé une méthode originale basée sur un alignement temporel du signal de parole avec un signal synthétique de bonne qualité pour générer la première segmentation utile pour initialiser le processus d'entraînement. Les résultats obtenus au niveau de la trame acoustique et du phonème sont très bons. Nous avons détaillé les résultats obtenus au niveau du mot pour des vocabulaires de 1000, 3000, 13000 et 65000 mots. C'est à notre connaissance la première fois que les modèles hybrides sont utilisés pour effectuer

une reconnaissance de la parole continue grand vocabulaire en français.

De nombreuses expériences ont déjà montré qu'une forte amélioration du taux de reconnaissance des systèmes HMMs traditionnels est observée lorsque plus de données d'entraînement sont utilisées. En revanche, l'augmentation du nombre de données d'entraînement pour les modèles hybrides s'accompagne d'une forte augmentation du temps nécessaire à l'entraînement des modèles mais pas ou peu des performances du système. Nous avons proposé une nouvelle méthode d'entraînement et de recombinaison des systèmes hybrides basée sur la fusion de données. Les données d'entraînement sont divisées de manière automatique en plusieurs sous ensembles sur lesquels sont entraînés des modèles hybrides. Lors de la reconnaissance, ces modèles sont combinés selon plusieurs critères pour fournir la phrase reconnue. Nous avons montré que cette méthode de fusion s'accompagne d'une amélioration significative du taux de reconnaissance.

Enfin, nous avons proposé dans ce travail quelques méthodes pour effectuer une réestimation des scores des N meilleures séquences de mots. Nous avons également montré qu'il est possible d'utiliser des modèles acoustiques dépendants du contexte ou des modèles de langage plus performants pour réordonner ces N meilleures séquences de mots de manière à améliorer les performances du système. Nous avons montré qu'il est possible de supprimer automatiquement 1 trame sur 2 et même 2 trames sur 3 sans observer de détérioration des taux de reconnaissance pour les modèles hybrides. Ce type d'opération permet d'accélérer fortement le processus d'estimation des probabilités a posteriori (donc le processus de reconnaissance). Nous répondons ainsi à une des principales critiques de ce type de système : le temps de calcul.

Abstract

The Automatic Speech Recognition (ASR) problem is an active field of study since the beginning of the 50's. Great progress has been made, especially since the 70's, using statistical modeling approaches with Hidden Markov Models (HMMs). Lately an extension of these models, using connectionist models, gave rise to very attractive systems named hybrid models. Hybrid models combine the technology of Hidden Markov Models and Artificial Neural Networks. Many publications already showed the effectiveness of these models in speaker independent (continuous or isolated) speech recognition systems. They are usually used with context independent phonemes. We proposed in this work to extend hybrid models by taking into account a more significant acoustic context. Context dependent models, which already proved their effectiveness with traditional HMMs, are more adapted because the coarticulation effects are explicitly modeled. Their use with hybrid models, however, is much more complex. We developed in this thesis several new methods to combine context-dependent models and hybrid models in an efficient way. The new methods, which were tested and compared with those developed by other laboratories on several different tasks, strongly improve the recognition rate.

We have developed a large vocabulary speaker-independent continuous speech recognition system in French, based on hybrid models. We use an original method based on the temporal alignment of the speech signal on a high quality synthetic speech pattern to phonetically label the database and generate a first segmentation before starting the training process. The results obtained at the frame and phoneme level using this system are good. The results obtained at the word level for a 1K, 3K, 13K and 65 K dictionary are detailed too. It is to our knowledge, the first time that hybrid models are used in a hybrid HMM/MLP continuous speech recognition system in French.

It is well known that traditional HMM systems lead to a considerable improvement when more training data or more parameters are used. However, using more data with hybrid HMM/MLP 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 on each set. During the recognition phase, models are combined using several criteria (based on data fusion techniques) to provide the recognized sentence. We showed in this thesis that this method significantly improved the recognition accuracy.

Lastly, we proposed in this work some methods to carry out some N-Best rescoring. We showed that it is possible to use context dependent acoustic models or more powerful language models to reorder these N best sentences. We showed in addition that it is possible to skip automatically 1 frame out of 2 and even 2 out of 3 without degrading the recognition accuracy. This method strongly accelerates the recognition process. We thus answer one of the most important criticism about hybrid systems : the complexity of the system.

Remerciements

Mes premiers remerciements vont à M. Henry Leich, chef de service du laboratoire TCTS, qui m'a acceuilli, accordé sa confiance et permis d'accomplir ce travail dans les meilleures conditions.

Je remercie les membres du jury, Messieurs J. Trecat, J.P. Martens, T. Dutoit, C. Conti pour avoir accepté d'évaluer le présent travail et particulièrement M. Hervé Bourlard pour les nombreux conseils précieux et son aide pendant les deux années où j'ai eu l'occasion de le cotoyer.

J'exprime ma plus grande gratitude aux personnes travaillant ou ayant travaillé pour le projet STRUT (Speech Training and Recognition Unified Tool). L'ensemble des expériences reportées dans ce travail on été réalisées avec cet outil. Je tiens à souligner l'extrème disponibilité et serviabilité qu'on démontraient à mon égard Messieurs J.M. Boite, L. Couvreur, S. Dupont, G. Durou, V. Fontaine, C. Ris, L. Zanoni, ... pour la mise en oeuvre de ce logiciel d'entraînement et de reconnaissance de la parole et ma fierté d'y avoir contribué.

Je tiens à remercier les groupes "synthèse de la parole", "codage de la parole" et "reconnaissance de caractères manuscrits" (sans oublier l'ingénieur système F. Bataille) du laboratoire pour leurs conseils et leur aide. Je mesure pleinement la chance que j'ai eue de travailler dans une aussi bonne ambiance et je remercie donc l'ensemble des personnes que j'ai cotoyées depuis mon arrivée.

Enfin je remercie ma famille et mes amis pour leur soutien tout au long de ce travail.

Table des matières

Table des figures

Liste des tableaux

Abréviations

Voici les principales abréviations utilisées dans ce travail.

- ANN : Artificial Neural Network, réseau de neurones artificiel.
- CD : Context Dependent, phonème dépendant du contexte.
- CI : Context Independent, phonème indépendant du contexte.
- CMS : Cepstral Mean Substraction, soustraction cepstrale.
- FFT : Fast Fourier Transform, Transformée de Fourier rapide.
- HMM : Hidden Markov Model, modèle de Markov caché.
- MLE : Maximum Likelihood Estimation, critère du Maximum de vraisemblance
- MAP : Maximum A Posteriori, critère du maximum a posteriori
- MFCC : Mel-scale Frequency Cepstral Coefficient, coefficients cepstraux calqués sur l'échelle de Mel (correspondant mieux à la perception humaine).
- MLP : Multilayer Perceptron, réseau de neurone perceptron à couches multiples.
- PLP : Perceptual Linear Predictive.
- RTE : Réduction du Taux d'erreur.
- RNN : Recurrent Neural Network, réseau de neurone récurrent.
- STRUT : Speech Training and Recognition Unified Tool
- T : Ensemble des données d'entraînement.
- TE : Taux d'erreur.

Notations

Voici les principales notations utilisées dans ce travail.

- $X = X_1, ..., X_N$: Séquence de vecteurs acoustiques de longueur N.
- $X_1^n = X_1, ..., X_n$: partie de X correspondant aux instants 1 à n.
- X_n : vecteur acoustique au temps n.
- q^n représente l'état du HMM au temps n.
- q_k^n représente l'état q_k d'un HMM visité au temps n.
- q_I et q_F état initial et final d'un modèle HMM.
- $Q_1^N = \{q^1, ..., q^n, ..., q^N\}$ séquence ordonnée de N états.
- $p(x_n|q_k)$: densité de probabilité d'observation du vecteur x_n dans l'état q_k.
- $p(q_k|x_n)$: probabilité a posteriori de l'état q_k conditionné par x_n.
- $p(q_k, c_j|X)$: probabilité d'être dans l'état q_k et le contexte c_j étant donné la séquence acoustique X.
- $P(.)$: probabilité ou densité de probabilité globale (relative à une séquence ou sous séquence d'états ou de vecteurs acoustiques).
- ω : fréquence angulaire.
- Ω : fréquence de Bark.

Chapitre 1

Introduction

1.1 Introduction

Le problème de la reconnaissance automatique de la parole consiste à extraire à l'aide d'un ordinateur l'information contenue dans le signal de parole. La technologie la plus utilisée depuis ces 20 dernières années est basée sur des modèles statistiques (modèles de Markov cachés HMM) capables de modéliser simultanément les caractéristiques fréquentielles et temporelles du signal étudié.

La reconnaissance automatique de la parole est un domaine d'étude actif depuis le début des années 50. Il est clair qu'un outil de reconnaissance de la parole efficace faciliterait l'interaction entre les hommes et les machines. Les applications possibles associées à un tel outil sont nombreuses et sont amenées à connaître un essor fabuleux dans les prochaines années. La plupart des applications en reconnaissance de la parole peuvent être regroupées en 4 catégories :

1. Commande et contrôle :
   Contrôler à l'aide de la parole des équipements particuliers (machines, robots ...) ou des programmes (ouvrir par exemple des fenêtres ou naviguer sous Windows, aide aux personnes handicapées).

2. Accès à des bases de données ou recherche d'informations :
   Home banking, composition automatique de numéros de téléphone, serveurs vocaux, réservation d'un vol, guidage automatique (dans une voiture), remplir un questionnaire, effectuer un audit qualité ...

3. Dictée vocale :
   Création de lettres, rapports et autres documents par l'intermédiaire de la parole.

4. Transcription automatique de la parole :
   Indexation de programmes télévision ou radio, sous-titrage et traduction automatique ...

De nombreux progrès ont été réalisés ces dix dernières années dans ce domaine. Il existe d'ailleurs des logiciels vendus actuellement dans le commerce se vantant d'effectuer une reconnaissance de la parole continue pour un vocabulaire important. Néanmoins, les performances de ces systèmes sont encore largement inférieures à celles des êtres humains. Même si les progrès réalisés en moins de 50 ans sont énormes, il reste de nombreux problèmes à résoudre.

Les capacités d'un système de reconnaissance de la parole et la difficulté de son développement peuvent être mesurées en se posant quatre questions sur le système correspondant à quatre classes de variabilité du signal de parole :

- Qui peut utiliser le système ? N'importe quel locuteur (indépendamment du sexe, de l'âge...) peut-il parler à la machine avec des chances d'être compris ?
- Comment doit-on parler au système ? Quel est le mode d'élocution permis ?
- Que reconnaît le système ? Quel est le vocabulaire permis ?
- Quel est le taux de reconnaissance ? Autrement dit, quel est le pourcentage de mots reconnus ?

Les spécialistes de la reconnaissance de la parole répondent à ces questions en utilisant un vocabulaire particulier qu'il est utile de rappeler.

1.2 Le mode de fonctionnement

Un système de reconnaissance peut être utilisé sous plusieurs modes :

- Dépendant du locuteur (monolocuteur) :
  Dans ce cas particulier, le système de reconnaissance a été configuré pour un locuteur spécifique [1].
- Pluri-locuteur (ou multi-locuteur) :
  Le système de reconnaissance a été élaboré pour un groupe restreint de personnes. Le passage d'un locuteur à un autre du même groupe se fait sans adaptation.
- Indépendant du locuteur :
  Tout locuteur peut utiliser le système de reconnaissance.

Il est évident que plus le nombre de locuteurs est élevé, plus l'adaptation du système de reconnaissance est difficile. Nous nous placerons pour tous les tests réalisés dans ces travaux dans la configuration système "indépendant du locuteur".

1. C'est le cas de la plupart des logiciels de reconnaissance de la parole vendus sur le marché actuellement. Les principaux systèmes de dictée vocale actuels possèdent une phase d'entraînement recommandée avant toute utilisation (voire même une adaptation continue des paramètres au cours de l'utilisation du logiciel) afin d'effectuer une adaptation des paramètres à la voix de l'utilisateur.

1.3 Elocution

Le mode d'élocution caractérise la façon dont on peut parler au système. Il existe quatre modes d'élocution distincts.

- Mots isolés :
  Chaque mot doit être prononcé isolément, c'est à dire précédé et suivi d'une pause.

- Mots connectés :
  Le système reconnaît des séquences de quelques mots sans pause volontaire pour les séparer (exemple : reconnaissance de chiffres connectés ou de nombres quelconques ...).

- Parole continue lue :
  C'est le discours usuel, si ce n'est que les textes sont lus.

- Parole continue spontanée :
  C'est le discours usuel, sans aucune contrainte.

La reconnaissance de mots isolés fonctionne relativement bien de nos jours. De très bons résultats ont été publiés par de nombreux laboratoires. Généralement, de tels outils de reconnaissance de la parole sont utilisés pour un vocabulaire de commande correspondant à des actions simples (gestion de menus ...). Le mode "mots connectés" est utile dans des cas particuliers comme la reconnaissance de nombres quelconques. Les laboratoires de recherche ont publié de bons résultats pour un vocabulaire restreint. Nous verrons d'ailleurs que le système développé à la Faculté Polytechnique de Mons est capable d'obtenir des taux de reconnaissance de l'ordre de 98% dans le cadre d'une reconnaissance de chiffres connectés indépendante du locuteur. Le mode "parole continue" est naturel et beaucoup plus compliqué. Quelques laboratoires (dont TCTS [31]) ont publié des premiers résultats impressionnants pour ce mode d'élocution avec un vocabulaire d'environ 65000 mots, mais il n'existe pas encore de systèmes avec des performances comparables à celles des humains. La parole spontanée est beaucoup plus difficile à traiter, les dernières études rapportent des taux d'erreur de l'ordre de 50%.

Les trois premiers modes d'élocution seront abordés lors de cette étude. Les expériences décrites dans ce travail ont été effectuées sur de la parole claire (absence de bruit, de réverbération ...).

1.4 Vocabulaire et syntaxe

Le vocabulaire est l'ensemble des mots que le système est capable de reconnaître. La taille du vocabulaire peut varier de quelques mots à plusieurs dizaines de milliers de mots. Il est évident que plus le vocabulaire est grand, plus le risque de se tromper augmente et donc moins le système est performant.

La syntaxe spécifie plutôt les contraintes imposées sur les suites de mots prononcés. Elle peut être inexistante (tout mot peut suivre n'importe quel autre mot), ou bien contraignante (après un mot donné seuls certains autres sont autorisés). L'utilisation d'une syntaxe facilite la tâche du système de reconnaissance en limitant le nombre de mots et d'hypothèses à traiter. Les nombres constituent un exemple de syntaxe naturelle simple (après les mots "trente et", il faut obligatoirement le mot "un" du fait de la grammaire française).

Des modèles de langages du type N-Gramme (tout mot peut être suivi par une séquence de N-1 autres avec une probabilité fixée) sont aussi utilisés. En général des modèles de grammaire du type bigrammes (N=2) et trigrammes (N=3) sont communément utilisés en reconnaissance de la parole et le seront dans ce travail notamment pour notre système de reconnaissance de la parole continue en français (voir paragraphe 6.4).

1.5 Taux de performance

Le taux de performance ou taux de reconnaissance permet de mesurer l'efficacité du système de reconnaissance testé. Ce taux varie fortement selon le type de canal de transmission utilisé (microphone, téléphone), la taille du vocabulaire, et le type d'élocution. Il est donc influencé par la plupart des paramètres décrits dans ce chapitre.

Il existe différentes valeurs mesurant les performances d'un système de reconnaissance :

- Le taux de reconnaissance : pourcentage de mots ou de phrases reconnus correctement.
- Le taux d'erreur : 1 - taux de reconnaissance (noté T. E. dans le reste du document).
- Le taux de substitution : pourcentage de mots pour lesquels le système a commis une erreur (noté Subs. dans le reste du document).
- Le taux de rejet : pourcentage de mots que le système n'a pas compris.

- Le taux d'omission : pourcentage de mots non détectés (noté Sup. dans le reste du document).
- Le taux d'insertion : pourcentage de mots reconnus alors qu'aucun mot n'a été prononcé (noté Ins. dans le reste du document).
- Réduction du taux d'erreur $= \frac{ancienT.E.-nouveauT.E.}{ancienT.E.}$ (noté R. T. E. dans le reste du document). Ce taux permet de mesurer l'influence d'une méthode testée sur le système de reconnaissance.

En résumé, nous nous placerons dans un cadre assez général pour cette étude puisque nous tenterons d'améliorer les systèmes de reconnaissance de la parole continue, connectée et isolée, pour différents types de vocabulaire (grand, moyen, petit vocabulaire) et pour différents langages (l'anglais et le français).

1.6 Objet de la thèse

Les modèles les plus utilisés en reconnaissance de la parole sont les modèles de Markov cachés (HMMs). Dernièrement, une extension de ces modèles a été mise au point donnant naissance aux modèles hybrides. Les modèles hybrides combinent la technologie des modèles de Markov cachés et des réseaux de neurones artificiels. De nombreuses publications ont déjà montré l'efficacité de ces modèles en reconnaissance de la parole (continue ou isolée) indépendamment du locuteur pour de petits et grands vocabulaires. Ils sont généralement utilisés avec des modèles de phonèmes indépendants du contexte. Nous avons proposé dans ce travail d'étendre ces modèles hybrides pour tenir compte d'un contexte acoustique plus important. Les modèles dépendants du contexte, qui ont déjà prouvé leur efficacité par rapport aux modèles de phonèmes classiques pour les HMMs, sont plus adaptés parce que les effets de coarticulation sont explicitement modélisés. Leur utilisation avec les modèles hybrides est beaucoup plus complexe. Nous avons développé plusieurs nouvelles méthodes pour combiner efficacement les modèles de phonèmes dépendants du contexte et les modèles hybrides. Cette nouvelle modélisation améliore fortement les taux de reconnaissance.

Nous avons développé un système de reconnaissance de la parole continue grand vocabulaire en français utilisant les modèles hybrides. Ce système a été entraîné sur le corpus BREF-80. Cette base de données n'étant pas segmentée, nous avons utilisé une méthode originale basée sur un alignement temporel du signal de parole avec un signal synthétique de bonne qualité pour générer la première segmentation utile pour initialiser le processus d'entraînement. Les résultats obtenus sur ce corpus au niveau de la trame acoustique et du phonème sont très bons. Nous développerons aussi les résultats obtenus au niveau du mot pour des vocabulaires de 1000, 3000, 13000 et 65000 mots. C'est à notre connaissance la première fois que les modèles hybrides sont utilisés pour effectuer une reconnaissance de la parole continue grand vocabulaire en français.

Nous avons proposé une nouvelle méthode d'entraînement et de recombinaison des systèmes hybrides basée sur la fusion de données. Les données d'entraînement sont divisées de manière automatique en plusieurs sous ensembles sur lesquels sont entraînés des modèles hybrides. Lors de la reconnaissance, ces modèles sont combinés par l'intermédiaire de plusieurs critères pour fournir la phrase reconnue. Nous avons montré que cette méthode de fusion s'accompagne d'une amélioration significative du taux de reconnaissance.

Nous avons aussi proposé dans ce travail quelques méthodes pour effectuer une réestimation des scores des N meilleures hypothèses. Nous avons montré qu'il est possible d'utiliser des modèles acoustiques dépendants du contexte ou des modèles de langage plus performants pour réordonner ces N meilleures séquences de mots. Nous avons montré que l'utilisation cumulée des méthodes de génération des N meilleures séquences de mots et de réévaluation des scores respectifs de chaque hypothèse s'accompagne aussi d'une amélioration des taux de reconnaissance.

1.7 Structure de la thèse

Dans le second chapitre, nous décrirons l'architecture générale d'un système de reconnaissance de la parole et les différents éléments qui le composent. Après avoir mis en évidence les principales caractéristiques du signal de parole et les difficultés rencontrées lors de sa modélisation, nous résumerons l'analyse acoustique préliminaire effectuée sur la parole avant le processus de reconnaissance. Nous décrirons ensuite les modèles les plus utilisés en reconnaissance de la parole (les modèles de Markov cachés, HMMs).

Le troisième chapitre sera consacré à une extension des modèles HMMs (les modèles hybrides). Nous montrerons que les modèles hybrides combinant la technologie des modèles de Markov cachés et des réseaux de neurones permettent de résoudre certains problèmes liés à l'utilisation des HMMs. Nous montrerons que les taux de reconnaissance obtenus avec ces modèles hybrides sont au moins aussi performants et même supérieurs aux modèles HMMs classiques décrits dans le chapitre 2 sur deux corpus différents (BDSONS et RESOURCE MANAGEMENT). De nombreuses publications ont montré l'avantage de combiner les réseaux de neurones et les modèles de Markov cachés en reconnaissance de la parole (isolée ou continue) indépendamment du locuteur pour de petits ou de grands vocabulaires. Ils seront à la base de la plupart des expériences menées dans ce travail.

Il est possible, en utilisant les modèles décrits dans les chapitres 2 et 3, de trouver la séquence de mots qui correspond le mieux à ce qui a été effectivement prononcé. Nous décrirons dans le chapitre 4 les méthodes permettant de générer la meilleure solution et plusieurs alternatives correspondant à ce qui a été énoncé. Il existe plusieurs méthodes permettant de générer les N meilleures séquences de mots au lieu de la meilleure uniquement. Nous décrirons les avantages et inconvénients de chacune d'entre elles et nous expliquerons les raisons pour lesquelles nous avons sélectionné une méthode en deux passes (une pour l'estimation des probabilités et l'autre utilisant un algorithme de recherche dans un graphe du type A*).

Nous montrerons dans le chapitre 5 que la génération de ces N meilleures solutions permet d'effectuer un deuxième traitement sur cette liste de propositions impliquant un meilleur taux de reconnaissance. Nous décrirons dans ce chapitre plusieurs tentatives de réestimation des scores. Nous démontrerons qu'il est possible d'utiliser des modèles acoustiques dépendants du contexte ou des modèles de langage plus performants pour effectuer un recalcul efficace des scores de ces N meilleures solutions. Nous montrerons que l'utilisation cumulée des méthodes de génération des N meilleures séquences de mots et de réévaluation des scores s'accompagne d'une amélioration significative du taux de reconnaissance sur des corpus de parole continue grand vocabulaire (RESOURCE MANAGEMENT et BREF-80). Enfin, nous développerons un nouveau type de modèle utilisant une information temporelle supplémentaire et nous prouverons qu'il est possible d'obtenir de très bons taux de reconnaissance pour peu que l'on utilise une bonne valeur de l'index temporel. Nous proposerons d'utiliser ce modèle pour effectuer un nouveau type de réestimation des scores des N meilleures séquences de mots.

Dans le chapitre 6, nous décrirons le décodeur grand vocabulaire Noway et la méthode utilisée pour l'élaboration d'un système de reconnaissance de la parole continue en français. Nous montrerons que ce système est capable d'effectuer une reconnaissance de phonèmes avec un taux d'erreur de 25%

(ce qui correspond à notre connaissance à l'un des meilleurs taux publiés sur le corpus utilisé). Nous donnerons aussi les résultats obtenus au niveau du mot pour des vocabulaires de 1000, 3000, 13000 et 65000 mots. Les résultats obtenus avec le modèle hybride sont à notre connaissance les premiers utilisant ce type de modèle pour le français.

Dans le chapitre 7, nous montrerons qu'il est possible d'accélérer le temps de calcul associé aux réseaux de neurones en supprimant de manière automatique certaines trames acoustiques présentées à l'entrée du modèle et ce, sans dégrader les taux de reconnaissance. Cette technique n'avait encore jamais été appliquée aux modèles hybrides. Nous montrerons qu'il est possible de supprimer 1 trame sur 2 et même 2 trames sur 3 sans dégrader les taux de reconnaissance. Ceci permet d'obtenir un gain de temps important lors du processus de reconnaissance.

Nous décrirons dans le chapitre 8 de nouveaux types de modèles dépendants du contexte, et nous montrerons qu'il est possible d'améliorer sensiblement les taux de reconnaissance en tenant compte d'un certain contexte acoustique. Ces modèles dépendants du contexte ont déjà prouvé leur efficacité par rapport aux modèles indépendants du contexte pour les HMMs; mais il est assez difficile de les utiliser avec des modèles hybrides (HMM+MLP). Nous développerons plusieurs nouvelles méthodes pour combiner les modèles dépendants du contexte et les modèles hybrides. Nous montrerons que cette nouvelle modélisation améliore fortement les taux de reconnaissance sur des corpus de parole isolée (PHONEBOOK) ou continue (RESOURCE MANAGEMENT).

Finalement, nous développerons dans le chapitre 9 une méthode d'entraînement et de recombinaison de systèmes hybrides basée sur la fusion de données. Les données d'entraînement sont divisées de manière automatique en plusieurs sous-ensembles sur lesquels sont entraînés des modèles hybrides. Lors de la reconnaissance, plusieurs critères de recombinaison ont été testés. Pour la plupart des critères testés, nous avons observé une amélioration sensible des taux de reconnaissance par rapport au système de base. Cette nouvelle méthode d'entraînement et de combinaison de plusieurs réseaux est facilement utilisable et particulièrement efficace lorsqu'on possède des corpus d'entraînement importants (lors d'une évaluation des systèmes de reconnaissance par exemple).

Nous résumerons en conclusion les différentes méthodes testées dans ce travail, les améliorations qu'elles ont engendrées et leurs possibles extensions.

Chapitre 2

Reconnaissance de la parole

2.1 Introduction

Le signal de parole appartient à la classe des signaux acoustiques produits par des vibrations de couches d'air. Les fluctuations de la pression de l'air produisent des variations de ce signal, en fonction du temps, qui peuvent être enregistrées de façon analogique ou digitale. Ceci constitue une représentation élémentaire du signal de parole [14]. Ce signal est le résultat du passage du flux laryngé (l'air qui est passé par nos poumons à travers nos cordes vocales) au travers du conduit vocal. La figure 2.1 présente les différentes étapes dans les processus d'entraînement et de reconnaissance d'un système. La ligne pointillée marque la séparation entre les deux processus. Lors de l'entraînement, il est nécessaire d'entraîner à la fois les modèles de grammaire (sur des textes) et des modèles statistiques particuliers, les modèles de Markov cachés HMMs [7, 9, 10, 16, 57, 85, 87] (sur des données acoustiques). Outre les données acoustiques, l'entraînement de ces modèles nécessite une définition précise des unités lexicales de base utilisées et un dictionnaire décrivant la liste des mots qui pourront être reconnus. Lors de la reconnaissance après extraction des paramètres acoustiques, un décodage est effectué et le système de reconnaissance fournit en sortie la phrase la plus probable étant donné la grammaire imposée et les modèles HMMs.

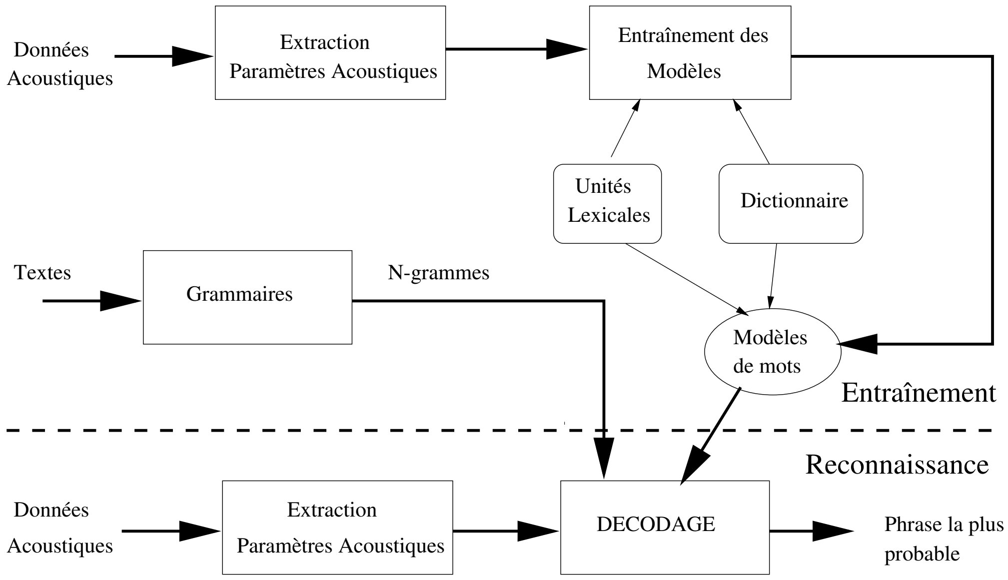

FIG. 2.1 - *Schéma général d'un système de reconnaissance de la parole : entraînement et reconnaissance.*

Chacun des éléments présents sur cette figure sera décrit dans ce chapitre.

2.2 Le signal de parole

Le signal de parole n'est pas un signal ordinaire; il est le vecteur d'un phénomène extrêmement complexe : la communication parlée. La reconnaissance de la parole pose de nombreux problèmes aux chercheurs depuis 1950. D'un point de vue mathématique, il est difficile de modéliser le signal de parole, car ses propriétés statistiques varient au cours du temps. Nous allons ici tenter de mettre en évidence quelques caractéristiques notoires de ce signal non-stationnaire afin de faire ressortir les problèmes posés lors de son traitement.

2.2.1 Redondance du signal

Le signal de parole est extrêmement redondant. Cette grande redondance lui confère une robustesse à certains types de bruits. De nombreuses recherches sont menées afin de rendre les systèmes de reconnaissance robustes aux bruits [34], mais les performances humaines sont encore loin d'être atteintes. Rappelons que les expériences décrites dans ce travail ont été réalisées sur des corpus de parole claire (sans bruits).

2.2.2 Variabilité du signal

Le signal de parole possède une très grande variabilité. Une même personne ne prononce jamais un mot deux fois de façon identique. La vitesse d'élocution peut varier, la durée du signal est alors modifiée. Toute altération de l'appareil phonatoire peut modifier la qualité de l'émission (exemple : rhume, fatigue ...). De plus, la diction évolue dans le temps. La voix est modifiée au cours des étapes de la vie d'un être humain (enfance, adolescence, âge adulte ...). La variabilité interlocuteur est encore plus évidente. La hauteur de la voix, l'intonation, l'accent diffèrent selon le sexe, l'origine sociale, régionale ou nationale. Un exemple flagrant de cette variabilité apparaît lorsque vous comparez la voix d'un locuteur originaire du nord avec celle d'un locuteur originaire du sud de la France. Enfin, la parole est un moyen de communication où de nombreux éléments entrent en jeu, tels le lieu, l'émotion du locuteur, la relation qui s'établit entre les locuteurs (stressante ou amicale). Ces facteurs influencent la forme et le contenu du message. L'acoustique du lieu (milieu protégé ou environnement bruité), la qualité du microphone ou de la ligne téléphonique, les bruits de bouche, les hésitations, les mots hors vocabulaire sont autant d'interférences supplémentaires sur le signal de parole que le système doit compenser.

2.2.3 Les effets de coarticulation

La production parfaite d'un son suppose un positionnement précis des organes phonatoires. Le déplacement de ces organes est limité par une certaine inertie mécanique. Les sons émis subissent alors l'influence de ceux qui les précèdent ou les suivent. Ces effets de coarticulation sont des interférences sur le signal de parole. Ils entraînent l'altération des formes sonores en fonction du contexte. L'effet de coarticulation est un facteur de variabilité supplémentaire important du signal de parole. Nous verrons d'ailleurs dans le chapitre 8 que la prise en compte de ce facteur améliore sensiblement les performances du système de reconnaissance.

2.3 Extraction des paramètres

Pour résoudre les problèmes liés à la complexité de la parole, il est possible de calculer des coefficients représentatifs du signal traité. Ces coefficients sont calculés à intervalles temporels réguliers. En simplifiant les choses, le signal de parole est transformé en une série de vecteurs de coefficients.

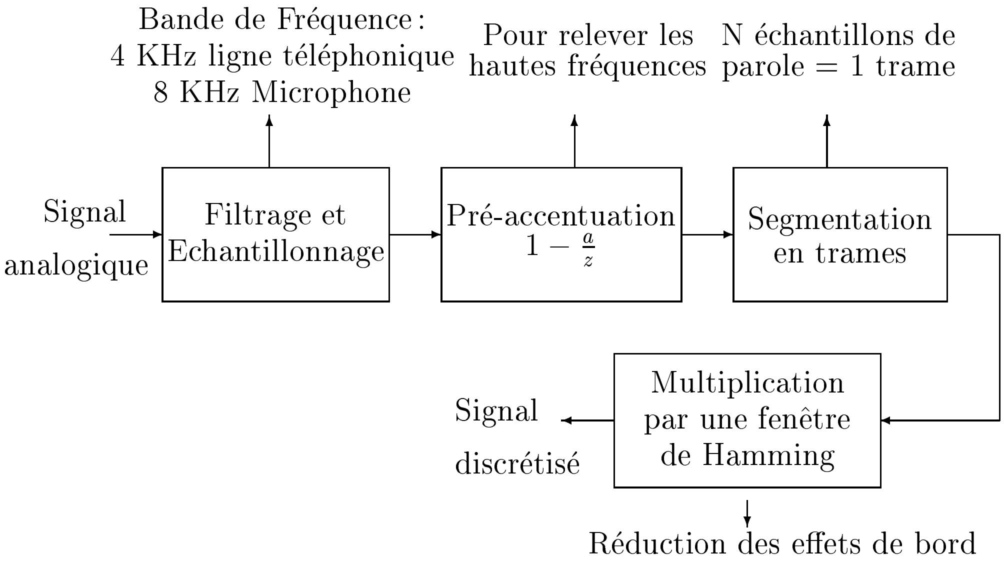

FIG. 2.2 - *Mise en forme du signal.*

Ces coefficients doivent représenter au mieux le signal qu'ils sont censés modéliser, et extraire le maximum d'informations utiles pour la reconnaissance. Nous étudierons dans ce paragraphe les coefficients les plus utilisés en reconnaissance de la parole. Nous commencerons par les coefficients cepstraux aussi appelés cepstres. Par la suite, nous mettrons en évidence les propriétés d'autres coefficients tels que les coefficients PLP et RASTA-PLP. Nous parlerons enfin brièvement des coefficients MFCC, CMS et LDA (Linear Discriminant Analysis, analyse discriminante linéaire). Avant tout calcul, il est nécessaire de mettre en forme le signal de parole. Pour cela, quelques opérations sont effectuées avant tout traitement. La figure 2.2 illustre l'ensemble de ces opérations. Le signal est tout d'abord filtré puis échantillonné à une fréquence donnée. Une pré-accentuation est effectuée afin de relever les hautes fréquences. Puis le signal est segmenté en trames. Chaque trame est constituée d'un nombre N fixe d'échantillons de parole. En général, N est fixé de telle manière que chaque trame corresponde à environ 30 ms de parole (durée pendant laquelle la parole peut être considérée comme stationnaire). Enfin, une multiplication par une fenêtre de "Hamming" [1] est effectuée, afin de réduire les effets de bords. Ce traitement implique une hypothèse importante du fait des limitations postérieures qu'elle occasionne :

Le signal vocal est supposé stationnaire sur une courte période.

Après cette mise en forme du signal (commune à la plupart des méthodes d'analyse de la parole), une transformée de Fourier (DFT (Transformée de Fourier Discrète), en particulier FFT (Transformée de Fourier Rapide)) est appliquée pour passer dans le domaine fréquentiel. Le spectre de puissance à court terme $P(\omega)$ est calculé selon :

$$P(\omega) = Re[X(\omega)]^2 + Im[X(\omega)]^2 = |X(\omega)|^2$$

où $X(\omega)$ est le spectre du signal temporel et ω représente la fréquence angulaire en $rad.s^{-1}$.

Ensuite, chaque méthode et type de coefficients ont des particularités que nous nous proposons de décrire dans les paragraphes suivants.

2.3.1 Les coefficients cepstraux.

Il existe deux méthodes de calcul des coefficients cesptraux couramment utilisées en reconnaissance de la parole :

- Analyse spectrale.

1. $w(n) = 0.54 + 0.46.cos(2\pi\frac{n}{N-1})$. Outre ce type de fenêtre couramment utilisée en reconnaissance de la parole, il existe plusieurs autres types de fenêtre telle que rectangulaire, Hanning, Blackman, ... La fenêtre de Hamming est un cas particulier de la fenêtre de Hanning.

– Analyse paramétrique.

Analyse spectrale.

Le signal de parole est modélisé en utilisant le spectre. Il est possible de calculer directement les coefficients cepstraux en suivant le processus décrit à la figure 2.3.

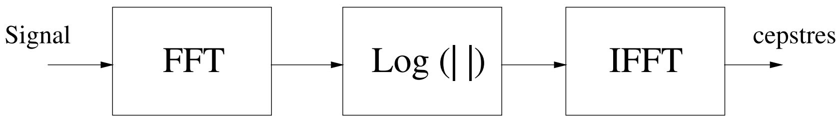

FIG. 2.3 - *Calcul des cepstres par analyse spectrale.*

Il suffit pour obtenir les coefficients cepstraux de prendre le spectre de puissance à court terme défini au début du paragraphe (c'est à dire la FFT (Transformée de Fourier Rapide) du signal, et sa norme[2]), puis de passer dans le domaine logarithmique et enfin de prendre la fonction inverse FFT (IFFT). Ce type de calcul des cepstres est généralement moins utilisé que la méthode paramétrique décrite ci-après du fait de la charge de calcul importante associée au calcul de la FFT et de la FFT inverse. Il est également possible de changer l'échelle de fréquence dans l'espace de Fourier pour obtenir des coefficients calqués sur une échelle "physiologique" correspondant mieux à la perception humaine. Une de ces transformations perceptuelles se fait selon l'échelle de Mel[3] [27] et les coefficients ainsi obtenus sont appelés MFCC (Mel-scale Frequency Cepstral Coefficients). La sélectivité de l'oreille diminue avec l'accroissement des fréquences. Par analogie, on utilise généralement des bancs de filtres dont la répartition reproduit cette sélectivité (échelles de Mel). Pour ce type de banc de filtres, la largeur de chacun d'eux augmente avec la fréquence centrale. Ceci permet, avec peu de filtres et tout en couvrant la bande passante, d'obtenir une bonne résolution dans les basses fréquences (là où se trouvent les premiers formants) et de garder suffisamment d'informations dans les hautes fréquences. Cette dernière représentation est actuellement considérée comme une des plus performantes en reconnaissance de la parole.

2. Seul le module de la FFT est retenu, la phase de la transformée de Fourier du signal de parole ne contient pas d'informations utilisables pour la reconnaissance de la parole.

3. Echelle de Mel : $Mel(f) = b.log_{10}(1 + \frac{f}{c})$ avec b = 2600 et c = 700, f représente la fréquence.

Analyse paramétrique.

Cette analyse permet de déterminer les coefficients cepstraux des signaux de la parole. L'étude du mécanisme de phonation montre qu'il est possible de modéliser le signal vocal par le passage d'un signal d'excitation $u(n)$ au travers d'un filtre dont la transmittance est $\frac{\sigma}{A(z)}$.

Cette hypothèse implique que le signal vocal peut alors être considéré comme un signal auto-régressif. On appelle signal auto-régressif d'ordre p, un signal qui exprime qu'un échantillon quelconque $x(n)$ est une combinaison linéaire des p échantillons qui le précèdent plus un terme d'excitation [14].

$$x(n) + \sum_{i=1}^{p} a(i).x(n-i) = \sigma.u(n) \tag{2.1}$$

Ainsi le signal vocal peut alors être considéré comme engendré par un filtre dont il faut trouver les coefficients a(i). Ce modèle de production du signal de parole est appelé AR (auto-régressif).

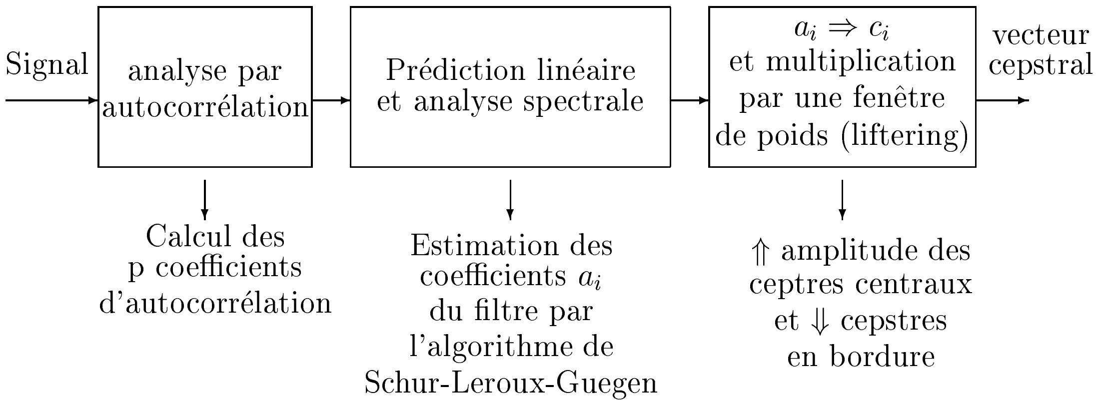

FIG. 2.4 - *Processus de calcul des cepstres par la méthode de prédiction linéaire*

L'obtention des coefficients $a(i)$ du modèle AR se fait en utilisant des méthodes classiques de prédiction linéaire[4](p désigne toujours l'ordre du filtre). La méthode de calcul des cepstres est basée sur une relation de récurrence liant

4. Analyse LPC, méthode de l'autocorrélation ou de la covariance [14].

les cepstres et les coefficients $a(i)$ (Voir figure 2.4) [2]. Ensuite un liftrage ("liftering") est effectué pour augmenter la robustesse des coefficients cepstraux [60, 86]. Ce liftrage consiste en une multiplication par une fenêtre de poids (voir formule 2.2) des coefficients cepstraux augmentant l'amplitude des coefficients connus pour être moins sensibles au canal de transmission et au locuteur [5]

$$\forall n \in [1, Q] \quad W(n) = 1 + \frac{Q}{2}.sin(\frac{\Pi.n}{Q}) \tag{2.2}$$

où Q est le nombre de coefficients. Cette méthode de prédiction linéaire est beaucoup plus utilisée en reconnaissance de la parole que celle de l'analyse spectrale. En effet, il existe de nombreux algorithmes (Schur, ou Leroux et Gueguen) qui peuvent être utilisés en temps réel, alors que l'analyse spectrale demande beaucoup plus de temps de calcul (calcul d'une FFT et d'une FFT inverse).

Il existe un autre modèle de production (transmittance du filtre de la forme : $\frac{\sigma.C(z)}{A(z)}$). Le modèle défini par cette transmittance est appelé ARMA (Auto Regressive Moving Average) pour lequel chaque échantillon est constitué par une combinaison linéaire de p échantillons passés et de (q+1) échantillons présents et passés de l'excitation (u).

$$x(n) + \sum_{i=1}^{p} a(i).x(n-i) = \sigma.\sum_{i=0}^{q} c(i).u(n-i) \tag{2.3}$$

L'estimation des modèles ARMA étant plus délicate que celle des modèles AR, on préfère utiliser les modèles auto-régressifs dans la plupart des applications. Ainsi pour l'analyse spectrale comme pour l'analyse paramétrique, le signal de parole a été transformé en une série de vecteurs cepstraux calculés pour chaque trame (correspondant à un nombre d'échantillons du signal d'entrée). Ces coefficients jouent un rôle capital dans les méthodes utilisées pour reconnaître la parole.

2.3.2 La soustraction cepstrale

La parole peut être modélisée par la convolution d'un signal de parole $x(n)$ et d'un terme relatif au canal de transmission $h(n)$. Soit en notant *

5. Ce type de technique s'accompagne d'une amélioration sensible des taux de reconnaissance pour les HMMs à distributions discrètes.

l'opérateur convolution :

$$y(n) = x(n) * h(n) \tag{2.4}$$

Dans ce cas, la transformée de Fourier discrète du signal $y(n)$ s'écrit[6]:

$$Y(f) = FFT(y(n)) = X(f).H(f) \tag{2.5}$$

Ce qui peut encore s'ecrire en prenant la norme et le log :

$$ln|Y(f)| = ln|X(f)| + ln|H(f)| \tag{2.6}$$

Reprenons maintenant la transformée de Fourier inverse pour calculer les coefficients cepstraux, il vient alors :

$$\tilde{y}(n) = \tilde{x}(n) + \tilde{h}(n) \tag{2.7}$$

Ce terme $\tilde{h}(n)$ est souvent appelé bruit convolutif ou distorsion du canal de transmission et influence fortement le processus de reconnaissance (puisqu'il a une influence sur les paramètres acoustiques). La soustraction cepstrale[7] (Cepstral Mean Substraction [41]) est en fait une technique de normalisation des paramètres acoustiques afin de les rendre moins sensibles à des facteurs extérieurs (bruits convolutifs).

Cette technique de normalisation consiste à calculer pour chaque locuteur (ou pour chaque phrase) la moyenne des vecteurs acoustiques et à effectuer la soustraction de cette moyenne pour obtenir de nouveaux paramètres acoustiques. On élimine ainsi l'influence de ce fameux bruit convolutif $\tilde{h}(n)$. Il va de soi que les taux de reconnaissance généralement observés avec cette technique de normalisation sont largement supérieurs à ceux correspondant aux vecteurs acoustiques classiques. En effet, on utilise une information qui n'est a priori pas accessible pour un système de démonstration (la moyenne des vecteurs acoustiques pour un locuteur ou une phrase donné).

Il est possible de calculer cette moyenne des vecteurs acoustiques en ligne, c'est-à-dire au fur et à mesure de l'arrivée des vecteurs acoustiques en initialisant cette moyenne à la moyenne globale des vecteurs acoustiques sur la base d'entraînement, et en recalculant cette moyenne pour chaque nouveau vecteur acoustique. Les taux de reconnaissance observés sont alors moins bons que pour le type de normalisation décrit précédemment, mais ce type de technique peut être utilisé pour un système de démonstration. Néanmoins, il est nécessaire d'attendre quelques secondes de parole avant que le processus de normalisation ne se stabilise. Les résultats obtenus avec le type de normalisation "offline" des paramètres acoustiques sont généralement largement

6. avec $f = \frac{n.f_e}{N}$ pour $n \in [0, N-1]$, f_e est la fréquence d'échantillonage
7. Notée CMS dans le reste du document.

supérieurs à ceux obtenus avec les paramètres acoustiques classiques[8], mais ils ne reflètent pas les réelles performances du système. De plus cette technique de normalisation produit une distorsion en présence de bruit additif.

2.3.3 Les coefficients PLP (Perceptual Linear Predictive).

L'estimation par prédiction linéaire (LP) des coefficients du modèle autorégressif (tout pôle) du spectre de la parole est très utilisée en reconnaissance de la parole. Néanmoins, un des principaux défauts de cette méthode est que le filtre « identifie » uniformément le spectre sur toutes les fréquences de la bande d'analyse. Or cette propriété est loin d'être vérifiée pour l'oreille humaine, car il a été établi que l'oreille humaine est plus sensible aux fréquences situées au milieu de la bande d'analyse du spectre (100 à 3000 Hz). Ainsi, il est possible que certains détails importants du spectre ne soient pas pris en compte lors de l'analyse LP. L'analyse PLP permet de résoudre ce problème [50]. Le but de cette analyse est d'estimer les paramètres d'un filtre auto-régressif tout pôle, modélisant au mieux le spectre auditif [51]. Le schéma général de cette méthode est visible sur la figure 2.5.

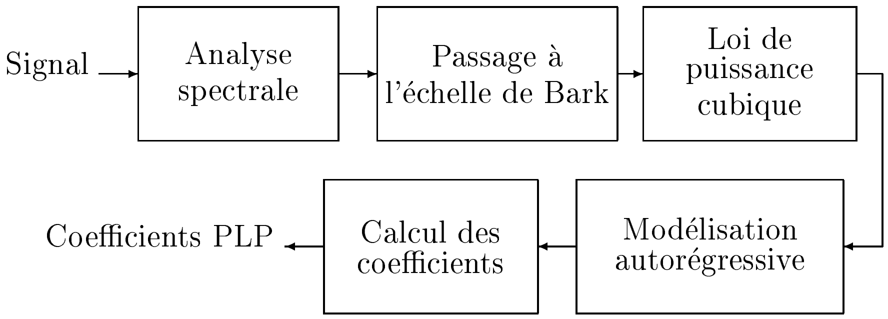

FIG. 2.5 - *Méthode de calcul des coefficients PLP.*

Observons plus en détail les différents blocs utilisés pour calculer les coefficients PLP. Comme défini auparavant, le spectre de puissance court terme $P(\omega)$ est calculé. Ensuite, un passage de l'échelle de fréquence usuelle à l'échelle de Bark est effectué. Ce passage à l'échelle de Bark, permet d'approximer de manière grossière ce que nous savons de la forme des filtres auditifs. La forme des filtres auditifs est approximativement constante le long de l'échelle de Bark. Le spectre d'énergie court terme dans l'échelle de

8. Notamment pour des canaux de transmission du type lignes téléphoniques.

Bark est convolué avec le spectre de puissance de la courbe de bande critique. ω représente la fréquence angulaire en $rad.s^{-1}$ et Ω la fréquence de Bark [9]. Cette transformation de l'échelle fréquentielle a été mise au point par Schroeder [104]

$$\Theta(\Omega_t) = \sum_{\Omega=-1.3}^{\Omega=2.3} P(\Omega - \Omega_t).\Psi(\Omega) \tag{2.8}$$

où $\Psi(\Omega)$ est la courbe de masquage :

$$\Psi(\Omega) = \begin{cases} 0 & \text{si } \Omega < -1.3 \\ 10^{2.5(\Omega+0.5)} & \text{si } -1.3 \leq \Omega \leq -0.5 \\ 1 & \text{si } -0.5 \leq \Omega \leq 0.5 \\ 10^{-1.0.(\Omega-0.5)} & \text{si } 0.5 \leq \Omega \leq 2.5 \\ 0 & \text{si } \Omega > 2.5 \end{cases}$$

On essaye ensuite d'approximer la sensibilité de l'oreille humaine à différentes fréquences par l'intermédiaire d'une fonction de transfert $E(\omega)$. Le spectre de puissance est multiplié par cette fonction de transfert.

$$E(\Omega) = E(\omega).\Theta(\Omega) \tag{2.9}$$

La non-linéarité entre l'intensité d'un son et sa force de perception par l'oreille est ensuite approximée par une loi de puissance :

$$\Gamma(\Omega) = E(\Omega)^{\frac{1}{3}}$$

Enfin, on effectue une modélisation auto-régressive classique du spectre du modèle auditif tout pôle, en calculant les coefficients auto-régressifs du filtre. Hermansky et al. [50] ont montré que les coefficients PLP introduisent une meilleure robustesse au bruit que les coefficients cepstraux classiques. Une extension de cette méthode a été développée et a donné de bons résultats : les coefficients RASTA-PLP [52]. Le filtrage RASTA (RelAtive SpecTrAl processing) consiste à supprimer les facteurs constants dans chaque composante spectrale du spectre auditif court terme avant l'estimation du modèle tout pôle. En effet, supposons que la parole soit corrompue par un bruit convolutif (exemple d'un changement des caractéristiques fréquentielles d'un canal

9. $\Omega(\omega) = 6.ln[\frac{w}{1200.\Pi} + ((\frac{w}{1200.\Pi})^2 + 1)^{\frac{1}{2}}]$.

de transmission causé par l'utilisation d'un nouveau microphone). Une telle distorsion apparaît sous la forme d'une constante additive dans le logarithme du spectre de la parole. La plupart des paramètres acoustiques et donc des systèmes de reconnaissance sont affectés par de tels phénomènes. Le filtrage RASTA supprime les composantes spectrales (ou log spectrales) qui varient plus lentement ou plus rapidement que le signal de parole et donc leurs influences sur les systèmes de reconnaissance de la parole. Une extension de cet algorithme nommée Adaptative lin-log RASTA ou J-RASTA permet d'effectuer une compensation pour les bruits à la fois convolutifs et additifs. Une transformation non linéaire dans le domaine spectrale est effectuée :

$$Y(f) = ln[1 + J.X(f)] \tag{2.10}$$

où J est une constante dépendante du rapport signal sur bruit (SNR). Cette transformation est du type linéaire pour de petites valeurs de J et logarithmique pour des valeurs élevées. Ce coefficient J peut être ajusté en le rendant inversement proportionnel à l'énergie moyenne du bruit. En utilisant ce type d'ajustement, Hermansky [52] a montré que les résultats obtenus avec ce type de coefficients sont comparables à ceux obtenus avec un système entraîné et testé dans les mêmes conditions de bruits (pour peu qu'il y ait suffisamment de données pour estimer l'énergie du bruit). Le filtrage log-RASTA étant moins sensible aux variations du canal de transmission (changement de microphone), il est le plus souvent à la base des systèmes de démonstration, c'est pourquoi nous l'utiliserons très souvent dans ce travail.

2.3.4 Les coefficients LDA

Plusieurs groupes de chercheurs [3, 47, 110] ont montré que l'analyse discriminante linéaire permettait d'améliorer les performances des systèmes de reconnaissance ainsi que leur robustesse à certains types de bruits. L'objet de l'analyse discriminante linéaire est de déterminer des paramètres adaptés au problème de classification. Les paramètres discriminants sont obtenus en calculant une transformation linéaire des vecteurs d'entrée x (de dimension n) vers des vecteurs y (de dimension m, $m < n$) de telle manière que la séparation des classes auxquelles sont assignés ces vecteurs lors de la classification soit maximale. Les critères d'optimisation le plus souvent rencontrés sont :

$$J_1 = tr(S_2^{-1} S_1) \tag{2.11}$$

$$J_2 = det(S_2^{-1} S_1) \tag{2.12}$$

où $tr(A)$ représente la trace de la matrice A et $det(A)$ son déterminant. Les matrices S_1 et S_2 sont deux matrices choisies parmi les matrices de covariance à l'intérieur des classes, entre les classes et mélange de classes [64]. On peut montrer que l'optimisation des critères J_1 et J_2 conduit aux mêmes paramètres discriminants et que cette optimisation est indépendante du choix des matrices de variance. Fontaine et al. ont aussi montré dans [37] qu'il était possible d'utiliser des réseaux de neurones pour effectuer une analyse discriminante non linéaire. Les expériences préliminaires ont effectivement montré que les réseaux étaient aptes à fournir des paramètres discriminants très efficaces pour les tâches de reconnaissance. Ce type de coefficients ne sera pas utilisé pour les expériences présentées dans cette étude.

2.3.5 Conclusion

Nous avons brièvement parcouru dans ce paragraphe les méthodes utilisées pour extraire les paramètres acoustiques qui seront fournis au système de reconnaissance. Nous n'avons fait que survoler ces techniques de calcul des coefficients représentatifs du signal vocal. Le lecteur pourra consulter, s'il le désire, les articles de base cités dans ce chapitre. Nous utiliserons principalement les coefficients PLP et log RASTA-PLP dans les expériences décrites dans ce travail. La justification de l'utilisation des coefficients log RASTA-PLP est leur indépendance vis-à-vis du canal de transmission, et donc leur possible utilisation dans un système de démonstration sans dégradations significatives des taux de reconnaissance par l'utilisation d'un microphone différent de ceux utilisés lors de l'enregistrement des bases de données ayant servi à l'entraînement des modèles. Nous donnerons aussi à titre de comparaison quelques résultats avec des coefficients PLP, MFCC et CMS.

2.4 Modèles de Markov cachés

2.4.1 Introduction

Le problème de la reconnaissance de la parole peut être formulé dans ces termes : Comment modéliser au mieux des unités représentatives du signal de parole ?

Il existe en fait deux types de modélisation possible des propriétés d'un signal donné :

- les modèles déterministes, qui exploitent les propriétés intrinsèques du signal,
- les modèles statistiques, qui caractérisent les propriétés statistiques du signal.

Dans ce travail, nous utiliserons des modèles statistiques précis : les modèles de Markov cachés (HMMs[10]) [57] qui se sont imposés comme la technologie prédominante en reconnaissance de la parole ces dernières années. Nous allons revoir rapidement les bases nécessaires à l'utilisation de ce type de modèle pour la reconnaissance automatique de la parole. Pour obtenir plus de renseignements, le lecteur pourra consulter des ouvrages de références tels que [16] [85] ou [87]. Ces modèles se sont avérés les mieux adaptés aux problèmes de la reconnaissance de la parole. La quasi-totalité des outils de reconnaissance de la parole disponibles actuellement sur le marché sont basés sur cette technologie. Un modèle de Markov caché est un automate stochastique particulier capable, après avoir été entraîné, d'estimer la probabilité qu'une séquence d'observations ait été générée par ce modèle. Idéalement, il faudrait pouvoir associer à chaque phrase possible un modèle. Il va de soi que ceci est irréalisable en pratique car le nombre de modèles serait beaucoup trop élevé. Des sous-unités lexicales comme le mot, la syllabe, ou le phonème sont utilisées afin de réduire le nombre de paramètres à entraîner. A chacune de ces unités est associé un modèle de Markov caché constitué d'un nombre fini d'états prédéterminés. Formellement, un modèle de Markov caché peut être défini par l'ensemble des paramètres λ [84]:

$$\lambda = \mathrm{L,A,B,\Pi}$$

1. L est le nombre d'états du modèle,
2. $A = a_{ij} = p(q_j|q_i)$ la matrice de transition sur l'ensemble des états du modèle (on définit ainsi une topologie par l'intermédiaire de cette matrice A),
3. $B = b_j(x_n) = p(x_n|q_j)$ l'ensemble des probabilités d'émission de l'observation x_n dans l'état q_j.
   - Dans le cas d'entrées discrètes (après quantification vectorielle des observations), les probabilités d'émission peuvent être décrites comme des fonctions de densité de probabilité discrète $p(y_i|q_j)$.

10. Dans le reste de ce document, l'abréviation anglaise HMMs sera utilisée couramment pour parler des modèles de Markov cachés.

- Dans le cas d'entrées continues ($x_n \in \Re^d$), $p(x_n|q_j)$ est :
  - supposée être de la forme d'une distribution multivariable gaussienne, entièrement définie par le vecteur moyenne et la matrice de covariance, ou d'une distribution de type multi-gaussienne (somme pondérée de gaussiennes, voir hypothèses HMM paragraphe 2.4.4).
  - supposée être représentée par un réseau de neurones (voir chapitre 3).

4. Π la distribution initiale des états,$\forall j \in [1, L]$ $p(q_j|q_I)$ [11].

En reconnaissance de la parole, des modèles de Markov gauche-droite d'ordre 1 sont le plus souvent utilisés du fait de l'aspect séquentiel du signal de la parole [9]. Un modèle de Markov à 3 états est visible sur la figure 2.6 [12].

Les modèles de Markov cachés (HMMs) supposent que la séquence de vecteurs acoustiques représentative du signal de parole soit une succession de segments stationnaires. Ainsi la parole est modélisée par une succession d'états, avec des transitions instantanées possibles entre ces états $p(q_j|q_k)$. Chaque observation est supposée être une fonction probabiliste de l'état.

Deux processus stochastiques concurrents sont observés :

- un premier processus qui est la séquence d'observations $X = X_1, ..., X_N$,
- un second processus, la séquence d'états non directement observable.

C'est pourquoi ces modèles sont dits "cachés". La séquence d'états n'est pas directement observable.

2.4.2 Problèmes à résoudre

Soit M le modèle de Markov caché associé à la phrase 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 $P(M|X, \lambda)$ (probabilité qu'un modèle M génère une séquence de vecteurs acoustiques X étant donné une série de paramètres λ). Cette

11. q_I représente l'état initial du modèle HMM, il ne peut émettre de vecteurs acoustiques.

12. D'autres topologies sont aussi utilisées. Citons par exemple les modèles de Markov avec modélisation de la durée [1, 100] qui améliorent sensiblement les taux de reconnaissance ou les modèles de Bakis.

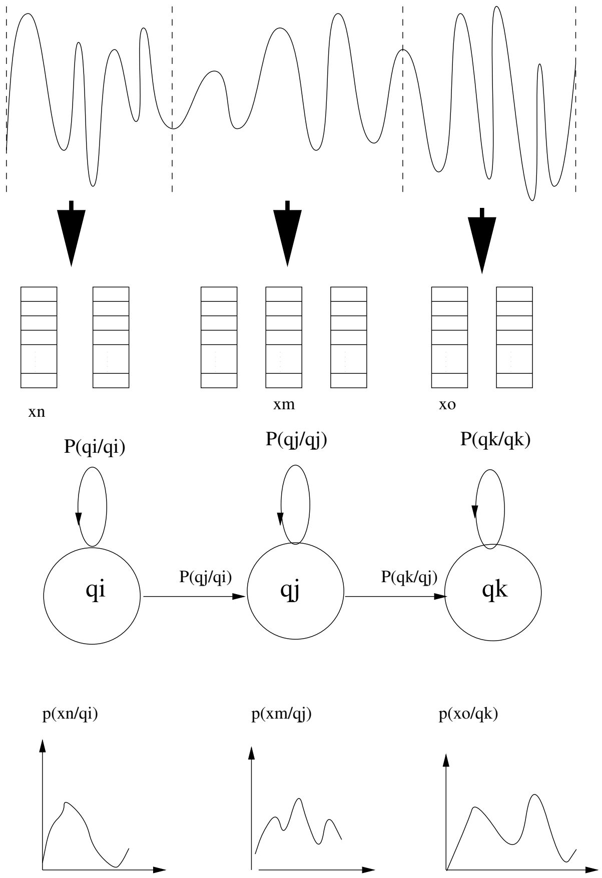

FIG. 2.6 - *Exemple d'un modèle de Markov d'ordre 1 à trois états caractérisés par une distribution de probabilités pour chaque état associé à une observation et par des probabilités de transitions entre les états.*

probabilité est aussi appelée probabilité a posteriori. Malheureusement, il n'est pas possible d'accéder directement à cette probabilité par le processus d'entraînement des modèles de Markov, mais seulement à la probabilité qu'un modèle donné générera une certaine séquence de vecteurs acoustiques $P(X|M)$.

En utilisant la loi de Bayes (2.13), il est possible de lier ces deux probabilités selon :

$$P(M|X) = \frac{P(X|M).P(M)}{P(X)} \tag{2.13}$$

où

– $P(X|M)$ est la vraisemblance de la séquence d'observations X étant donné 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'entraînement des paramètres $\lambda = \{A, B, \Pi\}$:

- Critère MAP (Maximum a Posteriori Probability) : maximum a posteriori.
- Critère MLE (Maximum Likelihood Estimation) : maximum de vraisemblance.

Les trois lois de probabilités suivantes seront couramment utilisées dans ce chapitre et le reste du document :

$$P(a, b) = P(a|b).P(b) = P(b|a).P(a) \tag{2.14}$$

$$P(a, b|c) = P(a|b, c).P(b|c) \tag{2.15}$$

Si les événements b_k sont mutuellement exclusifs et collectivement exhaustifs :

$$P(a) = \sum_{\forall k} P(a, b_k) \tag{2.16}$$

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 (voir paragraphe 2.4.9).
- **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 $\frac{P(X|M).P(M)}{P(X)}$ revient à maximiser $P(X|M).P(M)$.

Il faut alors résoudre 3 problèmes liés à ces modèles [16, 84, 87].

1. 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?

2. L'entraînement : étant donné une séquence d'observations 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 maximiseront $P(M|X, \lambda)$ pour l'ensemble des séquences de vecteurs acoustiques X associé au modèle M [13]? Cette probabilité n'étant pas directement accessible, on préfère maximiser $P(X|M, \lambda)$ (soit utiliser le critère MLE plutôt que MAP) [14].

3. Le décodage : étant donné une séquence de modèles de Markov avec leurs paramètres entraînés et une séquence d'observations X, comment trouver la meilleure séquence M_k de modèles de Markov élémentaires pour maximiser la probabilité que M_k ait généré les observations?

2.4.3 Problème 1 : Estimation des probabilités.

Le problème de l'estimation des probabilités peut être énoncé de la façon suivante : étant donné un modèle de Markov M, comment calculer la probabilité $P(X|M)$ qu'il génère la séquence de vecteurs acoustiques X?

De manière générale, le calcul de cette probabilité s'effectue selon :

$$P(X|M) = \sum_{\Gamma} P(C, X|M) = \sum_{\Gamma} P(Q_1^N, X|M)$$

où

- Γ = ensemble des chemins autorisés C dans le modèle M.
- $Q_1^N = \{q_I = q^0, q^1, ..., q^N, q^{N+1} = q_F\}$ est une séquence ordonnée de N états [15],
- q^n représentant l'état du HMM à l'instant n,
- q_k^n signifie que l'état q_k est visité à l'instant n.

13. Ce critère est discriminant car il minimise le taux d'erreur.

14. Le critère MLE n'est plus discriminant ce qui constitue une des premières limitations des modèles de Markov cachés.

15. Les états finaux q_F et initiaux q_I sont exclus car ils ne peuvent pas émettre de vecteurs acoustiques. Ils sont généralement utilisés pour initialiser les équations récurrentes.

Il existe deux procédures récurrentes de calcul de cette probabilité que nous nous proposons de décrire :

- L'algorithme Forward-Backward qui fournit une solution exacte à ce problème faisant intervenir tous les chemins dans le modèle HMM.
- L'algorithme Viterbi fournissant une solution approximative faisant intervenir uniquement le meilleur chemin dans le modèle HMM.

L'algorithme Avant-Arrière (Forward-Backward)

L'algorithme "Forward-Backward" [10] peut être utilisé pour calculer $P(X|M)$ de manière récursive en posant :

$$\forall l \in [1, L] \quad \alpha_n(l|M) = P(q_l^n, X_1^n|M) \tag{2.17}$$

il vient alors (voir annexe B):

$$\forall l \in [1, L] \quad \alpha_{n+1}(l|M) = [\sum_{k=1}^{L} \alpha_n(k|M).p(q_l|q_k, M)].p(x_{n+1}|q_l) \tag{2.18}$$

avec $\alpha_1(l) = p(x_1|q_l).p(q_l|q_1)$

De même posons :

$$\forall l \in [1, L] \quad \beta_n(l|M) = P(X_{n+1}^N|q_l^n, X_1^n, M) \tag{2.19}$$

il vient alors (voir annexe B):

$$\forall l \in [1, L] \quad \beta_n(l|M) = \sum_{k=1}^{L} p(q_k|q_l, M).p(x_{n+1}|q_k).\beta_{n+1}(k|M) \tag{2.20}$$

avec $\forall l \in [1, L] \quad \beta_N(l) = 1 \quad ou \quad 0$

Et en utilisant ces deux dernières formulations il est possible de calculer $P(X|M)$ selon :

$$\forall n \in [1, N] \quad P(X|M) = \sum_{l=1}^{L} \alpha_n(l|M).\beta_n(l|M) \tag{2.21}$$

Où les termes $\alpha_n(l|M)$ et $\beta_n(l|M)$ peuvent être calculés de manière récurrente. De plus en posant n = N , q_F état final et q_I état initial :

$$P(X|M) = \sum_{\forall l \in \Im} \alpha_N(l|M) = \alpha_{N+1}(q_F|M) = \beta_0(q_I|M) \tag{2.22}$$

Remarque : $\Im$ représente l'ensemble des états finaux.

C'est souvent cette dernière formulation qui est utilisée en reconnaissance de la parole, car elle ne nécessite que le calcul des termes α pour déterminer la probabilité $P(X|M)$ [16].

Il faut noter que les équations décrites ici ont nécessité une série d'hypothèses simplificatrices et limitatrices qui sont rappelées en annexe B avec le détail des calculs. Le lecteur pourra aisément trouver la justification de l'utilisation de ces hypothèses simplificatrices. Les équations présentées ici sont basées sur le critère du maximum de vraisemblance. Il est possible d'utiliser un autre critère permettant de réduire sensiblement la charge de calcul : le critère de Viterbi [38, 115].

L'algorithme de Viterbi

Au lieu de prendre en compte tous les chemins autorisés, seul le plus probable est gardé. Ainsi, il suffit de remplacer dans les équations précédentes l'opérateur $\sum$ par max. Ce critère est largement utilisé en reconnaissance de la parole du fait du faible coût qui lui est associé [17].

L'algorithme de Viterbi est donc une simplification de la récurrence avant qui devient :

$$P(q_l^n, X_1^n|M_j) = \max_k \; [P(q_k^{n-1}, X_1^{n-1}|M_j).p(q_l^n|q_k^{n-1}, M_j)].p(x_n|q_l^n, M_j) \tag{2.24}$$

Soit en passant au log :

$$\begin{aligned} -\,log(P(q_l^n, X_1^n|M_k)) \quad = \quad & \min_k \; [-log(P(q_k^{n-1}, X_1^{n-1}|M_j)) - log(p(q_l^n|q_k^{n-1}, M_j))] \\ & -log(p(x_n|q_l^n, M_j)) \end{aligned} \tag{2.25}$$

16. Bien que le calcul des α et β permette aussi de calculer la probabilité suivante qui est utilisée (nous le verrons plus loin) lors de l'entraînement des modèles HMMs

$$P(q_k^n, X|M) = \alpha_n(k).\beta_n(k) \tag{2.23}$$

17. En effet, il est évident que l'opération max est moins coûteuse en temps de calcul que l'opération $\sum$ sur tous les états.

Cette dernière équation montre que le calcul de la probabilité $P(X|M)$ est très semblable à une récurrence du type DTW (Dynamic Time Warping). Les algorithmes de programmation dynamique utilisés pour la reconnaissance de la parole seront décrits plus précisément dans le paragraphe 4.2.

2.4.4 Hypothèses simplificatrices

L'utilisation pratique de ces modèles ne peut se faire qu'après avoir fixé quelques hypothèses qu'il est important de rappeler (voir Annexe B).

- **H1** On a supposé que $P(M)$ peut être calculé indépendamment. 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.
- **H3** Les modèles de Markov sont supposés du premier ordre; ainsi la probabilité que la chaîne de Markov soit dans l'état q_l au temps n dépend uniquement de l'état de la chaîne de Markov au temps n-1 et est indépendante du passé.
- **H4** Les vecteurs acoustiques ne sont pas corrélés, la probabilité qu'un vecteur acoustique soit émis au temps n dépend uniquement de la transition de l'état q_k^{n-1} à q_l^n et est indépendante du passé.
- **H5** La probabilité d'émission est supposée dépendante uniquement de l'état courant pour réduire le nombre de paramètres.

Pour calculer la probabilité d'émission, chaque état q_l doit être associé à une densité de probabilité $p(x_n|q_l)$. Il faut donc émettre des hypothèses supplémentaires à propos de cette densité de probabilité.

- **H6** Dans le cas d'entrées continues, $p(x_n|q_l)$ est supposée être de la forme d'une distribution multivariable gaussienne [59]. Ainsi la probabilité peut être exprimée selon :

$$p(x|q_l) = \sum_{j=1}^{N} c_{lj}.N_{lj}(x) \tag{2.26}$$

où $N_{lj}(x)$ désigne la valeur au point x (trame acoustique) de la gaussienne dont les paramètres sont décrits ci-après.

$$N_{lj}(x) = \frac{1}{\sqrt{(2\Pi)^d.det(\Sigma_{lj})}}.e^{-\frac{1}{2}.(x-\mu_{lj})^T.\Sigma_{lj}^{-1}.(x-\mu_{lj})} \quad (2.27)$$

avec :

- μ_{lj} : vecteur moyen pour la gaussienne j de l'état q_l,
- Σ_{lj} : matrice de covariance pour la gaussienne j de l'état q_l,
- c_{lj} : coefficients de pondération.
- d : dimension de l'espace dans lequel sont mesurées les trames acoustiques.

Les systèmes basés sur ce type de modélisation sont appelés *systèmes à densités d'observations continues* ou *systèmes multi-gaussiens* [6, 7]. L'utilisation de plusieurs gaussiennes permet une meilleure modélisation de la probabilité $p(x_n|q_l)$. De plus, en général, les différentes composantes du vecteur x_n sont supposées non corrélées[18]. Le nombre de paramètres utilisés est alors beaucoup moins important car les matrices de covariances des distributions sont diagonales. De nombreux systèmes de reconnaissance sont basés sur ce type d'estimation des probabilités d'émissions. Malheureusement, ces modèles nécessitent un temps de calcul élevé pour estimer les probabilités (donc pour l'entraînement et la reconnaissance). Il est en effet nécessaire de calculer un grand nombre de densités gaussiennes pour chaque trame acoustique et chaque état HMM. Une solution intermédiaire consiste à utiliser des modèles de Markov semi-continus [11]. Ces modèles utilisent un dictionnaire de gaussiennes partagées par toutes les distributions associées à chaque état. Ce type de modèle a permis d'obtenir de très bonnes performances (du point de vue du taux de reconnaissance et du temps de calcul) pour la reconnaissance de la parole continue "grand vocabulaire" en combinant les avantages des deux méthodes (modèles continus et discrets) présentées dans ce paragraphe.

- **H7** Dans le cas d'entrées discrètes, l'hypothèse H6 n'est plus nécessaire. La séquence de vecteurs acoustiques X est quantifiée. Chaque vecteur acoustique x_n est remplacé par un centroïde y_i (le plus proche au sens d'une distance prédéfinie[19]) sélectionné dans un dictionnaire

18. C'est encore une hypothèse supplémentaire à ajouter.
19. Euclidienne par exemple.

prédéterminé (codebook [20]). Ainsi les probabilités d'émission peuvent être décrites comme des fonctions de densité de probabilité discrète $p(y_i|q_l)$. Ce type de système est couramment appelé *système discret*. Le temps de calcul nécessaire à l'utilisation de ces modèles est beaucoup moins important que pour les systèmes continus, mais les performances restent limitées.

2.4.5 Problème 2 : Estimation des paramètres et entraînement des modèles

Le but de l'entraînement des modèles acoustiques est de trouver l'ensemble des paramètres λ maximisant sur l'ensemble des données d'entraînement X_j la vraisemblance des données étant donné les modèles associés M_j, soit :

$$\underset{\lambda}{\operatorname{argmax}} \prod_{j=1}^{J} P(X_j|M_j, \lambda_j) \tag{2.28}$$

Il est nécessaire d'estimer deux ensembles de paramètres :

- Les probabilités de transitions entre les états : a_{ij}. Il faut noter que dans toutes les expériences décrites dans ce travail, ces paramètres a_{ij} ne sont pas entraînés. Ils sont fixés à 0.5 (même probabilité de boucler sur lui-même ou d'effectuer une transition) ou à 0 et 1 si on utilise un modèle de durée (voir paragraphe 2.4.8). Il faut néanmoins rester conscient qu'il est possible de les estimer par l'intermédiaire des récurrences décrites dans ce chapitre.

- Les probabilités d'émission des observations pour chaque état : $b_j(x_n)$

Les approches les plus utilisées sont basées sur des adaptations de l'algorithme EM (Expectation-Maximization) [29] appelées :

- Algorithme de Baum-Welch : $P(X|M)$ est estimée en tenant compte de tous les chemins possibles (voir paragraphe précédent), implémentation de l'algorithme Expectation-Maximization (EM).

- Algorithme de Viterbi : $P(X|M)$ est estimée en tenant compte du meilleur chemin uniquement (approximation de l'algorithme EM).

20. Ce dictionnaire est obtenu par une technique consistant à regrouper l'ensemble des vecteurs acoustiques en classes. Chaque classe contient des vecteurs relativement proches les uns des autres au sens de la distance choisie. Le centroïde d'une classe est alors défini comme la moyenne de l'ensemble des vecteurs de cette classe et est utilisé comme vecteur prototype.

Entraînement Baum-Welch

L'algorithme de Baum Welch est un processus itératif où, à chaque itération, de nouvelles valeurs des paramètres λ des modèles sont estimées à partir des anciennes valeurs. L'entraînement des modèles est effectué à partir de l'estimation de $P(X|M)$ en tenant compte de tous les chemins possibles. On utilise les formules de récurrence "avant" et "arrière" définies dans ce chapitre. Nous avons déjà montré en posant :

$$\forall l \in [1, L] \quad \alpha_n(l|M) = P(q_l^n, X_1^n|M)$$

$$\forall l \in [1, L] \quad \beta_n(l|M) = P(X_{n+1}^N|q_l^n, X_1^n, M)$$

que :

$$\forall n \in [1, N] \quad P(X|M) = \sum_{l=1}^{L} \alpha_n(l|M).\beta_n(l|M)$$

et

$$\gamma_n(k|M) = P(q_k^n|X, M) = \frac{\alpha_n(k|M).\beta_n(k|M)}{P(X|M)}$$

Dans le cas d'une distribution monogaussienne, les formules de réestimations des paramètres sont données par les équations suivantes :

$$\mu_i = \frac{\sum_{n=1}^{N} \gamma_n(i|M).X_n}{\sum_{n=1}^{N} \gamma_n(i|M)} \tag{2.29}$$

$$\Sigma_i = \frac{\sum_{n=1}^{N} \gamma_n(i|M).(X_n - \mu_i).(X_n - \mu_i)^T}{\sum_{n=1}^{N} \gamma_n(i|M)} \tag{2.30}$$

où μ_i et Σ_i sont la moyenne et la matrice de covariance de l'état i. L'extension au cas multigaussiennes se fait aisément.

Le processus itératif mis en oeuvre consiste donc en deux phases :

- Une phase d'estimation où les récurrences avant et arrière nous permettent d'obtenir $p(q_k^n|X, M, \lambda)$ à partir d'un ensemble de paramètres λ.
- Une étape de maximisation où les paramètres λ sont mis à jour en utilisant les formules décrites dans ce paragraphe.

Entraînement Viterbi

Les paramètres sont optimisés de façon à maximiser la vraisemblance du meilleur chemin. Cela revient à supposer que les probabilités $\gamma_n(k) = P(q_k^n|X, M)$ sont égales à 0 ou 1. Comme pour l'algorithme EM classique, on part d'un ensemble de paramètres initiaux λ^0 et les paramètres optimaux λ sont obtenus de manière itérative. Le processus d'entraînement est composé d'une étape d'estimation (E) qui sert à trouver la segmentation qui maximise la vraisemblance à partir des paramètres, et d'une étape de maximisation (M), qui effectue une mise à jour des paramètres étant donné cette segmentation. L'ensemble des paramètres initiaux λ^0 peut être estimé à partir d'une segmentation initiale (linéaire ou autre, voir chapitre 6.4) ou à partir de modèles déjà entraînés (par exemple par l'intermédiaire d'un corpus déjà segmenté comme TIMIT [122]). Il est ensuite possible à partir de la segmentation optimale trouvée de calculer les paramètres des fonctions de vraisemblance en considérant tous les vecteurs associés à chacune des classes. Ce processus de réalignement des données acoustiques à l'aide d'un modèle et de réentraînement d'un nouveau modèle[21] est effectué jusqu'à ce qu'une certaine convergence soit atteinte (la segmentation ne varie plus ou l'accroissement relatif de la vraisemblance pour l'ensemble des données d'entraînement est inférieur à un seuil fixé).

2.4.6 Problème 3 : le décodage

Après l'entraînement des paramètres des modèles HMM, la reconnaissance d'une séquence de vecteur acoustique X correspondant à une phrase s'effectue par le calcul de la probabilité $P(X|M_j)$ pour tous les modèles (ou séquence de modèles) M_j. Ce calcul peut être effectué en utilisant les récurrences déjà décrites dans ce chapitre (voir équations (2.17), (2.19) et (2.21) pour le critère Baum-Welch et (2.24), (2.25) pour le critère Viterbi). Il faut toutefois noter que dans le cas de grands vocabulaires, ces récurrences sont coûteuses en temps de calcul. De plus, dans le cas de la parole continue, le nombre de combinaisons possibles des modèles (séquences de M_j) est très important. Ainsi, des techniques d'élagage ("pruning") ont été développées pour permettre l'utilisation de ces récurrences. Ces techniques seront détaillées dans le paragraphe 4.2. La procédure de reconnaissance consiste à trouver le modèle (ou la séquence de modèles) M_k pour lequel :

$$k = \underset{\forall j}{\operatorname{argmax}}\ P(M_j|X) = \underset{\forall j}{\operatorname{argmax}}\ P(X|M_j).P(M_j) \tag{2.31}$$

21. comme pour l'entraînement Baum-Welch décrit précédemment.

Pour une reconnaissance de la parole continue, le score du modèle acoustique $P(X|M_j)$ doit être multiplié par la probabilité de la séquence de mots associée $P(M_j)$ [22]. Nous verrons dans le paragraphe 2.4.9 comment il est possible de calculer ce terme.

2.4.7 Modèles utilisés

Les modèles de Markov cachés peuvent être utilisés à plusieurs niveaux linguistiques pour modéliser la parole (phonèmes, syllabes, mots [103, 119]...). L'unité la plus naturelle pour effectuer la reconnaissance de la parole est le mot. Les modèles de mots permettent une bonne modélisation de la coarticulation à l'intérieur du mot. Ils sont le plus souvent utilisés dans des systèmes de reconnaissance de mots clés ou pour des vocabulaires de petite ou moyenne taille (par exemple la reconnaissance de chiffres connectés ou de nombre quelconques). Néanmoins, l'utilisation de ces modèles génère quelques problèmes :

- il est difficile d'entraîner de nouveaux modèles de mots. Il faut en effet plusieurs enregistrements par différents locuteurs pour estimer les paramètres du nouveau mot.
- pour un nombre élevé de mots, les données d'entraînement nécessaires sont énormes.

C'est pourquoi, dans la plupart des systèmes de reconnaissance de la parole "grand vocabulaire", de plus petites unités linguistiques comme les phonèmes sont utilisées. En effet, le nombre de phonèmes permettant de construire n'importe quel mot présent dans la plupart des langages est faible. En français, tous les mots du vocabulaire peuvent être formés à partir de 35 phonèmes [23]. Du fait de leur petit nombre, ils peuvent être facilement entraînés avec des bases de données relativement petites [24]. De plus, il suffit d'ajouter les transcriptions phonétiques associées à un nouveau mot pour pouvoir en effectuer la reconnaissance (il n'y a plus d'entraînement à réaliser, le nouveau mot est modélisé par une concaténation des modèles de phonèmes définis par la transcription (voir figure 2.7)). Une bonne unité phonétique se doit de satisfaire aux deux critères suivants [67] :

- *Stationnarité* :
  Différents exemples de la même unité doivent avoir des réalisations acoustiques similaires.

22. L'équation 2.31 utilise la loi de Bayes et suppose que $P(X)$ est constant et indépendant de j lors de la reconnaissance.

23. On dénombre 35 phonèmes en Français, 47 en Anglais et 48 en Allemand.

24. Il y a une taille minimale au dessous de laquelle les modèles ne sont pas bien entraînés.

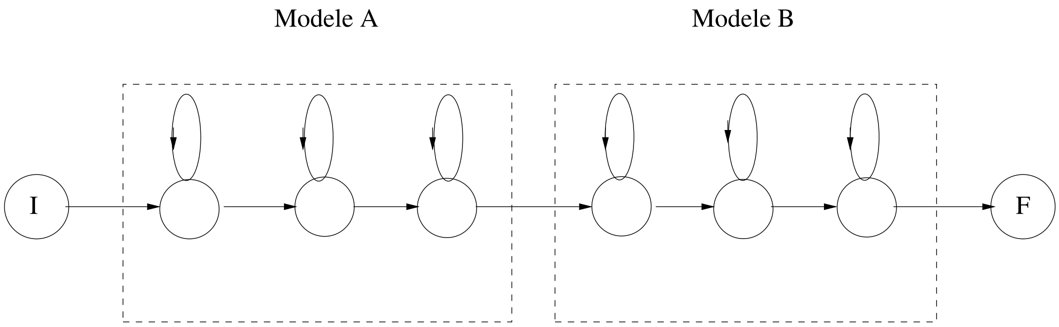

FIG. 2.7 - *Exemple de combinaison de modèles de phonèmes pour former un modèle de mot. L'état final du modèle A est remplacé par le premier état émetteur du modèle B pour former le mot M possédant la transcription "A B".*

- *Apprentissage*:
  Il doit y avoir suffisamment d'exemples de chacune des unités pour garantir une bonne estimation des paramètres associés à chacun des modèles.

Néanmoins, les phonèmes sont des unités linguistiques très petites et leur réalisation acoustique est extrêmement variable. Ainsi comme compromis entre les deux types de modèles (mots et phonèmes), de nombreux systèmes de reconnaissance de la parole utilisent des modèles de phonèmes dépendants du contexte. Ces modèles sont capables de capturer plus facilement les variations acoustiques et sont en nombre beaucoup moins important que les mots. Différents degrés de contexte peuvent être utilisés. Les modèles utilisant uniquement un contexte gauche (ou droit) sont appelés diphones, ceux utilisant à la fois le contexte gauche et droit sont appelés triphones. Ce dernier type de modèle permet d'obtenir des taux de reconnaissance plus performants que les modèles de phonèmes car l'information contextuelle est utilisée à part entière [68, 105]. Le choix de cette unité acoustique dépend du type de reconnaissance effectuée (isolée ou continue), de la taille du vocabulaire et même du type de densité de probabilité utilisée. En effet, les modèles de Markov continus sont le plus souvent associés à des modèles de phonèmes dépendants du contexte alors que les modèles hybrides[25] utilisent généralement des modèles de phonèmes indépendants du contexte. Il est, en outre, possible de modéliser les transitions entre les mots en utilisant des modèles spécifiques [109]. Le nombre de triphones et de diphones (respectivement n^3 et n^2 au maximum, où n est le nombre de phonèmes) étant poten-

25. que nous détaillerons précisément dans le chapitre 3.

tiellement élevé, il est courant d'utiliser en reconnaissance de la parole "grand vocabulaire" des phonèmes généralisés. Ces phonèmes offrent la possibilité de réduire le nombre de HMMs associés aux modèles tout en conservant le potentiel accru de modélisation. Le qualificatif "généralisé" fut introduit pour la première fois par Kai Fu Lee [67]. Un triphone généralisé est un modèle de triphone obtenu par regroupement de modèles de triphones similaires. L'idée proposée par Lee est de combiner de façon successive les HMMs jugés similaires selon un critère défini. En fait, certains contextes gauches et/ou droits similaires risquent d'avoir le même effet sur le phonème étudié. Ainsi il est possible de combiner les triphones représentant ces contextes. Par exemple, la réalisation acoustique du phonème "a" a de fortes chances d'être similaire lorsque celle-ci est coincée entre l'une des deux fricatives "s" et "z" d'une part et l'une des deux plosives "p" et "b" d'autre part. Il est possible d'utiliser un seul HMM pour représenter les triphones "s-a-b", "s-a-p", "z-a-b" et "z-a-p". Il existe de nombreuses méthodes pour effectuer cette généralisation. Il est, en effet, possible comme dans l'exemple précédent de se baser sur les classes phonétiques des phonèmes droit et gauche. Un algorithme de quantification automatique de tous les phonèmes en transition peut aussi être défini [35]. Chaque état étant associé à une gaussienne, un algorithme "K-Means" peut être utilisé pour quantifier ces gaussiennes. Une autre possibilité souvent utilisée dans la littérature consiste à utiliser des arbres de décision pour classer les diphones. Ces arbres de décision nécessitent une connaissance très précise de la nature et des particularités de chacun des éléments à classer. L'utilisation de phonèmes généralisés est très fréquente en reconnaissance de la parole, particulièrement dans le cas de grand vocabulaire en parole continue utilisant des modèles HMMs continus. Nous verrons dans le chapitre 8 qu'il est possible de les utiliser aussi avec succès dans les modèles hybrides.

2.4.8 L'architecture du modèle

Dans tout système de reconnaissance de la parole, il est nécessaire de modéliser de manière explicite la durée des phonèmes pour améliorer les taux de reconnaissance au niveau du mot. Le modèle de durée le plus souvent employé avec les modèles HMMs à distributions discrètes et multigaussiennes et les modèles hybrides est décrit sur la figure 2.8.

Les paramètres du modèle correspondant sont :

- la durée minimum du modèle N,
- la valeur des N-1 premières probabilités de transitions a,
- la valeur de la probabilité de boucler sur le dernier état b,

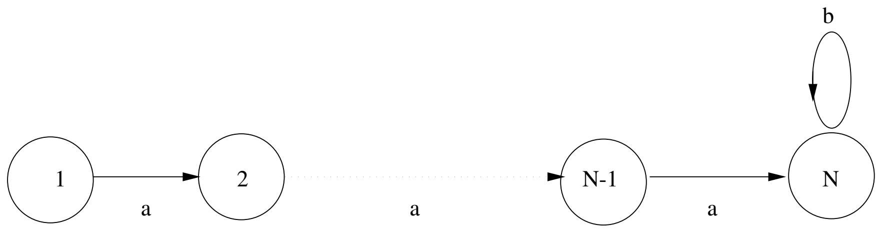

FIG. 2.8 - *Le modèle de phonème HMM de base. Le modèle impose une durée minimum de N trames et tous les états partagent la même distribution.*

Ces paramètres sont le plus souvent déterminés de la manière suivante: la durée minimale N est fixée à la moitié de la durée moyenne du phonème (estimée à partir d'un alignement Viterbi des données d'entraînement). Les paramètres a et b sont fixés de manière arbitraire à 1 et à 0.5. Enfin n'oublions pas un dernier paramètre, le plus souvent optimisé sur une partie de développement de la base de données : la probabilité de transition entre les mots. Ce paramètre utilisé en reconnaissance de la parole continue joue un rôle important sur les taux d'insertions et de suppressions de mots. Dans toutes les expériences décrites dans ce travail, les modèles utilisés seront du même type que celui décrit dans cette partie. De nombreuses autres modélisations de la durée intrinsèque des phonèmes ont été testées, mais les meilleurs résultats ont toujours été obtenus avec la modélisation décrite ici.

2.4.9 Grammaire et modèles de langage

Un modèle de langage [92] calcule une probabilité pour toute séquence de mots. Ces probabilités peuvent diriger la procédure de recherche parmi toutes les hypothèses possibles. Un bon modèle de langage permet une augmentation du taux de reconnaissance et une réduction de la complexité de la procédure de recherche [80]. La grammaire la plus simple est la grammaire "wordpair" [26]. Elle fixe l'ensemble des successeurs possibles pour un mot donné (décision 0/1).

De manière générale, on parle en reconnaissance de la parole de N-gramme (sachant que pratiquement N=2,3) qui permettent d'obtenir la probabilité d'un mot étant donné les N-1 mots précédents. Ainsi, un modèle de langage bigramme (respectivement trigramme) permet d'obtenir la probabilité

26. Que nous pouvons traduire par paire de mots.

d'un mot étant donné le mot (resp. les deux mots) précédent(s). Ces modèles peuvent être utilisés pour pondérer les séquences de mots reconnues. Un poids plus important peut être donné aux séquences de mots grammaticalement correctes. En reconnaissance de la parole des bigrammes, trigrammes et même dernièrement des multigrammes sont utilisés [28]. Ces modèles aident fortement le processus de reconnaissance [45].

La perplexité est une mesure permettant de comparer différents modèles de langage. De manière mathématique elle est définie par la formule suivante (liée à l'entropie) :

$$log_2(perplexity) = H \tag{2.32}$$

où H est l'entropie moyenne du modèle de langage et log_2 le logarithme base 2. De manière un peu plus simplifiée, la perplexité représente le nombre moyen de choix (généralement de mots) possibles à chaque noeud de décision. Ainsi dans le cas particulier de la grammaire wordpair de RESOURCE MANAGEMENT, la perplexité est de 60. Cela veut dire que pour un mot donné, il y a en moyenne 60 successeurs possibles. Plus la perplexité est élevée et moins les performances du système de reconnaissance seront bonnes.

Les probabilités utilisées dans les différents modèles de langage doivent être estimées sur de très grandes bases de textes. Il suffit pour estimer la probabilité du trigramme $P(w_3|w_1, w_2)$ de calculer la fréquence d'apparition du trigramme sur un corpus d'entraînement [26]. Il va de soi que les corpus utilisés pour estimer ces modèles de langage sont énormes. Il est courant de voir des corpus de 1 à 500 millions de mots pour estimer des trigrammes. En effet pour un vocabulaire de 20.000 mots, il y a 8 milliards (correspondant à 20.000^3) de trigrammes possibles. Un grand nombre de ces trigrammes ne sont jamais observés dans le corpus d'entraînement. Ainsi la probabilité associée à ces trigrammes (en utilisant le mode de calcul précédent) sera nulle. Pour résoudre ce problème, on utilise généralement une interpolation linéaire des fréquences d'apparition des trigrammes, bigrammes et unigrammes [26]. Ainsi, si la probabilité d'un N-gramme n'est pas bien estimée (nombre d'observations inférieur à un seuil fixé), la probabilité s'exprime sous forme d'une somme pondérée des fréquences d'apparition des (N-k)-grammes (k=1,..,N-1).

Une fois ces probabilités calculées, il faut les introduire dans le décodage. La contribution du modèle de langage est le facteur $P(M)$ dans l'équation de Bayes (2.13) au début de la présentation des modèles HMM. Plus le modèle de langage est compliqué (par exemple des trigrammes ou plus),

plus il est difficile d'introduire les probabilités des séquences de mots pour des systèmes de reconnaissance de la parole continue "grand vocabulaire", particulièrement ceux utilisant des algorithmes du type Viterbi synchrone (voir chapitre 6). L'introduction de cette probabilité $P(M)$ dans la récurrence avant (voir équations (2.17) (2.19) et (2.21)) ou dans le décodage Viterbi (voir équations (2.24) ou (2.25)) pose de nombreux problèmes pratiques (si on ne modifie pas les algorithmes utilisés). En général, l'introduction de modèles de langage d'ordre élevé dans les algorithmes de reconnaissance se fait (outre le fait d'augmenter considérablement la complexité des algorithmes) :

- soit en utilisant des algorithmes de décodage du type A* (voir paragraphe 4.4) où il est possible d'introduire ce score du modèle de langage directement dans le calcul du score pour chaque hypothèse (comprenant la liste des mots reconnus).
- soit en utilisant des techniques de générations des N meilleures séquences de mots dans une première passe[27], puis en recalculant les scores pour chaque hypothèse avec le modèle d'ordre élevé. Nous avons d'ailleurs appliqué cette technique dans le chapitre 5 où nous avons réévalué les 200 meilleures phrases des données de test du corpus BREF-80 (générées avec un modèle bigramme ou trigramme) avec un modèle de langage 5-gramme.

2.5 Conclusion

Les modèles de Markov cachés (HMMs) sont les plus utilisés en reconnaissance de la parole continue. Ils bénéficient d'algorithmes d'entraînement et de décodage performants. Néanmoins, les hypothèses nécessaires à la mise en oeuvre de ces algorithmes peuvent pénaliser les performances de ces modèles.

Les principales hypothèses les plus contraignantes sont :

- Entraînement non discriminant (maximisation de la vraisemblance au lieu des probabilités a posteriori).
- Forme des densités de probabilité fixée (multi-gaussiennes ou discrète).
- Les composantes des vecteurs acoustiques sont supposées non corrélées.
- La séquence des états est un processus de Markov du premier ordre.

27. Avec un modèle de langage du type bigramme par exemple.

- Pas de contexte acoustique pris en compte[28]. Aucune corrélation entre les vecteurs acoustiques n'est directement modélisable.

- Le formalisme est rigide, l'intégration d'autres sources de connaissance (syntaxe, sémantique ...) est difficile.

Notons tout de même que la plupart des systèmes de reconnaissance proposés sur le marché actuellement sont basés sur ce type de technique [5, 8, 101, 120][29].

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.

28. Une solution à ce problème proposée par Furui [42] puis par Lee [67] 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.

29. Watson d'AT&T, VoiceType d'IBM et Easy Speaking de Dragon Dictate...

Chapitre 3

Le modèle hybride HMM+MLP

3.1 Introduction

Ces 20 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 [98] a montré les multiples possibilités des réseaux de neurones à couches multiples. Ils sont en particulier utilisés comme estimateur statistique. Il existe de nombreux types de réseaux de neurones utilisés en reconnaissance de la parole [69, 90, 108, 117]:

– les réseaux de neurones multi-couches perceptrons (MLP : Multi-Layer Perceptron). De nombreuses publications ont montré l'utilité de ce type de réseau en reconnaissance automatique de la parole [15, 17, 19, 91]. Nous n'utiliserons que ce type de réseau dans les expériences décrites dans ce travail.

– les réseaux récurrents (Recurrent Neural Network [95]). Robinson et d'autres ont montré dans de nombreuses publications que ce type de réseau pouvait être utilisé efficacement en reconnaissance de la parole [63, 95, 96, 97]. En outre, un système de démonstration *ABBOT* basé sur les modèles hybrides utilise ce type de réseau pour estimer les probabilités d'émission [53, 54, 61] et effectuer une reconnaissance de la parole continue en anglais pour un vocabulaire de 20000 mots.

– les réseaux de neurones à délais temporels (TDNN : Time Delay Neural Network [66, 116]). Ce type de réseau introduit un délai temporel visant d'une part à apprendre la structure temporelle des événements acoustiques et les relations entre ces événements et d'autre part à rendre ce réseau insensible à de possibles erreurs d'alignement (ne reposant donc pas sur une segmentation précise de l'entrée). Les résultats publiés avec ce type de réseau ne portent que sur de petites tâches telles que la reconnaissance de phonèmes particuliers pour quelques locuteurs.

– les réseaux prédictifs (Predictive Neural Network) [40, 114]. Ces réseaux sont très peu utilisés en reconnaissance automatique de la parole, pourtant ils ne sont pas à écarter complètement.

Ces réseaux ont été appliqués avec succès à une multitude de problèmes de classification. Les réseaux de neurones les plus utilisés en reconnaissance de la parole sont les perceptrons multi-couches [73, 74]. Une seule couche cachée est généralement utilisée [1]. L'addition de cette couche cachée permet

1. 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 [118].

au réseau de modéliser des fonctions de décision complexes et non linéaires entre n'importe quel espace d'entrée et de sortie. Nous allons décrire dans ce chapitre comment ils peuvent être utilisés en reconnaissance de la parole.

3.2 Réseaux de neurones et modèles hybrides

La figure 3.1 représente un réseau de neurones perceptrons à une couche cachée. Nous utiliserons ce type de réseau pour développer la théorie associée à ces nouveaux modèles (gardant en mémoire le fait que les autres réseaux cités au début du chapitre peuvent aussi être utilisés).

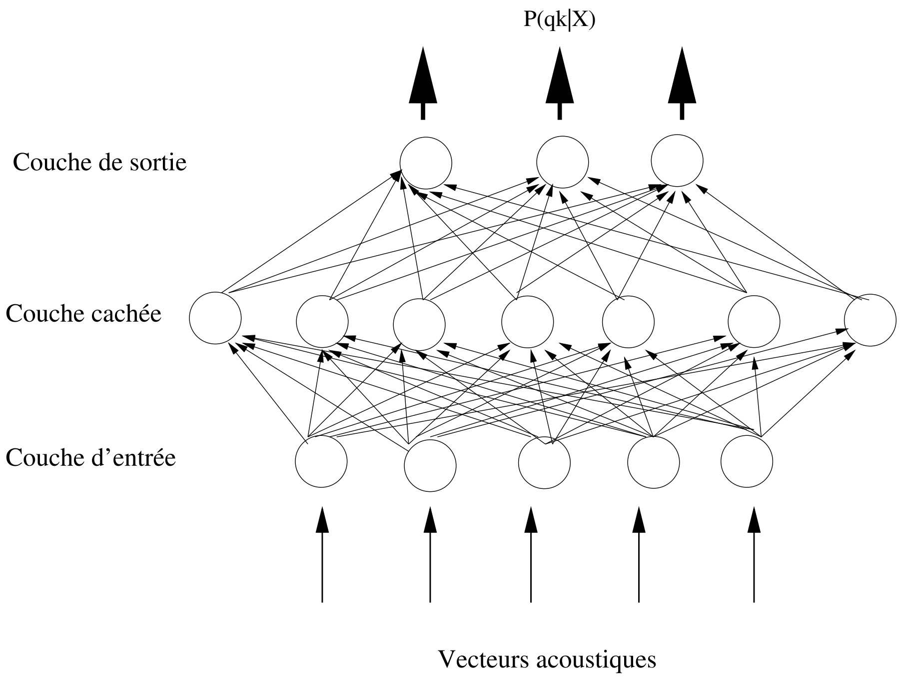

FIG. 3.1 - *MLP à une couche cachée.*

Ils sont généralement constitués d'unités élémentaires interconnectées par des liens avec des poids variables. Le réseau présenté en figure 3.1 est constitué d'unités élémentaires appelées neurones. Le neurone calcule une somme pondérée des composantes du vecteur d'entrée et ajoute un biais à cette somme. Ensuite une fonction non-linéaire est appliquée au résultat. L'architecture du neurone peut donc être résumée simplement sur la figure 3.2. Plusieurs types de fonctions non-linéaires peuvent être définies :

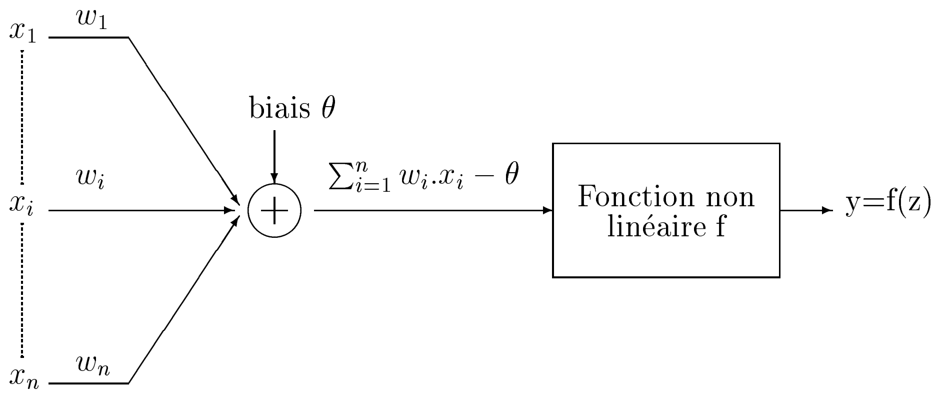

FIG. 3.2 - *Le neurone.*

– fonction softmax :

$$f(z_i) = \frac{e^{z_i}}{\sum_{k=1}^{n} e^{z_k}} \tag{3.1}$$

– fonction signe :

$$f(z) = \{ \begin{array}{ll} +1 & si\ z > 0 \\ -1 & si\ z < 0 \end{array} \tag{3.2}$$

– fonction sigmoïde :

$$f(z) = \frac{1}{1 + e^{z}} \tag{3.3}$$

La fonction non linéaire la plus utilisée est la sigmoïde. Les différents termes de la somme pondérée des noeuds du réseau constituent les paramètres (ou poids) de celui-ci. Ils sont estimés lors d'un entraînement dit *supervisé*. Un entraînement est dit *supervisé* lorsque les entrées sont présentées au réseau les unes après les autres avec la sortie imposée correspondant à chaque entrée. Le problème de l'entraînement d'un réseau revient à trouver un ensemble de paramètres (poids) W qui minimise une fonction d'erreur E. Il existe plusieurs critères d'erreur qui peuvent être utilisés. Notons T l'ensemble des données d'entraînement, les sorties réelles y et désirées ϵ. Les critères les plus connus sont :

– le critère des moindres carrés :

$$E = \sum_{t \in T} \frac{1}{2} . \sum_{l=1}^{L} (y_l^t - \epsilon_l^t)^2 \tag{3.4}$$

– le critère entropique :

$$E = \sum_{t \in T} \sum_{l=1}^{L} \epsilon_l^t ln(y_l^t) \tag{3.5}$$

Les techniques du type descente de gradient sont les plus adaptées pour estimer ces paramètres (les poids du réseau W). Elles sont basées sur le calcul du gradient de l'erreur par rapport aux paramètres du réseau ($\frac{\delta E}{\delta w_{ij}}$) et l'ajustement de ces paramètres jusqu'à ce qu'un minimum de l'erreur soit atteint.

La méthode de "back-propagation" (rétro-propagation) [99], minimisant un des critères d'erreur (erreur quadratique ou entropie) entre les sorties réelles et désirées pour chacun des vecteurs d'entrée, est utilisée pour modifier la valeur des poids. La fonction d'erreur (pour un réseau à plus d'une couche cachée) est une fonction non linéaire des poids avec plusieurs minima possibles. Le gradient de l'erreur est employé pour ajuster les poids de l'élément de sortie et est propagé pour modifier les poids des couches cachées (d'où le nom de rétro-propagation de la couche de sortie vers la couche cachée). L'ajustement des paramètres se fait généralement selon la formule :

$$W^{T+1} = W^T - \eta \frac{\delta E}{\delta w_{ij}} \bigg|_{W^T} \tag{3.6}$$

où η est appelé taux d'apprentissage et doit être assez faible pour garantir la convergence du processus et assez grand pour éviter une convergence trop lente [2].

Cet algorithme est le plus utilisé de nos jours pour l'entraînement des MLPs. Ces réseaux peuvent être utilisés comme classificateur statistique. Les vecteurs acoustiques présentés à l'entrée du MLP sont classés en K classes correspondant aux états stationnaires des modèles de Markov cachés. Lorsque les réseaux de neurones sont entraînés à minimiser un critère des moindres carrés ou d'entropie [3], les valeurs optimales obtenues en sortie peuvent être interprétées en termes de probabilité a posteriori des classes de sorties conditionnées par les entrées. Ainsi il est possible de démontrer que la sortie correspondant à l'état q_k est une estimation de la probabilité locale $p(q_k|x_n)$ si le vecteur x_n est présenté à l'entrée du réseau (voir Annexe C pour une démonstration) [4] [93]. Or en appliquant la loi de Bayes il vient :

2. Il existe d'autres méthodes itératives telles que les méthodes du second ordre ou celles utilisant un moment...

3. Ce sont les fonctions coûts les plus utilisées.

4. Rappelons que les réseaux de neurones récurrents peuvent aussi estimer cette probabilité [94].

$$p(q_k|x_n) = \frac{p(x_n|q_k).p(q_k)}{p(x_n)} \tag{3.7}$$

En divisant la probabilité locale fournie par le réseau $p(q_k|x_n)$ par la probabilité a priori $p(q_k)$ [5], il est possible d'obtenir la vraisemblance $\frac{p(x_n|q_k)}{p(x_n)}$. Lors de la phase de reconnaissance, le coefficient $p(x_n)$ étant constant, nous possédons donc une estimation de la "fameuse" probabilité $p(x_n|q_k)$ à la base de l'équation des HMMs (notamment des récurrences α et β de l'algorithme Baum-Welch et des récurrences Viterbi permettant le calcul de $P(X|M)$). Il n'est donc plus nécessaire d'effectuer, comme pour les HMMs classiques, d'hypothèses restrictives sur la distribution des densités de probabilités.

Le terme de modèle hybride est couramment employé de nos jours pour nommer des systèmes qui tentent d'allier les avantages de deux méthodes différentes. Nous combinons ici les avantages des modèles de Markov (alignement temporel statistique) et des réseaux de neurones (estimation de la probabilité d'émission d'une observation par un état et entraînement discriminant au niveau de la trame). Ces modèles hybrides HMM/ANN sont de plus en plus utilisés en reconnaissance de la parole.

3.3 Pourquoi utiliser les réseaux de neurones ?

Les réseaux de neurones possèdent de nombreux avantages les rendant fortement attractifs pour la reconnaissance automatique de la parole.

1. Ils effectuent un entraînement discriminant local au niveau de la fenêtre d'analyse entre les états du HMM représentés par les sorties du MLP. L'entraînement Viterbi classique des HMMs n'est pas du tout discriminant car un critère de maximum de vraisemblance est maximisé (Voir chapitre 2) ...

2. Il n'est plus nécessaire d'effectuer des hypothèses contraignantes à propos de la distribution des paramètres d'entrée. En effet, la probabilité $p(q_k|x_n)$ est estimée directement par le réseau.

3. Ils possèdent une architecture flexible.

4. Les réseaux de neurones sont typiquement des structures régulières et parallèles, les rendant facilement utilisables du point de vue "hardware".

5. Facilement calculable à partir de l'alignement Viterbi.

5. Ils offrent la possibilité de tenir compte de la corrélation entre les vecteurs acoustiques[6]. Il est en effet possible de fournir un contexte droit et gauche à l'entrée du réseau en introduisant par exemple une fenêtre temporelle du type $X_{n-c}^{n+d} = \{x_{n-c}, ..., x_n, ..., x_{n+d}\}$. Le réseau estimera alors des probabilités du type $p(q_k|X_{n-c}^{n+d})$ pouvant être également utilisées dans les équations HMMs.

Cette approche s'est révélée dernièrement des plus efficaces en particulier pour des tâches complexes telles que la reconnaissance de la parole continue, grand vocabulaire, indépendamment du locuteur, avec un nombre de paramètres beaucoup moins élevé que les HMMs classiques [13, 31].

3.4 Inconvénients de la méthode

Il existe quand même quelques inconvénients :

- L'entraînement du réseau est un processus compliqué. Il arrive souvent que le réseau subisse un surentraînement. En effet, si le réseau possède trop de degrés de liberté par rapport à la complexité du problème, il aura tendance à se spécialiser uniquement sur les exemples qui lui sont présentés perdant ainsi son caractère de généralisation. Pour pallier ce problème, une partie de la base d'entraînement est généralement utilisée pour tester le réseau. Dans le cas où de faibles variations sont observées d'une itération à l'autre (révélant que les paramètres du réseau se sont trop spécialisés sur les données d'entraînement), l'entraînement est stoppé[7] [91]. Le taux de classification sur une partie de cross-validation de la base de données est donc utilisé pour contrôler le taux d'apprentissage et pour décider quand il faut arrêter l'entraînement. Le taux d'apprentissage est maintenu constant jusqu'à ce que le taux de classification augmente de moins de 0.5% après quoi, il est réduit d'un facteur $\frac{1}{2^n}$[8] jusqu'à ce que les performances ne varient plus.

- Le type de modèle le plus adapté est le modèle de phonème. Le nombre d'états correspondant aux modèles dépendants du contexte ou de mots est beaucoup trop élevé pour que ceux-ci soient relativement bien entraînés. C'est pourquoi la plupart des outils actuels basés sur ce type de technologie utilise des modèles de Markov de phonèmes. Néanmoins

6. Plus précisément que les HMMs avec les dérivées...

7. C'est une méthode appelée couramment, dans le jargon de la reconnaissance de la parole, *Cross-Validation* ou Validation croisée.

8. n étant le nombre d'itérations effectuées.

de récentes études ont montré qu'il était aussi possible d'utiliser des modèles dépendants du contexte [18, 25, 35, 103] avec les modèles hybrides. Nous développerons d'ailleurs dans le chapitre 8 quelques méthodes efficaces permettant l'utilisation de modèles hybrides et de modèles dépendants du contexte.

- Comme pour les HMMs à distributions multi-gaussiennes et discrets, l'entraînement des réseaux est supervisé (les sorties désirées sont fournies au MLP). Il faut donc une première segmentation pour initialiser le processus d'entraînement. Cette segmentation est généralement obtenue par l'intermédiaire d'un autre système (par exemple un HMM discret ou continu). Il est aussi possible d'utiliser une segmentation linéaire, mais dans ce cas, pour les MLPs, il peut y avoir des problèmes de convergence. Malfrère et Dutoit [71] ont montré qu'il était possible d'utiliser d'autres méthodes (basées sur un synthétiseur vocal de bonne qualité) pour générer une première segmentation lorsqu'on ne dispose pas de modèles HMM pré-entraînés. Nous avons utilisé avec succès cette méthode pour l'entraînement d'un système de reconnaissance en français (Voir paragraphe 6.4). Il va de soi que meilleure sera la segmentation initiale, plus vite le réseau convergera vers une solution optimale.

- Le principal problème associé à l'utilisation des réseaux de neurones en reconnaissance de la parole, est lié à la phase d'entraînement. En effet, cette phase est beaucoup plus longue que celle correspondant aux méthodes discrètes et continues. Ceci est principalement dû à l'algorithme de rétro-propagation du gradient utilisé lors de l'entraînement sur de très gros réseaux de neurones. Il existe des méthodes permettant une réduction significative du temps nécessaire à l'entraînement d'un réseau (par exemple la suppression de trames acoustiques, voir chapitre 7).

Ces quelques inconvénients peuvent être contournés mais constituent une limitation de ces modèles. Il ne faut pas oublier que seules quelques hypothèses des modèles de Markov cachés ont été supprimées. C'est néanmoins un premier pas vers un système plus robuste. L'amélioration du taux de reconnaissance généralement observée avec ce type de modèles, justifie pleinement leur emploi.

3.5 Comparaison des différents modèles

Nous venons de voir dans ce chapitre et le précédent les principales méthodes utilisées en reconnaissance de la parole. Nous avons mis en évidence les avantages et défauts respectifs de chacune d'entre elles. Nous allons maintenant comparer ces méthodes sur plusieurs bases de données.

3.5.1 Expériences en reconnaissance de mots isolés

La reconnaissance des chiffres a toujours été un sujet particulier d'attention en reconnaissance de la parole. De nombreuses expériences ont déjà été réalisées par divers laboratoires dans le monde. L'objectif de cette étude est la comparaison des différents types de modèles définis précédemment pour un processus de reconnaissance de chiffres isolés et enchaînés. Pour cela nous avons choisi d'effectuer les expériences sur la base de données BDSONS (Voir Annexe D).

Les vecteurs acoustiques utilisés dans cette expérience sont les coefficients log RASTA-PLP [52] (voir paragraphe 2.3.3). 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 chapitre, les trois types de modèles ont été utilisés avec les caractéristiques suivantes (reconnaissance des chiffres isolés et connectés).

- *Modèle 1*: **HMM discret**. Les vecteurs acoustiques ont été quantifiés en 5 dictionnaires

  - 512 pour les coefficients log RASTA-PLP,
  - 512 pour les Δlog RASTA-PLP,
  - 256 pour les $\Delta\Delta$log RASTA-PLP,
  - 32 pour la dérivée première de l'énergie Δénergie,
  - 32 pour la dérivée seconde de l'énergie $\Delta\Delta$énergie.

  Des modèles de mots à 10 états ont été utilisés pour modéliser chacune des unités élémentaires (les chiffres : 10 mots). Ainsi le nombre de paramètres associés à ce type de modèle est :

$$\begin{cases} \text{nb-paramètres} = \text{nb-centroïde.nb-total-états} \\ (512+512+256+32+32).10.10 = 134400 \end{cases}$$

- *Modèle 2*: **Un modèle HMM continu** de mots à 10 états a été utilisé. Chaque état est modélisé par une distribution multi-gaussiennes à

16 gaussiennes par état. La matrice de covariance est supposée diagonale et comme pour les modèles discrets, les vecteurs acoustiques ainsi que leurs dérivées premières et secondes ont été utilisés. Le nombre de paramètres associés à ce type de modèle est donc [9]:

$$\begin{cases} \text{nb-paramètres} = \text{nb-total-états.nb-gaussiennes.(nb-coeff.2+1)} \\ 10.10.16.[(12+12+1+12+1).2+1] = 123200 \end{cases}$$

- *Modèle 3*: **Le modèle hybride**. Un réseau de neurones MLP possédant en entrée 234 neurones correspondant à 9 trames de contexte (plp, Δplp, Δénergie, $\Delta\Delta$énergie), une couche cachée de taille variable, une couche de sortie composée d'autant de neurones qu'il y a d'états HMM. Des modèles de Markov cachés de phonèmes à 3 états ont été utilisés pour les expériences reportées dans ce chapitre. Sachant que les dix chiffres peuvent être transcrits avec 22 phonèmes différents, le réseau possède donc 66 sorties. Le nombre de neurones de la couche cachée a été choisi de manière à obtenir un nombre de paramètres sensiblement égal aux deux autres modèles. Ainsi un réseau MLP à une couche cachée comprenant 234 noeuds à l'entrée, 450 noeuds pour la couche cachée et 66 noeuds de sorties a été entraîné. Le nombre de paramètres associés à ce modèle est :

$$\begin{cases} \text{nb-paramètres} = \text{(nb-entrées+1).nb-cachées+nb-sorties.(nb-cachées+1)} \\ (234+1).450+66.(450+1) = 135500 \end{cases}$$

Ces trois types de modèles ont été comparés dans le cadre d'une reconnaissance de chiffres isolés puis connectés. Les 8600 phrases de la partie entraînement de la base de données ont été utilisées pour entraîner les modèles [10]. La partie test est composée de 2300 chiffres isolés énoncés par des locuteurs différents de ceux qui ont servi à l'entraînement. Le tableau 3.1 résume les différents résultats obtenus.

9. La matrice de covariance étant supposée diagonale, le nombre de paramètres par gaussienne est donc nb-coeff (pour la matrice de covariance) + nb-coeff (pour le vecteur moyenne) + 1 (pour le coefficient de pondération).

10. Les 8600 phrases ont été divisées en 8000 pour l'entraînement et 600 pour la cross-validation pour le modèle hybride.

Type du modèle	Taux d'erreur
1	0.8 %
2	0.3 %
3	0.4 %

TAB. 3.1 - *Taux d'erreur pour une reconnaissance de chiffres isolés pour les trois types de modèles. Coefficients log RASTA-PLP. Test composé de 2300 chiffres isolés.*

3.5.2 Expériences en reconnaissance de chiffres connectés

Les mêmes conditions que dans l'expérience précédente ont été gardées (mêmes locuteurs pour l'entraînement et pour le test). Les modèles ont été entraînés sur environ 20000 séquences de trois, quatre et cinq chiffres. La partie test est composée de 2770 séquences de trois quatre et cinq chiffres. Les résultats obtenus sont résumés dans le tableau 3.2.

Type du modèle	Ins.	Sup.	Subs.	Taux d'erreur	Tps
1	1.0 %	1.1 %	1.4 %	3.5 %	1429 s
2	0.8 %	0.9 %	1.2 %	2.9 %	18100 s
3	0.8 %	0.7 %	0.7 %	2.2 %	3500 s

TAB. 3.2 - *Taux d'erreur pour une reconnaissance de chiffres connectés pour les trois types de modèles. Coefficients log RASTA-PLP. Le temps d'exécution associé à chacun des modèles a été affiché (SUN SPARC Station, OS: Solaris 2.5.6). 2770 phrases de test composées de séquences de trois, quatre et cinq chiffres enchaînés.*

3.5.3 Expériences en reconnaissance de la parole continue "vocabulaire moyen"

Nous comparons dans ce paragraphe les systèmes à distributions multi-gaussiennes et les modèles hybrides sur une tâche un peu plus compliquée que la reconnaissance de chiffres. Pour cela, nous avons sélectionné la base de données RESOURCE MANAGEMENT[11] (Voir Annexe D). L'entraînement des deux modèles a été effectué sur les 3990 phrases utilisées pour les campagnes de tests officielles[12]. Le fichier test est composé des 300 phrases correspondant

11. Noté RM dans le reste du document.
12. 3591 phrases pour l'entraînement et 399 pour la validation croisée

à la campagne d'enregistrement "feb89". Nous avons utilisé 68 phonèmes pour décrire l'ensemble des 1000 mots du dictionnaire. Les deux types de modèles ont été entraînés à partir d'une segmentation obtenue à l'aide d'un réseau déjà entraîné sur le corpus TIMIT [122]. Le modèle de langage est une grammaire "wordpair" (fournie avec le corpus) possédant une perplexité de 60. Les résultats obtenus sur ce corpus pour les deux types de modèles sont visibles dans le tableau 3.3.

	Ins.	Sup.	Subs.	Taux d'erreur
HMM-MLP	0.6%	1.2%	3.6%	5.4%
HMM-Gauss	2.9%	1.6%	7.9%	12.4%

TAB. 3.3 - *Taux d'erreur pour une reconnaissance de la parole continue sur RM (feb89), modèles hybrides et multi-gaussiens ,coefficients PLP, grammaire "wordpair" de perplexité 60.*

Pour le modèle continu, chaque état est modélisé par une distribution multi-gaussienne à 16 gaussiennes par état. La matrice de covariance est diagonale[13] et les coefficients cepstraux ainsi que les dérivées premières et secondes de ces coefficients complétés par les dérivées première et seconde de l'énergie sont utilisées. Le MLP possède 9 trames de contexte acoustique et le vecteur d'entrée possède 26 composantes (plp, Δplp, Δénergie et $\Delta\Delta$énergie). La couche cachée est composée de 1000 noeuds. Les taux d'erreur du tableau 3.3 sont assez semblables à ceux obtenus par d'autres laboratoires [24]. Là encore nous observons que le modèle hybride est meilleur que la modélisation multi-gaussienne. Néanmoins, il faut rester conscient que de nombreux laboratoires ont publié de meilleurs résultats avec la modélisation HMM à distributions multigaussiennes sur ce corpus en utilisant des modèles dépendants du contexte et des modèles de transitions entre mots ("crossword"). Nous verrons d'ailleurs un peu plus loin qu'il est possible d'obtenir des performances identiques aux systèmes CD multi-gaussien et même meilleures en utilisant des modèles hybrides plus compliqués (voir chapitres 5 et 8).

3.6 Conclusion

La technologie mise en oeuvre, en vue d'une reconnaissance de chiffres isolés ou de séquences de chiffres connectés basée sur les modèles statistiques

13. On néglige la corrélation entre les composantes des vecteurs acoustiques.

(HMMs et modèles hybrides), fonctionne très bien. Des taux de reconnaissance de l'ordre de 99% ont été observés pour des locuteurs n'ayant pas participé à l'entraînement pour une reconnaissance de chiffres isolés. De même, pour un vocabulaire de 1000 mots et de la parole continue, un taux d'erreur de 5.4% a été obtenu avec le système hybride. Du point de vue de l'efficacité des modèles, il semble évident (voir les tableaux 3.1, 3.2 et 3.3) que les modèles hybrides sont au moins aussi performants (dans le cas d'une reconnaissance de chiffres isolés) voire même supérieurs (dans le cas reconnaissance de chiffres connectés ou de la parole continue vocabulaire de taille moyenne) que les HMMs discrets et continus. De plus, du point de vue du temps d'exécution, les modèles hybrides sont un peu plus lents que les modèles discrets mais beaucoup plus rapides que les modèles continus. Les meilleurs résultats publiés dans la plupart des comparaisons sur des corpus précis sont le plus souvent obtenus avec les modèles hybrides [14]. De nombreuses publications ont montré l'avantage de combiner les réseaux de neurones et les modèles de Markov cachés en reconnaissance de la parole (isolée ou continue) indépendamment du locuteur pour de petits ou de grands vocabulaires. Cette même conclusion a pu être effectuée dans le cas particulier des corpus BDSONS et RM. Les avantages dans l'utilisation cumulée des modèles de Markov cachés et des réseaux de neurones se reflètent donc bien dans les résultats obtenus sur ces corpus particuliers. Le modèle hybride entraîné sur le corpus BDSONS est actuellement utilisé pour la démonstration de reconnaissance de chiffres connectés en français du laboratoire TCTS [15]. Les capacités de cet outil à reconnaître n'importe quelle séquence de chiffres énoncée par n'importe quel locuteur sont très impressionnantes.

14. Sans compter que le nombre de paramètres utilisés est le plus souvent largement inférieur à celui des meilleurs systèmes HMM continus.

15. Laboratoire de Théorie des Circuits et Traitement du Signal.

Chapitre 4

N meilleures séquences de mots

4.1 Introduction

En reconnaissance de la parole continue, la meilleure phrase reconnue par les systèmes de reconnaissance usuels ne coïncide pas toujours effectivement avec la phrase prononcée. De récents travaux ont montré l'utilité de générer une liste des N meilleures séquences de mots reconnues afin de trouver, avec l'aide d'informations supplémentaires, la phrase effectivement énoncée (par exemple la phrase satisfaisant à des restrictions supplémentaires ou encore la meilleure phrase après une réévaluation des scores respectifs de chaque élément de la liste).

De nombreux algorithmes sont utilisés pour générer la phrase qui a la plus grande probabilité d'avoir été émise :

- Les algorithmes du type Viterbi synchrone : Bridle [20].
- Le Level Building : Rabiner et Myers [76].
- Le Two Level : Sakoe [102].

Les premières méthodes utilisées pour générer les N meilleures séquences de mots sont apparues vers la fin des années 80 et le début des années 90. Chacune de ces méthodes possède un critère de validité : *l'optimalité.*

Une méthode est optimale si les N meilleures solutions fournies sont les N meilleures effectives, c'est à dire s'il n'y a pas d'oubli.

Les N meilleures séquences obtenues doivent être syntaxiquement différentes et non pas des alignements différents d'une même solution. Ceci constitue une contrainte supplémentaire pour éviter d'obtenir en sortie du processus les mêmes séquences de mots avec des scores différents correspondant à des segmentations différentes.

Il existe plusieurs types de méthodes pour effectuer une recherche des N meilleures séquences de mots en reconnaissance de la parole :

- les méthodes dérivées de la programmation dynamique [72, 106, 107, 112].
- les méthodes basées sur l'algorithme A* [55, 79, 81, 111].

Nous allons étudier ces différentes méthodes en mettant en exergue leurs avantages et inconvénients respectifs. Le processus de reconnaissance de la

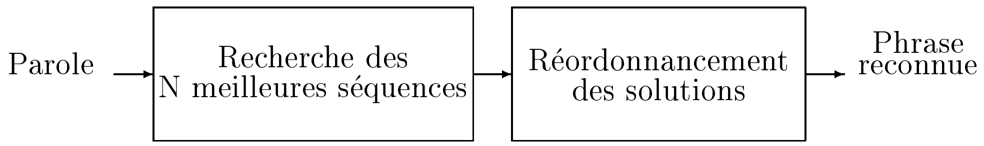

FIG. 4.1 - *Principe de base de la reconnaissance de la parole utilisant la recherche des N meilleures séquences de mots.*

parole utilisant les méthodes de recherche des N meilleures séquences de mots peut être visualisé sur la figure 4.1.

Le processus de reconnaissance est donc séparé en deux parties :

- La recherche des N meilleures solutions. La sortie de ce processus est une liste contenant les N meilleures séquences de mots.
- Le calcul des nouveaux scores pour ces hypothèses est réalisé dans une seconde partie.

La séparation du processus de reconnaissance en deux parties est indispensable dans certains cas. En effet, pour un vocabulaire important, le nombre de combinaisons possibles formant les phrases est très important. Ainsi, l'utilisation directe de modèles de langage (N-grammes par exemple avec $\mathrm{N} > 3$) demande un effort de calcul très important par rapport à l'amélioration du taux de reconnaissance qui en découle. La liste obtenue par ces méthodes de recherche des N meilleures séquences de mots peut être facilement réestimée par une grammaire N-gramme avec un coût de calcul relativement faible. Ainsi dans le cas particulier d'une grammaire quadrigramme, il apparaît beaucoup plus efficace d'utiliser cette grammaire sur un petit nombre d'alternatives plutôt que sur toutes les phrases.

Il existe bien d'autres exemples mettant en exergue l'utilité de ces méthodes pour des vocabulaires plus restreints. Prenons le cas d'un système de reconnaissance des numéros de cartes de crédits (16 chiffres). Sur les 10^{16} combinaisons possibles, il n'y a peut-être que quelques millions de combinaisons autorisées. Il va de soi qu'une telle contrainte peut aider le processus de reconnaissance. Incorporer les combinaisons valables dans la grammaire coûterait cher en temps de calcul et mémoire.

Une vérification de la validité des N meilleures séquences de mots reconnues peut être effectuée rapidement et permettre une amélioration du processus de reconnaissance.[1]

En plus des avantages en coût de mémoire et en temps de calcul, il y a d'autres raisons justifiant l'utilisation de ces méthodes.

1. La sortie de la première partie du processus étant une liste de phrases (donc du texte), l'interface fournissant la phrase reconnue est facilement réalisable.

2. La séparation du processus de reconnaissance en deux parties permet d'insérer facilement un autre traitement possible de la sortie de la première phase. De plus, les deux parties peuvent être optimisées et programmées de manière indépendante.

Dans ce chapitre, nous allons nous focaliser sur les méthodes de recherche des N meilleures séquences de mots en reconnaissance de la parole. Nous mettrons en évidence les coûts en mémoire et en temps nécessaires à chacun des algorithmes.

4.2 Algorithme Viterbi synchrone

De nombreuses méthodes de recherche des N meilleures séquences de mots sont inspirées des algorithmes de programmation dynamique couramment utilisés en reconnaissance de la parole. Ainsi, nous décrirons dans un premier temps l'algorithme de programmation dynamique le plus employé en reconnaissance de la parole continue : l' *algorithme de programmation dynamique en une passe* ou "One Pass" [20, 77, 78]. Nous développerons ensuite les méthodes basées sur cet algorithme pour générer les N meilleures séquences de mots [107, 112].

4.2.1 Algorithme de base

Nous décrirons ici l'algorithme de base de programmation dynamique utilisé en reconnaissance de la parole continue; mais il ne faut pas oublier que de nombreuses adaptations de cet algorithme ont du être réalisées pour la mise en oeuvre d'algorithmes capables de reconnaître la parole continue pour un grand vocabulaire. Le "One Pass" est l'algorithme du type Viterbi synchrone le plus utilisé de nos jours en reconnaissance de la parole [20].

1. En effet, dans le cas particulier de la reconnaissance de numéro d'identification, il est possible de vérifier la validité des numéros reconnus selon le nombre de chiffres présents dans le numéro, ou même (ce qui est souvent le cas) en effectuant une combinaison des chiffres reconnus.

Trois types d'opérations sont effectuées en même temps et en une seule passe [77].

- Détection des frontières d'unités dans la phrase de test.
- Recherche de la meilleure séquence d'unités de référence.
- Concaténation des unités reconnues.

Ces opérations sont effectuées par l'algorithme au rythme de l'arrivée des vecteurs acoustiques correspondant à la phrase de test (algorithme synchrone) [78]. Soit $X = \{x_1, x_2, ..., x_N\}$ une séquence de vecteurs acoustiques représentant une chaîne de mots à reconnaître. Notons M= $\{M_k; k = 1...K\}$ l'ensemble des unités de référence associées à des modèles (ou à des états) de Markov cachés.

La figure 4.2 représente en abscisse les vecteurs acoustiques et en ordonnée deux indices :

- k pour référencer le modèle de l'unité $M_k(k = 1..K)$
- j pour l'état considéré de ce modèle (j=1..J(k) où J(k) est le nombre d'états du mot k).

Une telle représentation est classique en reconnaissance de la parole et permet une meilleure visualisation du mode opératoire utilisé. Cette grille va nous permettre de définir un graphe représentant l'ensemble des chemins correspondant à un alignement possible entre la séquence de vecteurs acoustiques X et un modèle de référence M.

Cet algorithme de programmation dynamique permet d'obtenir la solution optimale à un problème de minimisation d'un critère d'erreur sans être obligé de considérer toutes les solutions possibles. Rappelons l'équation de récurrence Viterbi vue au chapitre 2.

$$P(q_j^n, X_1^n|M_k) = \max_i \, [P(q_i^{n-1}, X_1^{n-1}|M_k).p(q_j^n|q_i^{n-1}, M_k)].p(x_n|q_j^n, M_k) \tag{4.1}$$

Soit en passant au log :

$$\begin{aligned} -\,log(P(q_j^n, X_1^n|M_k)) \;\; &= \;\; \min_i \, [-log(P(q_i^{n-1}, X_1^{n-1}|M_k)) - log(p(q_j^n|q_i^{n-1}, M_k))] \\ & \qquad -log(p(x_n|q_j^n, M_k)) \end{aligned} \tag{4.2}$$

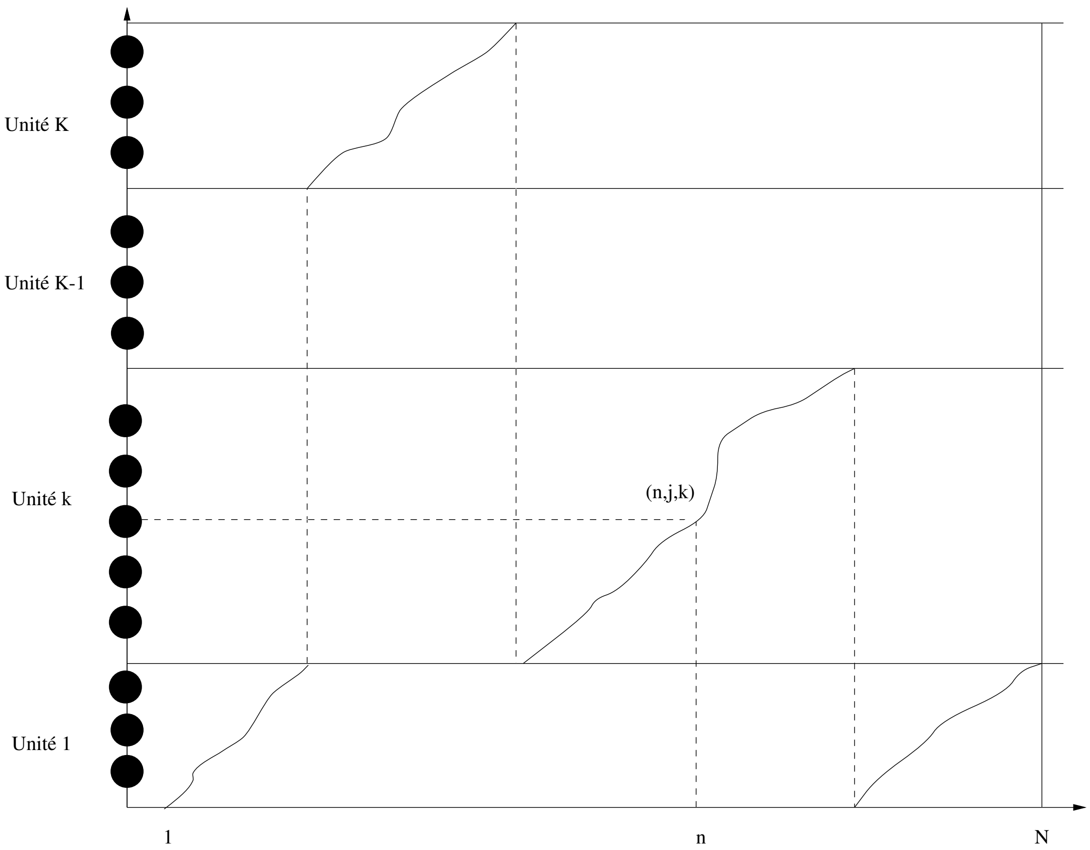

FIG. 4.2 - *Une grille de programmation dynamique.*

En posant :

$$\begin{cases} D(n,j,k) = -log[P(q_j^n, X_1^n|M_k)] \\ \text{distance cumulée à l'instant } n \text{ pour l'état } q_j \text{ du modèle } M_k. \\ d(n,j,k) = -log[p(x_n|q_j^n, M_k)] \\ \text{distance locale pour l'instant } n \text{ et l'état } q_j \text{ du modèle } M_k. \\ p(i,j,k) = -log[p(q_j^n|q_i^{n-1}, M_k)] \\ \text{une pénalité pour le passage de l'état } q_i \text{ à } q_j \text{ du modèle } M_k. \end{cases}$$

Il vient une équation récurrente du type programmation dynamique :

$$D(n,j,k) = d(n,j,k) + \min_i \, [D(n-1,i,k) + p(i,j,k)] \tag{4.3}$$

Le problème de la reconnaissance de la suite de mots dans la séquence X peut être vu comme un problème d'optimisation. Il suffit de trouver le chemin optimal correspondant à la distance cumulée $D(N, J(k), k)$ minimale.

En calculant les distances cumulées de tous les chemins optimaux pour tous les points de la grille, la solution optimale à l'équation (4.3) et la séquence d'unités associées à cette distance cumulée minimale pourront être calculées.

Pour chaque vecteur de test x_n, des règles de deux types sont appliquées à toutes les hypothèses courantes de chemins optimaux contenus dans la colonne n :

- La règle intra-unité : réalise l'alignement des vecteurs de test sur une unité de référence.

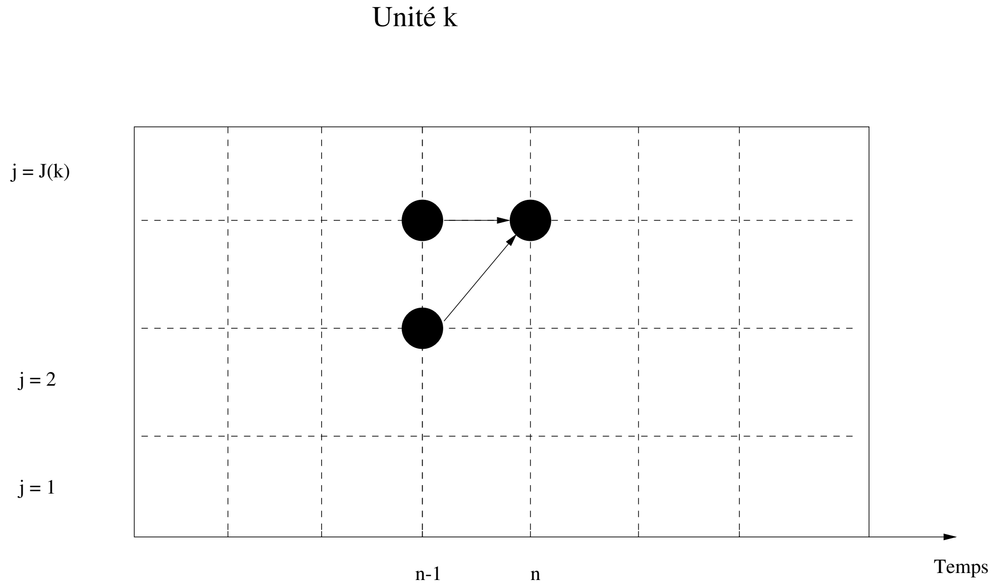

FIG. 4.3 - *Règle intra-unité employée dans la programmation dynamique.*

Cette règle est liée à la topologie du modèle de Markov caché représentant l'unité à reconnaître. En effectuant l'hypothèse du premier ordre (un état ne peut effectuer qu'une transition récurrente vers lui-même ou vers l'état suivant [2]), à l'intérieur d'une unité k, pour calculer le score à l'instant n, seul le score du même état à l'instant n-1 ou celui de l'état précédent au même instant, doivent être pris en compte (voir figure 4.3). D'autres règles peuvent être appliquées selon les modèles qui sont utilisés.

2. Voir hypothèse **H3** paragraphe 2.4.4, pour des HMMs gauche-droite d'ordre 1, $p(q^t|q^{t-1}...q^{t-M}) = p(q^t|q^{t-1})...$

-Règle inter-unités : autorise le passage d'une unité à l'autre au cours de l'alignement.

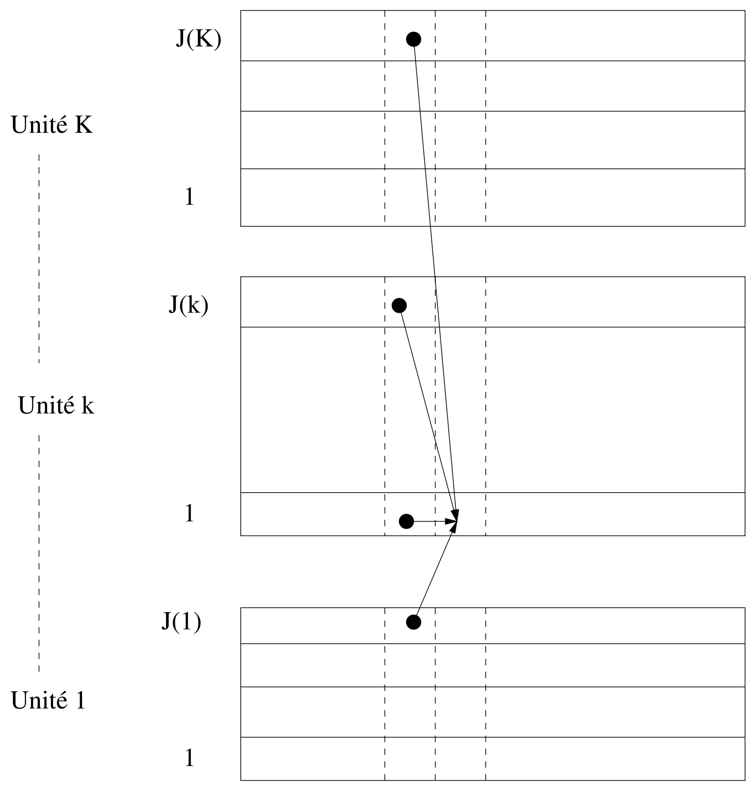

FIG. 4.4 - *Règle inter-unités employée dans la programmation dynamique.*

Pour le passage d'une unité à une autre et le calcul des scores à l'instant n, seuls les scores de toutes les unités dans leurs états finaux (à l'instant n-1) ou le score de l'unité courante dans l'état 1 (à l'instant n-1) sont pris en compte (voir figure 4.4)[3].

L'algorithme s'achève avec le calcul de la dernière trame n=N; la séquence de mots correspondant à la solution optimale est ensuite récupérée par une procédure de recherche arrière (back track). Sachant que pour obtenir la séquence de mots correspondant au chemin optimal, seules les transitions

3. L'introduction d'une grammaire peut limiter le nombre d'unités à considérer. De plus, comme déjà explicité dans le chapitre 2, il est possible d'introduire une pénalité de transition entre les mots pour jouer sur les taux d'insertions et de suppressions de mots. En effet, plus la pénalité introduite est importante et moins il y aura de transitions entre les mots.

entre les mots doivent être gardées. Il suffit, lors de la programmation dynamique, de mémoriser pour chaque instant n, le mot k pour lequel la distance accumulée $D(n, J(k), k)$ est minimale et l'instant où le mot a commencé. A la fin de la programmation dynamique, un simple algorithme de parcours de ces informations permet d'obtenir la séquence de mots optimale.

La simplicité de cet algorithme constitue un énorme avantage sur les autres méthodes (Two Level [102] et Level Building [76]) qui le place comme le plus utilisé en reconnaissance de la parole.

4.2.2 Méthodes d'élagage

La complexité de l'algorithme de recherche de la meilleure solution varie en fonction du nombre de mots présents dans le vocabulaire et de la perplexité du modèle de langage. Dans le cas de "grands vocabulaires" (60000 mots et plus), il est bien entendu nécessaire de limiter l'espace de recherche en effectuant une sélection des chemins. Pour cela des techniques d'élagage sont utilisées visant à limiter l'espace de recherche afin que l'algorithme puisse tourner en temps réel. Généralement deux techniques d'élagage sont employées pour réduire l'espace de recherche lors du décodage. La première technique, très répandue et appelée "beam search", définit un intervalle de variation absolu Δ par rapport au score de l'hypothèse la plus probable à l'instant n[4]. Les hypothèses en dehors de cet intervalle (avec un score cumulé inférieur à $logP_{max}(n) - \Delta$) sont supprimées de l'espace de recherche. On évite ainsi de maintenir des hypothèses qui ont très peu de chances d'être utilisées pour la recherche de la meilleure séquence.

Il est aussi possible de limiter le nombre d'hypothèses étendues pour réduire l'espace de recherche. La technique d'élagage consiste à calculer pour chaque trame, une liste des noeuds actifs (par comparaison avec un seuil des distances cumulées) sur lesquels est réalisée la programmation dynamique. Selon le seuil choisi, il est possible de réduire fortement l'espace de recherche de la meilleure solution et d'accélérer d'autant plus le processus de reconnaissance. Le nombre d'hypothèses à garder (ainsi que la plupart des paramètres des techniques d'élagage comme le Δ) est généralement fixé sur des données de développement.

La seconde technique d'élagage a été développée pour tirer avantage des propriétés des modèles hybrides (estimation des probabilités a posteriori et nature discriminante des MLPs au niveau de la trame). En effet, les sorties des MLPs peuvent être considérées comme une estimation de la présence d'un phonème pour une trame acoustique particulière. D'ailleurs la figure 4.5 per-

4. notée $logP_{max}(n)$

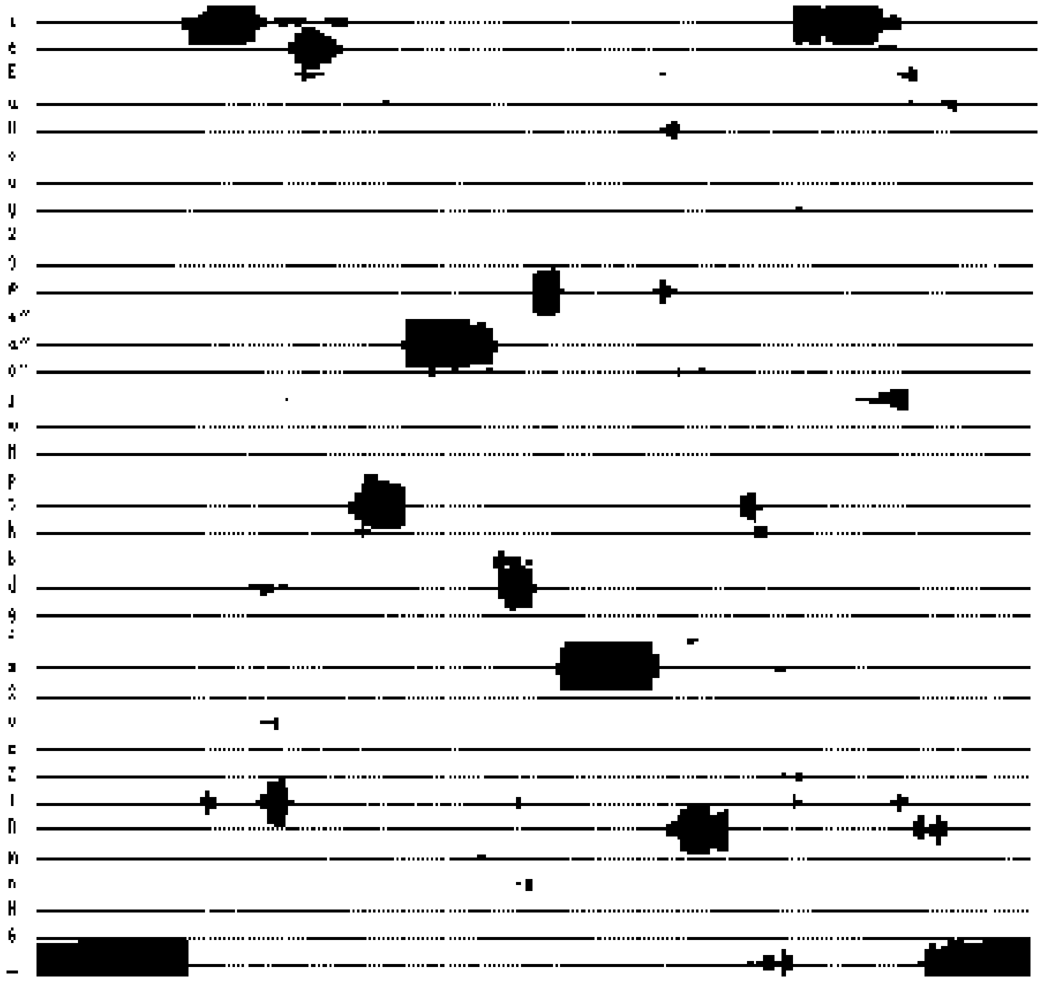

FIG. 4.5 - *La sortie d'un MLP pour la phrase "Il est temps de sortir".*

met d'observer de manière significative l'aspect discriminant des MLPs au niveau de la trame acoustique. A chaque trame acoustique de 10 ms, le MLP fournit un vecteur de 35 probabilités a posteriori. Chacune de ces probabilités correspondant à un des phonèmes utilisés pour notre système de reconnaissance en français (Voir annexe A) constitue l'axe des ordonnées. Il apparaît sur cette figure qu'une seule classe de sortie (donc un seul phonème) a tendance à dominer à chaque instant (les zones sombres correspondent à des probabilités élevées). Même s'il y a conflit, seules quelques classes sont actives en même temps. Cette propriété peut être utilisée pour effectuer un élagage efficace. Dans le cas où la probabilité d'un phonème, étant donnée une séquence de vecteurs acoustiques, est inférieure à un seuil fixé, alors tous les mots contenant ce phonème à cet instant peuvent être supprimés. Ce seuil est arbitrairement fixé à l'aide de données de développement et reste constant pour tous les phonèmes. Cette approche est réellement efficace car elle permet d'accélérer fortement le décodage sans introduire trop d'erreurs supplémentaires. Ce type d'élagage est utilisé par le décodeur "grand vocabulaire" Noway (voir chapitre 6).

4.2.3 Conclusion

Nous avons détaillé dans ce paragraphe l'algorithme de base qui est employé par la plupart des outils de reconnaissance de la parole continue. La complexité de cet algorithme est fonction du nombre de mots présents dans le vocabulaire. Dans le cas de "grands vocabulaires", il est nécessaire de limiter l'espace de recherche en utilisant des techniques d'élagage. Ces techniques permettent d'accélérer fortement le décodage sans introduire trop d'erreurs supplémentaires. Cet algorithme est à la base des méthodes de génération des N meilleures séquences de mots reconnues que nous proposons de décrire dans les paragraphes suivants.

4.3 Méthodes basées sur le One-Pass

De nombreuses méthodes de recherche des N meilleures séquences de mots basées sur des algorithmes de programmation dynamique ont été mises en oeuvre [72, 106, 107, 112]. Nous nous intéresserons tout d'abord à la première tentative proposée par Steinbiss. Ensuite, nous développerons d'autres méthodes plus récentes. Nous mettrons en évidence les défauts relatifs à chacune de ces méthodes et les solutions apportées.

4.3.1 La méthode proposée par Steinbiss

Volker Steinbiss [112] proposa une méthode exacte de recherche des N meilleures solutions à partir de l'algorithme One-Pass. A chaque noeud correspondant à un état HMM q_k et pour chaque instant, on mémorise dans une liste les N meilleurs scores correspondant à des séquences de mots différentes.

La figure 4.6 présente la procédure de recherche des 2 meilleures séquences de mots. L'extension à N ne pose pas de problèmes particuliers.

Au noeud 1 arrivent trois chemins (s_1, s_2, s_3), on choisit les deux meilleurs qui fournissent les scores à mémoriser (s_1 et s_3 par exemple). La même chose est effectuée pour le noeud 2. On récupère ainsi s_4 et s_5. De ces quatre chemins, le noeud 3 va sélectionner les deux meilleurs chemins s_1 et s_4 et ainsi

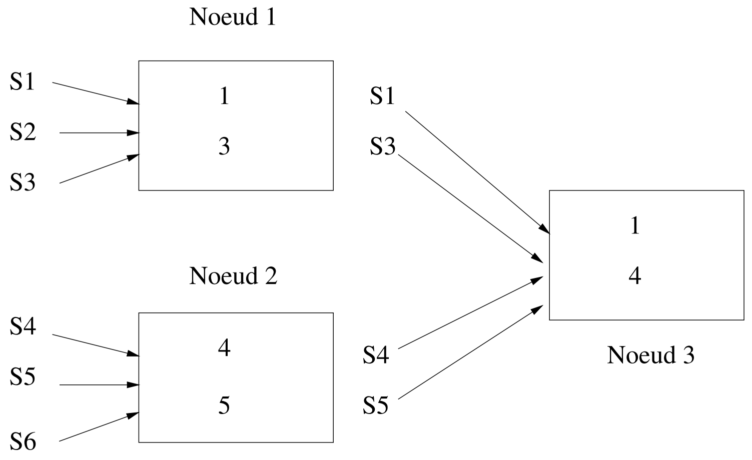

FIG. 4.6 - *Procédure de recherche des deux meilleurs chemins.*

de suite ...

A la fin de la phrase, on récupère facilement les deux meilleurs chemins par "backtracking" pour obtenir les séquences complètes (dans ce cas particulier les séquences correspondant aux chemins s_1 et s_4). Ainsi, à chaque hypothèse et pour chaque état, les N meilleurs chemins finissant sur cet état sont conservés. Le principe de base est donc de conserver à chaque étape de l'algorithme non plus le meilleur chemin mais les N meilleurs. Pour ce faire, il est nécessaire de multiplier par N tous les vecteurs utilisés dans la programmation dynamique classique pour conserver l'information utile.

Pour un développement efficace, il est important d'avoir :

– un accès rapide à la séquence de mots associée à un point de l'espace de recherche. Ceci peut être réalisé en construisant un arbre contenant toutes les séquences de mots possibles et en utilisant uniquement le numéro de la séquence lors de la recombinaison.

– un algorithme de tri rapide pour la recombinaison des N meilleures distances.

Il est donc possible de modifier l'algorithme One-Pass pour générer une liste des N meilleures séquences de mots reconnus. Pourtant, des tests effectués par Steinbiss [112] montrent que, même en utilisant des techniques d'élagage, pour un vocabulaire moyen (1000 mots), l'espace de recherche des N meilleures séquences de mots reste trop important. L'effort nécessaire pour calculer ces N meilleures solutions est immense. Ce problème vient principalement du coût nécessaire à la recombinaison et à la création de la séquence.

En effet, à chaque instant, il faut mémoriser N scores et séquences de mots reconnus. De plus, à chaque instant, il faut trier les N meilleurs chemins pour décider des chemins à développer.

Une méthode exacte fournissant les N meilleures séquences de mots à partir de l'algorithme "One Pass" (Viterbi synchrone) a été décrite. Malheureusement, cette méthode est trop lourde en temps de calcul et en mémoire pour pouvoir être appliquée en reconnaissance de la parole continue pour des vocabulaires relativement importants. Des méthodes sous-optimales ont été proposées pour tenter de résoudre ces problèmes. Ces solutions sont moins coûteuses en temps de calcul, mais le résultat obtenu est sous-optimal. C'est à dire que les N meilleures solutions fournies ne sont pas nécessairement les N meilleures séquences de mots.

4.3.2 Méthode sous-optimale

La méthode définie ci-dessus comportait quelques défauts qui la rendaient inutilisable en particulier pour un vocabulaire important. Ainsi, Steinbiss proposa une méthode dérivée, sous-optimale et moins coûteuse en calcul. La procédure de recherche précédente engendre un temps de calcul important car il est nécessaire de trier toutes les hypothèses, à chaque instant, pour décider des chemins à développer. Steinbiss [112] proposa d'obtenir les N meilleures solutions à partir de la segmentation de la meilleure solution. Ainsi, chaque solution contient une séquence de mots (un ou plusieurs mots) de la solution optimale. Cette approche ignore donc les recombinaisons des chemins à l'intérieur du mot (celles qui prennent le plus de temps). Le tri n'est effectué qu'aux transitions entre mots. La sous-optimalité de cette méthode est évidente, mais Steinbiss montra que cette approche résolvait le problème de la recherche des N meilleures séquences de mots avec une précision “relativement” suffisante. Cette méthode fut aussi appliquée par Marinõ [72]. Comparé à la recherche de la meilleure solution, cet algorithme requiert juste un peu plus de calcul. Il est donc beaucoup plus rapide que la méthode précédente, mais l'optimalité de la méthode n'est plus vérifiée.

Richard Schwartz [106] a proposé d'agir sur l'état syntaxique au lieu de l'état acoustique. L'état syntaxique correspond en fait aux états situés aux extrémités des modèles HMM de mots (voir figure 2.8 du chapitre 2). Ils sont représentatifs des transitions entre les mots et ont donc une importance un peu plus grande en comparaison des états centraux (dit acoustiques). Richard Schwartz a proposé 3 méthodes différentes, une optimale et deux sous-optimales.

4.3.3 Méthode de recherche en largeur: "Sentence-Dependent N-Best algorithm"

Cet algorithme est quasiment similaire à l'algorithme de programmation dynamique en une passe du type Viterbi déjà développé dans ce chapitre (voir paragraphe 4.2). Sur chaque noeud du réseau syntaxique est définie une liste. Cette liste sert à mémoriser les séquences de mots arrivant sur ce noeud pour chaque instant. L'idée de base est de garder en mémoire toutes les hypothèses correspondant à des séquences de mots différentes. Ainsi, il peut y avoir plusieurs hypothèses différentes avec des séquences de mots différentes pour le même état et au même instant. L'algorithme fonctionne de la manière suivante :

- Si deux chemins arrivant au même noeud (même état et même instant) correspondent à la même séquence de mots leurs probabilités sont sommées.
- Sinon un nouveau chemin est créé.

On ne garde dans la liste de chaque noeud syntaxique que les solutions ayant un score supérieur à un seuil fixé au départ par rapport à la meilleure solution contenue dans cette liste[5]. Les N meilleures séquences de mots sont propagées sur la suite du réseau pour les trames suivantes.

Malheureusement, nous sommes à nouveau confrontés au même problème que pour la méthode développée par Steinbiss. Le temps de calcul est beaucoup trop élevé. C'est pourquoi, deux méthodes sous-optimales dérivées de l'algorithme "Sentence-Dependent N-Best" ont été introduites pour réduire le temps de calcul [107].

- Lattice N-Best Algorithm.
- Word Dependent N-best Algorithm.

4.3.4 Lattice N-Best Algorithm

Cet algorithme permet de fournir une liste approximative des N meilleures phrases avec un temps de calcul légèrement plus important que pour la recherche de la meilleure solution. Cette méthode consiste à garder dans chaque noeud syntaxique les N meilleurs chemins arrivant à ce noeud, puis à propager la meilleure solution sur le reste du réseau. En d'autres termes, à

5. Ce qui revient à la technique d'élagage déjà développée dans le paragraphe 4.2.2.

l'intérieur des mots, l'algorithme Viterbi synchrone classique "One Pass" est appliqué. Une seule théorie par état est donc préservée. En revanche, à chaque noeud correspondant à la transition vers un autre mot, au lieu de garder uniquement le meilleur chemin, les différents mots arrivant à ce noeud et leurs scores respectifs sont mémorisés dans une liste. Cette méthode ne requiert pas plus de temps de calcul que la méthode classique de programmation dynamique fournissant la meilleure séquence de mots reconnue. Le reste du décodage est poursuivi par la procédure classique. A la fin de l'algorithme, on remonte récursivement à partir du dernier noeud pour retrouver les N meilleures solutions à partir de la liste construite lors de la phase aller. Ces solutions sont obtenues d'une manière indirecte à partir de la meilleure solution. Cette idée rejoint celle proposée par Steinbiss et décrite précédemment. Cette méthode reste cependant très peu performante. Elle est en effet clairement sous-optimale. Swchartz [107] a montré que cette méthode est l'une des moins performantes.

4.3.5 Word Dependent N-Best Algorithm

Cette méthode est un compromis entre les deux précédentes. Elle dépend du mot obtenu à l'intérieur des N meilleures solutions. Dans l'algorithme Lattice N-Best, l'instant où commence un mot, est supposé indépendant du mot précédent de la séquence. Dans l'algorithme "Sentence-Dependent N-Best", l'instant où commence un mot dépend de tous les mots précédents de la séquence. L'hypothèse de départ est basée sur la réflexion suivante :

La détermination de l'instant où a commencé un mot, dépend uniquement du mot précédent et non de la séquence de mots qui le précède.

Ainsi, les chemins développés dépendent du mot précédent. A chaque état du mot, on retient la probabilité totale pour n mots précédents différents (avec $n << N$). A la fin de chaque mot, on mémorise le score correspondant à chaque hypothèse ainsi que le nom du mot précédent associé. La programmation dynamique est ensuite étendue avec le nom du mot qui vient juste de se terminer. Enfin, un backtracking est effectué pour obtenir la liste des phrases reconnues les plus probables. Le temps de calcul est proportionnel à n (le nombre de chemins gardés localement). Typiquement, n prend des valeurs de 3 à 6. Le temps de calcul de cette méthode est donc bien supérieur à celui de l'algorithme Lattice N-best, mais reste beaucoup moins important que pour l'algorithme Sentence-Dependent N-Best. Les tests réalisés par Schwartz [106] montrent que lorsque le nombre de solutions demandées augmente, l'algorithme Word Dependent N-Best donne les meilleurs résultats. La stratégie finale adoptée par Schwartz est de rechercher les N meilleures solutions en deux phases :

- Une phase aller en utilisant une grammaire bigramme et des modèles de Markov discrets pour mémoriser les scores de chaque mot rencontré aux différents instants.

- Une phase retour produisant la liste des N meilleures solutions en utilisant l'algorithme des N meilleures solutions dépendant du mot "Word Dependent N-Best Algorithm".

4.3.6 Conclusion

Les méthodes de recherche des N meilleures séquences de mots basées sur un algorithme de programmation dynamique ont été développées dans ce chapitre. Ces méthodes sont malheureusement coûteuses en temps de calcul et en mémoire. Elles sont donc assez rarement utilisées en reconnaissance de la parole continue pour des vocabulaires de taille importante. D'autres méthodes de recherche des N meilleures solutions en deux passes plus efficaces ont fait l'objet de recherches intensives et ont mené aux méthodes basées sur l'algorithme A*.

4.4 Méthodes basées sur l'algorithme A*

Les méthodes de recherche dans un graphe sont utilisées pour générer les N meilleures séquences de mots reconnues. Nous décrirons dans un premier temps les méthodes usuelles de recherche dans un graphe. Puis nous examinerons plus précisément l'algorithme A* en vue d'une reconnaissance des N meilleures séquences de mots [55, 79, 111].

4.4.1 Généralités

Un graphe G est un couple (S,A) tel que :

- S : ensemble d'éléments appelés sommets ou noeuds.

- A : ensemble des couples d'éléments de S, appelés arcs.

A ce graphe G est associé un noeud début noté "n0". Les paires de noeuds $(n, n^{'})$ de l'ensemble A sont connectées par des arcs directs. Si un arc est orienté de n vers $n^{'}$, on dit que $n^{'}$ est le successeur (ou le fils) de n et le noeud n est dit parent (ou père) de $n^{'}$. Un chemin dans G est une suite de sommets. Dans le cas de la reconnaissance automatique de la parole par modèles de Markov cachés, le graphe G représente l'ensemble des chemins possibles dans le réseau acoustique. Un noeud est défini par une position fixée par la paire (q_i, t) où q_i est un état du réseau et t un instant. Les arcs sont définis par une règle de progression à l'intérieur du réseau acoustique.

4.4.2 Méthodes de recherche dans un graphe

Une méthode de recherche dans un graphe détermine l'ordre dans lequel les noeuds doivent être développés afin de trouver le noeud final à partir du noeud de départ. Une méthode de recherche sera d'autant meilleure qu'elle produira un graphe de recherche avec un effort de génération peu coûteux. Il existe deux catégories de méthodes de recherche :

- méthode de recherche aveugle :
  L'ordre dans lequel les noeuds sont développés ne dépend que de l'information collectée par la recherche.
- méthode de recherche ordonnée :
  Une information partielle sur le domaine du problème et sur la nature du but à atteindre est utilisée pour aider le guidage de la recherche vers une direction plus prometteuse.

Méthodes de recherche aveugle

Dans ce type de méthodes, l'algorithme de choix des noeuds à développer ne dépend que des connaissances de la position du noeud objectif ou d'un chemin vers la position de ce noeud objectif. Il existe deux types de méthodes :

- méthode de recherche en largeur.

  Il suffit de générer toute la descendance de la racine en appliquant chacune des règles sur le noeud initial. Ensuite pour chaque noeud fils, il faut générer tous les successeurs en appliquant les règles appropriées. La progression de la recherche est visible sur la figure 4.7. Les noeuds sont numérotés dans l'ordre de leur développement.

  L'exploration est effectuée par couche en utilisant une pile du type FIFO (First In First Out, premier arrivé premier servi). Ces méthodes garantissent l'obtention d'une solution exacte si elle existe. Néanmoins cette méthode est coûteuse en mémoire.

- méthode de recherche en profondeur.

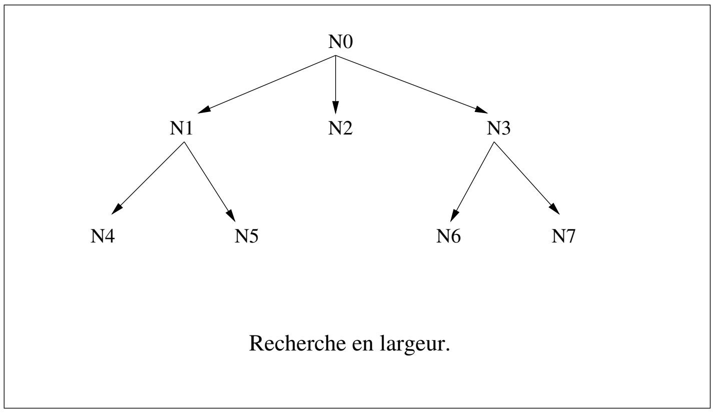

FIG. 4.7 - *Progression de recherche en largeur.*

Cette méthode consiste à développer tous les successeurs d'un sommet avant de traiter un autre sommet. Lorsqu'il n'y a plus de des-

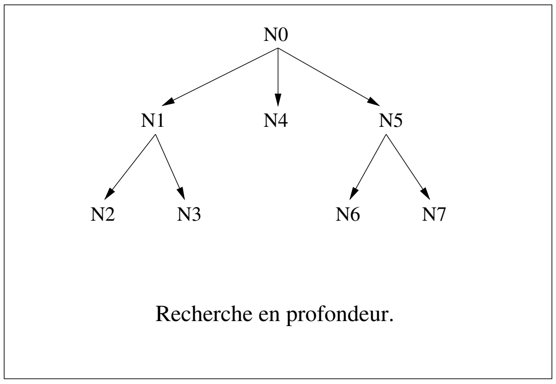

FIG. 4.8 - *Procédure de recherche en profondeur.*

cendants, on remonte jusqu'au dernier niveau où il reste des sommets à explorer. Le danger de cette méthode est la possibilité d'explorer indéfiniment une branche stérile. C'est pourquoi une profondeur limite dans le cas de graphes infinis est fixée. La progression de recherche est visible sur la figure 4.8. Les noeuds sont numérotés dans l'ordre de leur développement. Cette méthode correspond à une gestion de la pile en LIFO (Last In First Out, dernier arrivé premier servi). Les méthodes de recherche aveugle sont des méthodes exhaustives pour la recherche du noeud final dans un graphe contenant un nombre très élevé de noeuds. Les limitations de ces méthodes sont dues essentiellement au temps de calcul et à la capacité mémoire requise pour développer tous les noeuds.

Méthodes de recherche ordonnée

Il est possible d'utiliser d'autres informations pour guider la recherche et donc réduire le temps de traitement. Chaque information est spécifique d'une tâche donnée et d'un graphe donné. Ce type d'information est appelé heuristique. Une stratégie possible consiste à déterminer le meilleur noeud à développer parmi tous les noeuds rencontrés indépendamment de leur position dans le graphe. Cette stratégie est aussi appelée l'algorithme du "meilleur d'abord". Elle utilise pour sa recherche une fonction coût f, appliquée à chaque noeud à développer. Cette fonction représente une estimation du coût minimum global du chemin entre la racine et l'objectif passant par le noeud n. Plusieurs types d'algorithmes utilisent cette stratégie, parmi eux l'algorithme A*. Le principal avantage de ce type d'algorithme vient du fait qu'il n'est plus nécessaire de parcourir l'espace complet pour trouver la solution. Nous allons décrire cet algorithme et ses différentes caractéristiques.

4.4.3 L'algorithme A*

Cet algorithme [79, 111] est l'un des plus connus et des plus utilisés pour générer les N meilleures séquences de mots. Lors de la recherche avec l'algorithme A*, une pile des hypothèses des différentes séquences de mots se terminant à différents instants est gardée. A chaque hypothèse est associée une estimation du score correspondant lorsqu'elle est étendue à la fin de la phrase. La théorie possédant le score estimé le plus élevé est étendue.

Lorsque l'une de ces théories atteint le noeud initial, si le score correspondant est encore le plus élevé, alors cette théorie est l'hypothèse d'une phrase reconnue. Lorsque N séquences de mots ont été accumulées avec un score plus élevé que pour toutes les autres théories, nous possédons alors les N meilleures séquences de mots. La principale difficulté liée à cet algorithme est l'estimation du score de la théorie qui est traitée. La figure 4.9 schématise la manière dont le processus fonctionne. Le score pour le noeud Ni, f(Ni), est calculé en sommant les coûts g(ni) et h(ni).

- h(ni) étant la distance cumulée ($-logP$) pour aller du noeud initial N0 au noeud Ni.
- g(ni) étant le distance cumulée pour aller du noeud Ni au noeud final NF.

Ces distances peuvent être reliées aux récurrences α (h(ni)) (2.17) et β (g(ni)) (2.19) déjà présentée dans le chapitre 2 permettant le calcul de $P(X|M)$ (2.21).

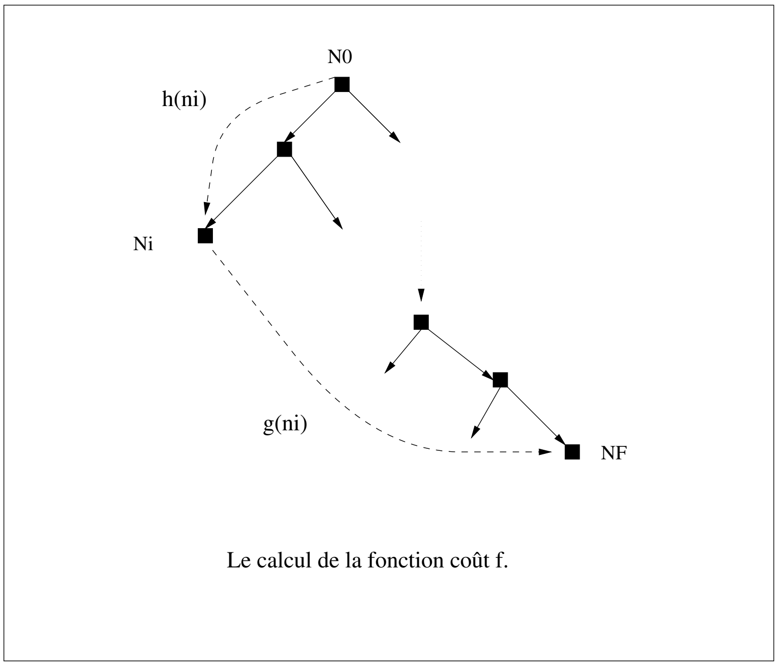

FIG. 4.9 - *Processus de recherche basé sur l'algorithme A*.*

Il existe de nombreuses possibilités d'utiliser cet algorithme (ou d'estimer la fonction coût de chaque hypothèse) pour générer les N meilleures séquences de mots. Nous allons développer la méthode en deux passes qui a été retenue. Elle s'inspire fortement de celle présentée par Soong dans [55, 111]. La première passe, effectuée de manière synchrone, calcule par l'algorithme de Viterbi les probabilités d'émission (meilleur chemin) à chaque instant et pour chaque état acoustique. Ces probabilités seront utilisées plus tard pour calculer le score associé à chaque hypothèse. La phase retour (asynchrone) utilise l'algorithme A* pour retrouver les N meilleures séquences de mots, c'est à dire les N meilleurs chemins entre le noeud final (dernier état du dernier mot pour la dernière trame) et le noeud de départ (premier état du premier mot à la première trame).

4.4.4 Description de la méthode utilisée

La méthode de recherche choisie se déroule en deux phases.

1. Une phase aller :

   Elle est réalisée par l'algorithme de Viterbi (One Pass). Cet algorithme se charge de calculer et de mémoriser, pour chaque instant t et pour chaque état q_j du réseau acoustique, la probabilité maximale d'observation des t premières trames le long du meilleur chemin atteignant l'état q_j à l'instant t. Le coût du noeud $n_j = (t,q_j)$ est déterminé par la récursion classique de la programmation dynamique Viterbi (voir équation (4.3))

2. Une phase retour :

   L'objectif de cette phase est de retrouver, à partir du noeud final du réseau acoustique, le noeud de départ en développant les N meilleurs chemins syntaxiquement différents. Cette phase est réalisée par l'algorithme A*. Le coût global de la séquence de mots correspondant à un chemin en cours de développement à l'instant t est défini par :

   $f(n_j) = g(n_j) + h(n_j)$

   - $g(n_j)$ représente le coût du chemin entre le noeud final et le noeud n_j.
   - $h(n_j)$ représente une estimation du coût du chemin restant à parcourir entre le noeud n_j et le noeud initial.

Dans cette approche, l'estimation de la portion du signal non encore traitée provient des valeurs mémorisées pendant la phase aller. Cette estimation (au sens de l'algorithme A*) est optimale et conduit alors à une recherche très efficace des N meilleures solutions.

Algorithme utilisé :

1. Initialisations

2. Première passe :
   Calcul des probabilités cumulées pour tous les noeuds n_j.
   Programmation dynamique classique type Viterbi $\Longrightarrow$ mémorisation des scores.

3. Seconde Passe : Algorithme A*.

   (a) Se placer au dernier noeud.
   (b) Soit C le meilleur candidat.
   (c) v = noeud initial du candidat C.
   (d) Si v = noeud de départ alors séquence trouvée
   (e) Soit U l'ensemble des prédécesseurs de v.
   Pour tout u de U $\Longrightarrow$ insérer u dans la pile avec le score f(u).
   (f) trier la pile.
   (g) Si pile non vide retourner à l'étape b.

Dans cet algorithme, l'estimation des scores correspondant à l'extension d'une théorie jusqu'au noeud initial a été réalisée dans la première passe. Il existe plusieurs autres possibilités pour estimer ces scores. L'estimation telle qu'elle a été implémentée ici permet d'obtenir rapidement lors du backtracking les solutions. Généralement, le temps requis pour la phase de backtracking est largement inférieur à celui nécessaire à l'estimation des probabilités (première phase). Cet algorithme a été testé sur la base de données BDSONS (Voir Annexe D). L'algorithme fonctionne en temps réel[6] pour générer les N (N=10 à 50) meilleures solutions pour un système de reconnaissance de chiffres connectés. Cependant pour des vocabulaires plus importants (> 5000 mots), le temps nécessaire à la génération des N meilleures séquences reste élevé. Il est nécessaire pour de tels vocabulaires d'ajouter des méthodes d'élagage du type de celles décrites au paragraphe 4.2.2.

4.4.5 Extension aux grands vocabulaires

L'efficacité de l'algorithme A* est liée à la manière dont la fonction h est estimée. Si h est un estimateur parfait, alors l'algorithme A* convergera immédiatement vers le but sans développement de chemins inutiles. Si h (distance cumulée) est surestimée, nous ne pouvons avoir l'assurance de trouver le chemin solution le moins coûteux. En revanche, si h est sous estimée, la meilleure solution peut être trouvée [79]. Plus h est proche de l'estimateur parfait (tout en étant inférieur), plus nous obtiendrons rapidement la meilleure solution. L'algorithme A* est assuré de trouver le chemin optimal si h est minorante et si la taille de la pile est suffisante pour assurer le développement de la solution [81].

6. Sur une station SUN-SPARC, OS = Solaris 2.5.6

L'algorithme de reconnaissance de la parole continue grand vocabulaire que nous utiliserons dans ce travail utilise ce type d'algorithme A* pour générer la meilleure séquence de mots. La seule différence réside dans l'estimation de la fonction coût f. Nous détaillerons plus précisément cet algorithme (Noway) dans le chapitre 6. Cet algorithme est aussi à la base de l'étude que nous avons réalisée pour générer les N meilleures séquences dans le cas de grands vocabulaires (13000 ou 65000 mots). Noway est utilisé pour générer dans une première passe un treillis. Les treillis permettent une représentation des phrases reconnues sous forme de graphes. Un treillis est représenté par un ensemble de noeuds et de liens. Chaque lien porte un mot correspondant à une hypothèse et le score acoustique correspondant et chaque noeud correspond à un instant donné. Une représentation graphique d'un exemple de treillis peut être visualisée sur la figure 4.10.

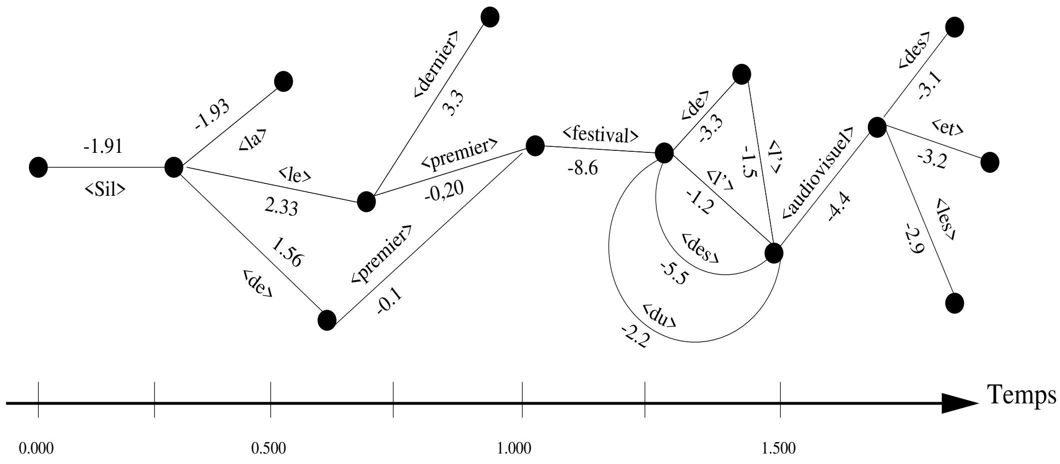

FIG. 4.10 - *Un exemple de treillis.*

Les informations de ce treillis peuvent être utilisées pour générer les N meilleures séquences de mots dans une seconde passe. Ce treillis est utilisé par un algorithme du type Viterbi synchrone tel que celui décrit au début de ce chapitre.

4.5 Conclusion

Nous avons décrit dans ce chapitre les différentes méthodes utilisées pour générer les N meilleures séquences de mots. La plupart des algorithmes utilisent des méthodes dérivées de la programmation Viterbi ou des méthodes de recherche basées sur l'algorithme A*. Les méthodes utilisant une extension de la programmation dynamique classique fonctionnent bien pour des vocabulaires de taille relativement faible. Dès que la taille du vocabulaire devient importante, l'espace de recherche des N meilleures solutions devient beaucoup trop grand (même en utilisant des techniques d'élagage). Dans ce cas, les algorithmes utilisant la méthode A* semblent beaucoup plus efficaces. L'algorithme que nous avons retenu pour générer les N meilleures phrases fonctionne en deux passes. Dans une première passe, une programmation dynamique classique du type Viterbi se charge de mémoriser pour chaque instant et pour chaque état la probabilité maximale d'observation des n premières trames. Dans une seconde passe, l'algorithme A* est utilisé pour générer les N meilleures séquences. Le score associé à l'hypothèse développée est calculé à partir des valeurs mémorisées lors de la première passe. Cette estimation (au sens de l'algorithme A*) étant optimale, elle conduit à une recherche très efficace des N meilleures séquences de mots. Cette technique fonctionne particulièrement bien pour des vocabulaires de tailles relativement importantes (jusqu'à 5000 mots). Pour des vocabulaires beaucoup plus important, nous avons utilisé dans une première passe Noway, un algorithme de reconnaissance de la parole continue grand vocabulaire utilisant aussi l'algorithme A* pour générer la meilleure séquence de mots et un treillis de mots. Dans une seconde passe, ce treillis est utilisé pour générer les N meilleures séquences de mots. Ces techniques développées dans le logiciel STRUT[7] ont été testées sur les corpus Resource Management et BREF-80. Il est maintenant nécessaire de réestimer les scores de ces N meilleures séquences et de tenter ainsi d'améliorer les taux de reconnaissance des systèmes de base. C'est ce que nous proposons de faire dans le chapitre suivant.

7. http://tcts.fpms.ac.be/speech/strut.html.

Chapitre 5

Réestimation des N meilleures séquences

5.1 Introduction

Nous venons de développer dans le chapitre précédent les méthodes utilisées pour générer les N meilleures séquences de mots au lieu de la meilleure uniquement. De récents travaux ont montré l'utilité de générer une liste des N meilleures séquences de mots reconnues afin de trouver, avec l'aide d'informations supplémentaires, la phrase effectivement énoncée (par exemple la phrase satisfaisant à des restrictions supplémentaires ou encore la meilleure phrase après une réestimation du score de chaque hypothèse de la liste). Nous allons développer dans ce chapitre les méthodes que nous avons utilisées pour effectuer une réestimation des scores de ces N meilleures séquences de mots. Le type d'informations supplémentaires utilisées pour réordonner ces séquences varie d'une information temporelle (durée depuis laquelle on est dans le phonème) à des modèles dépendants du contexte ou des modèles de langage plus compliqués.

5.2 Utilisation de modèles dépendants du contexte

Les modèles dépendants du contexte (voir chapitre 8 pour une étude détaillée) peuvent être utilisés efficacement pour réestimer les scores des N meilleures séquences de mots. Dans le cas de la reconnaissance de la parole continue, il est difficile d'utiliser des phonèmes dépendants du contexte pour modéliser les transitions entre les mots ("cross-word"). En effet, lorsqu'on se trouve à la fin d'un mot lors de la programmation dynamique, il faut pouvoir étendre le mot avec le bon contexte qui est fonction des successeurs du mot traité. Il existe un nombre important de possibilités pour modéliser la transition entre ce mot et tous les successeurs possibles; c'est pourquoi généralement, dans le cas de grands vocabulaires, on se contente d'utiliser le plus souvent des modèles dépendants du contexte à l'intérieur des mots uniquement ("intra-word"). Dans le cas d'un recalcul des scores des N meilleures séquences de mots, le problème ne se pose plus car on connaît cette fois précisément la séquence de mots et donc les modèles dépendants du contexte qu'il faut utiliser pour modéliser précisément les transitions entre les mots. Ce type de réestimation est donc assez simple puisqu'il ne nécessite que les N meilleures séquences de mots et les probabilités associées aux modèles dépendants du contexte pour recalculer, pour chaque hypothèse, un nouveau score tenant compte d'une information supplémentaire importante: le contexte acoustique. Le système complet peut être visualisé sur la figure 5.1.

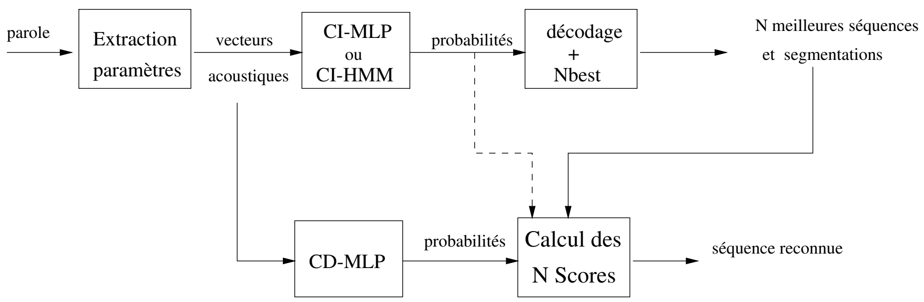

FIG. 5.1 - *Le système de reconnaissance utilisant les modèles dépendants du contexte pour réestimer les scores des N meilleures séquences de mots.*

5.2.1 Test de la méthode

Ce type de méthode a été testé sur la base de données RESOURCE MANAGEMENT (voir Annexe D). Nous avons généré les 20 meilleures phrases pour toutes les données de test (300 phrases de feb89) à partir du système basé sur les phonèmes indépendants du contexte (voir chapitre 2). Les modèles dépendants du contexte utilisés pour réestimer le score de chaque hypothèse sont du type de ceux décrits dans le paragraphe 8.6. Les transitions entre phonèmes sont modélisées par des distributions généralisées alors que les unités phonétiques de base sont toujours des modèles de phonèmes indépendants du contexte entraînés sur la partie la plus stable (généralement celle correspondant au milieu) du phonème. Les mots sont ainsi modélisés par des séquences de distribution de probabilité (estimées par l'intermédiaire d'un MLP) représentant alternativement la partie centrale du phonème et les transitions entre les phonèmes.

Les transitions entre les phonèmes ont été quantifiées en un nombre limité de transitions généralisées pour réduire le nombre de paramètres à entraîner. Dans les expériences reportées ici, nous n'utiliserons qu'un unique réseau (MLP) pour estimer à la fois les probabilités associées aux modèles indépendants du contexte et celles associées aux phonèmes dépendants du contexte généralisés (basés sur le lieu de l'articulation ou sur la classe phonétique voisine) représentant les transitions entre phonèmes. Ces modèles ont été entraînés sur les 3990 phrases des données d'entraînement. Les résultats obtenus avec ce type de réestimation sont présentés dans le tableau 5.1

Modèles	Ins.	Sup.	Subs.	Taux Erreur
CI-MLP	0.6%	1.2%	3.6%	5.4%
articul.-CD	0.3%	1.2%	2.8%	4.3%
classe-CD	0.6%	1.1%	3.3%	4.9%

TAB. 5.1 - *Taux d'erreur au niveau du mot pour feb89 avec un modèle hybride HMM/ANN et une grammaire wordpair (perplexité 60) (CI) et après recalcul des scores des 20 meilleures séquences de mots avec des modèles dépendants du contexte CD généralisés basés sur le lieu de l'articulation ou sur la classe phonétique des phonèmes droits et gauches.*

5.2.2 Conclusion

Les résultats obtenus avec ce type de réestimation sont bons, puisque nous avons observé une réduction du taux d'erreur de l'ordre de 20% par rapport au modèle hybride de base. Il est donc possible de réévaluer de manière efficace les scores des N meilleures séquences de mots en utilisant des modèles dépendants du contexte. Le choix de $N = 20$ correspond à un optimum après avoir essayé plusieurs valeurs de N. Nous avons utilisé, pour la génération des N meilleures séquences de mots, l'algorithme A* en deux passes développé dans le chapitre précédent. Il faut noter que les modèles dépendants du contexte utilisés pour le recalcul des scores ne possèdent qu'un nombre légèrement supérieur de paramètres par rapport au modèle hybride classique (voir paragraphe 8.6)[1]. Cette technique est donc très efficace et rapide.

Les quelques tests effectués sur le corpus BREF-80 sont beaucoup moins impressionnants. Aucune amélioration n'a été observée. Il faut noter qu'une possible explication des résultats observés sur ce corpus est que les modèles dépendants du contexte, que nous avons utilisés pour recalculer les scores de chaque hypothèse, sont probablement moins bien entraînés.

5.3 Utilisation de modèles de langage plus performants

Une des méthodes de réévaluation des scores les plus utilisées en reconnaissance de la parole consiste à utiliser un modèle de langage plus performant pour trouver parmi les N meilleures séquences de mots celle qui

1. 146 sorties au lieu de 47, soit une augmentation du nombre de paramètres de 100000, ce qui reste faible pour un système dépendant du contexte

correspond à des règles grammaticales plus précises. Nous avons déjà signalé que l'utilisation de modèles de langage du type k-gramme (avec $k > 2$) complique fortement le processus de reconnaissance. Ainsi, de nombreux système de reconnaissance se contentent d'utiliser dans une première passe un modèle de langage plus simple (bigramme par exemple) pour générer les N meilleures séquences de mots. Ensuite, un modèle de langage plus compliqué (du type trigramme par exemple) est utilisé pour recalculer les scores des N meilleures séquences de mots. Ce procédé visant à séparer en deux passes le processus de reconnaissance est largement utilisé en reconnaissance de la parole. Nous avons appliqué ce type de technique au corpus BREF-80. Nous avons en effet généré les N ($N = 200$) meilleures séquences de mots avec notre système CI (voir paragraphe 6.4) en utilisant un modèle de langage du type bigramme (entraîné sur les textes du journal "Le Monde"), et nous avons utilisé un k-gramme ($k > 2$) pour recalculer les scores de ces N meilleures séquences de mots. Nous possédons (voir paragraphe 6.4) trois modèles de langage entraînés sur les textes du journal "Le Monde" (années 90-91-92 soit environ 80 millions de mots) pour le corpus BREF-80 :

- un bigramme de perplexité 151.6 utilisé pour la génération des N meilleures séquences de mots.
- un trigramme de perplexité 94.4 utilisé pour la génération des N meilleures séquences de mots et la réestimation des scores des N meilleures séquences fournies par le bigramme.
- un 5-gramme de perplexité 80.3 utilisé pour la réestimation des scores des N meilleures séquences de mots fournies par le bigramme et le trigramme.

Ces modèles de langage ont été utilisés pour réestimer simplement les scores des 200 meilleures phrases. Les résultats obtenus avec ce type de méthode sont présentés dans le tableau 5.2

Il est donc possible d'utiliser des modèles de langage d'ordre supérieur pour recalculer de manière efficace les scores des N meilleures séquences de mots. Nous avons observé une réduction relative du taux d'erreur de l'ordre de 7% et 5% en utilisant un 5-gramme pour réévaluer les scores des $N = 200$ meilleures phrases obtenues avec un système hybride et un modèle de langage bigramme et trigramme sur le corpus BREF-80. Les résultats obtenus avec ce type de technique corroborent ceux obtenus par d'autres laboratoires. L'amélioration observée est sensiblement inférieure à celle obtenue avec la méthode précédente. Ceci peut s'expliquer par le fait que le 5-gramme utilisé n'a été entraîné que sur très peu de données (80 millions de mots). Nous

Modèles	Ins.	Sup.	Subs.	Taux Erreur
CI-MLP+2-grams	2.8%	6.0%	18.9%	27.7%
Rescore 3-grams	3.1%	6.2%	17.4%	26.7%
Rescore 5-grams	2.8%	6.1%	17.1%	25.9%
CI-MLP+3-grams	2.5%	6.0%	16.5%	25.0%
Rescore 5-grams	2.6%	5.1%	16.1%	23.7%

TAB. 5.2 - *Taux d'erreur au niveau du mot pour un vocabulaire de 65K mots avec un modèle hybride HMM/ANN et des modèles de langage bigramme et trigramme (CI) et après réestimation des scores des 200 meilleures séquences de mots avec des modèles de langage d'ordre supérieur (trigramme et 5-gramme pour le bigramme et 5-gramme pour le trigramme).*

aurions probablement obtenu de meilleurs résultats si nous avions entraîné ce modèle sur les 10 années du journal "Le Monde" (au lieu de 3 années).

5.4 Le modèle temporellement indexé

Le modèle "Time-Index" que nous pourrions traduire par "indexé temporellement" a été introduit pour la première fois par Koenig dans sa thèse [62]. Il avait réussi à obtenir des résultats impressionnants sur la base de données Resource Management (Voir Annexe D). Nous avons essayé dans ce chapitre de mener un peu plus loin l'étude qu'avait commencée Koenig sur ces modèles particuliers. Dans un premier temps, nous décrirons en détail le modèle indexé par le temps et les problèmes qui lui sont liés. Puis, nous donnerons une méthode pour estimer ce fameux index temporel et utiliser ce type de modèle pour réestimer les scores des N meilleures séquences de mots. Enfin, nous présenterons les résultats obtenus sur la base de données PHONEBOOK (Voir Annexe D) et les quelques conclusions que nous en avons tirées.

A la base, ce modèle est né de la question suivante : Quels types d'informations supplémentaires (autres que les paramètres acoustiques) pourrions-nous utiliser pour construire des modèles plus performants permettant une amélioration sensible des taux de reconnaissance? Une des informations qu'il est possible de fournir à l'entrée d'un MLP est la durée depuis laquelle on est présent dans le phonème. Cette information temporelle peut être facilement estimée lors de l'entraînement par l'intermédiaire de la segmentation Viterbi. Nous sommes en effet capables d'estimer à chaque instant le temps depuis lequel nous nous trouvons dans un phonème particulier. Ainsi il est facilement possible d'introduire cette information temporelle dans l'architecture du MLP selon la figure 5.2. Lorsqu'il est entraîné en mode classificateur et avec suffisamment de données, le MLP de la figure 5.2 estime la probabilité $p(q_k|X, t_i)$.

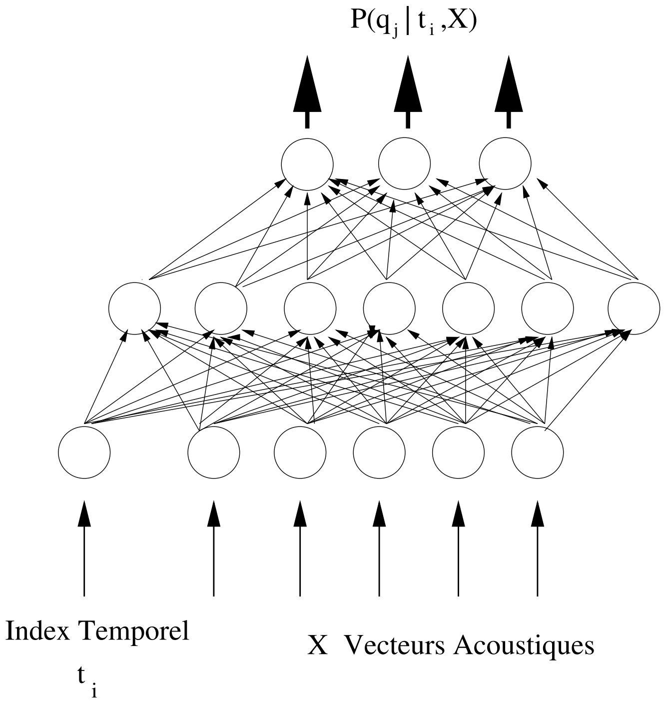

FIG. 5.2 - *La topologie du réseau time index.*

Nous cherchons à estimer $P(X_t^{t+T}|q_j)$, avec $X_t^{t+T} = X_t, X_{t+1}, ..., X_{t+T}$ or

$$P(X_t^{t+T}|q_j) = \prod_{i=t}^{t+T} p(X_i|q_j, i - t + 1) \tag{5.1}$$

Et en utilisant la loi de Bayes,

$$\frac{P(X|q_j, t_i)}{P(X)} = \frac{p(q_j, t_i|X)}{p(q_j, t_i)} \tag{5.2}$$

et comme

$$p(q_j, t_i|X) = p(q_j|t_i, X).p(t_i|X) \tag{5.3}$$

Il vient alors :

$$\frac{P(X|q_j,t_i)}{P(X)} = \frac{p(q_j|t_i,X).p(t_i|X)}{p(q_j,t_i)} \tag{5.4}$$

avec $p(q_j|t_i,X)$ qui peut être estimé par le réseau de la figure 5.2 et $p(t_i,q_j)$ qui peut être estimé par comptage sur les données d'entraînement. Il reste donc à estimer $p(t_i|X)$. Il existe plusieurs manières d'estimer cette probabilité, mais dans un premier temps, pour vérifier la validité du modèle, un petit test préliminaire a été effectué.

5.4.1 Test du modèle Time Index

Nous avons décidé pour tester ce modèle d'imposer les index temporels (t_i). En effet à partir d'une segmentation Viterbi classique des données de test, il est possible de calculer pour chaque instant l'index temporel correspondant de la même manière que lors de la phase d'entraînement. Il est important de noter ici que seule l'information temporelle est sauvegardée, aucune information supplémentaire (phonétique par exemple) n'est gardée. Les tests effectués sur la base de données PHONEBOOK (Voir Annexe D) sont assez impressionnants et valident le modèle décrit dans ce chapitre. Les vecteurs acoustiques utilisés sont les log RASTA-PLP (voir paragraphe 2.3.3). Le MLP contient donc 235 entrées correspondant à 9 trames acoustiques $(9 * 26$ paramètres et l'index temporel), 1000 noeuds cachés et 47 sorties. Le nombre de paramètres du nouveau réseau est donc sensiblement égal à celui du réseau de base. Ainsi en fixant les index temporels pour l'ensemble de test, il est possible de calculer, pour tous les états, $p(q_j|t_i,X)$ et d'effectuer une programmation dynamique classique pour trouver la séquence reconnue. Le tableau 5.3 présente les résultats obtenus pour des vocabulaires de 75, 600 et 8000 mots avec le modèle indexé temporellement. Dans ces expériences, la formulation (5.4) quelque peu modifiée (en divisant par $p(q_j)$ au lieu de $p(q_j,t_i)$) a été adoptée.

Il est facile de constater que le taux d'erreur a diminué très fortement en utilisant le modèle dépendant du temps décrit dans ce chapitre. Il faut néanmoins rester conscient que les taux de reconnaissance obtenus ici constituent des bornes supérieures puisque nous avons supposé connu l'index temporel pour les données de test (estimé à partir d'une segmentation Viterbi). Ces premiers résultats sont très encourageants et viennent corroborer ceux obtenus par Koenig [62]. Il reste néanmoins à trouver une méthode efficace pour calculer cet index temporel.

Voc.	système de base	Time Index	R. T. E.
75	2.7%	1.0%	63%
600	7.0%	3.9%	44.3%
8000	22.0%	14.7%	33.2%

TAB. 5.3 - *Taux d'erreur pour trois dictionnaires différents (75, 600 et 8000 mots) et utilisant des paramètres log RASTA-PLP pour le modèle indexé temporellement et le réseau de base sur le corpus* PHONEBOOK.

5.4.2 Utilisation du modèle en reconnaissance de la parole

Il est possible de mettre au point de nombreuses méthodes pour estimer le facteur $p(t_i|X)$ ou l'index temporel. La plus simple consiste à entraîner un MLP pour classifier les t_i en lui présentant les vecteurs acoustiques à l'entrée. Malheureusement les quelques essais effectués dans cette direction se sont révélés infructueux, ce qui peut se comprendre aisément du fait que les données acoustiques n'ont que très peu de rapport avec l'index temporel. Il est donc très difficile d'apprendre au réseau à prédire un certain index temporel en fonction d'une séquence de vecteurs acoustiques fournie à l'entrée. D'un autre côté, il est possible de calculer pour tous les t_i les probabilités $p(q_j|t_i, X)$ et d'effectuer une programmation dynamique dans un espace à trois dimensions (le temps, les états et l'index temporel). Mais dans ce cas précis la charge de calcul s'en trouverait considérablement alourdie. Là encore, les quelques essais que nous avons menés se sont révélés infructueux.

La manière la plus simple d'utiliser ce modèle était de recalculer les scores associés aux N meilleures séquences de mots issues du processus de reconnaissance classique avec notre modèle indexé temporellement. En effet, il est possible, pour chacune des N meilleures hypothèses générées par des algorithmes du type de ceux décrits dans les paragraphes 4.3 et 4.4, de sauvegarder la segmentation associée à chacune des hypothèses. Dans le cas d'une reconnaissance de mots isolés, le problème est encore plus simple. Il suffit de choisir les N meilleurs scores parmi l'ensemble des scores pour chaque mot du dictionnaire. Nous pensions donc utiliser cette segmentation pour calculer à chaque instant et pour chaque hypothèse l'index temporel correspondant. Il nous était alors facile d'utiliser le modèle indexé temporellement pour réestimer les scores de chacune des hypothèses produites par le premier processus. Le processus de reconnaissance que nous avons testé peut être résumé par la figure 5.3.

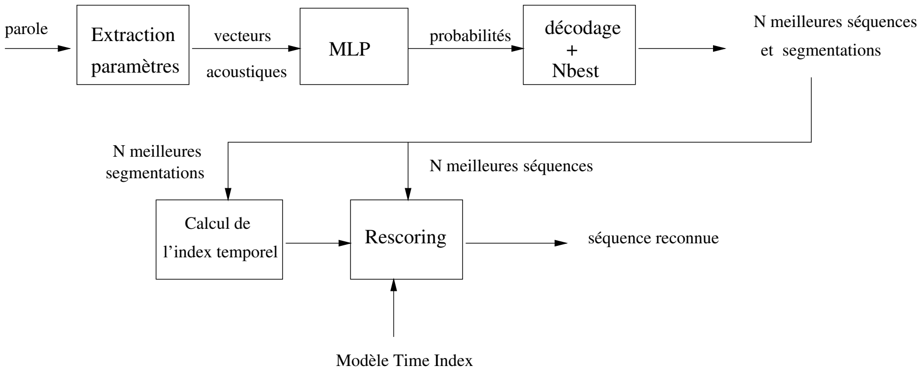

FIG. 5.3 - *Le système de reconnaissance utilisant le réseau time index.*

Sachant que dans le cas particulier de la base de données PHONEBOOK, pour un vocabulaire de 600 mots, plus de 99% des solutions sont présentes dans les 10 premières propositions; nous avons décidé de tester ce système sur cette base de données (cela nous permettait en outre de comparer avec les résultats de notre système de base). Malheureusement les résultats ne sont pas à la hauteur de nos espérances puisque aucune amélioration n'a été observée par rapport au système de base. Une possible explication viendrait du fait qu'il existe une différence notoire entre les segmentations obtenues après génération des N meilleures solutions et celles utilisées pour l'entraînement du réseau "time index". Ainsi, il se peut que certaines séquences d'index temporel observées lors de la génération des N meilleures solutions n'aient jamais été vues lors de l'entraînement. Une analyse un peu plus fine des résultats a montré que les scores obtenus après réestimation des N meilleures séquences de mots étaient sensiblement les mêmes. Nous espérions en utilisant ce modèle écarter certaines segmentations (donc solutions) peu probables. Il serait peut être nécessaire pour améliorer le système d'utiliser des contre-exemples comme c'est souvent le cas dans ce type de méthode. Il est clair que l'information supplémentaire fournie au MLP aide considérablement le processus de reconnaissance. Malheureusement, nous ne sommes pas en mesure pour le moment d'estimer directement cette information.

5.5 Conclusion

Nous avons décrit dans ce chapitre les différentes méthodes que nous avons testées en vue d'une réestimation des scores des N meilleures phrases. Dans un premier temps, nous avons utilisé des modèles dépendants du contexte pour recalculer un score pour chacune des hypothèses. Les modèles dépendants du contexte utilisés sont assez simples. Il est possible de trouver une description précise de ces modèles au paragraphe 8.6. Nous avons observé une réduction significative du taux d'erreur de 20% sur le corpus Resource Management par rapport au meilleur système hybride HMM/MLP. Nous avons aussi montré qu'il était possible d'utiliser des modèles de langage d'ordre supérieur pour recalculer les scores des N meilleures phrases de manière efficace (réduction relative de 7% du taux d'erreur sur le corpus Bref-80). Nous espérions, en utilisant le réseau indexé temporellement, être capable de réestimer de manière efficace les scores des N meilleures phrases. En effet, nous avons montré que ce réseau, si on lui fournit le bon index temporel, était largement supérieur à la plupart des autres modèles. Or il est possible de calculer, pour chacune des N meilleures hypothèses, la séquence d'index temporel correspondante et donc d'utiliser ce modèle pour recalculer un score pour chaque hypothèse. L'utilisation du réseau indexé temporellement n'a apporté aucune amélioration par rapport au système de base. Le principal problème vient probablement du fait que les segmentations obtenues lors de la génération des N meilleures solutions et celles utilisées pour l'entraînement du réseau indexé temporellement sont différentes. Il semble nécessaire pour améliorer le système d'utiliser des contre-exemples (par exemples les segmentations des N meilleures hypothèses des données d'entraînement). C'est d'ailleurs la même observation qui avait été faite par Zavaliagkos [121] lors de l'entraînement de son modèle SNN (Segmental Neural Network) pour effectuer la réestimation des scores des N meilleures séquences de mots issues d'un modèle HMM classique[2].

2. Rappelons que les réseaux SNN (Segmental Neural Network) diffèrent quelque peu de l'approche classique développée dans ce travail par le fait qu'ils essayent de calculer une probabilité pour chaque phonème en utilisant à l'entrée toutes les trames acoustiques correspondant à un segment phonétique.

Chapitre 6

Le décodeur grand vocabulaire Noway

6.1 Introduction

NOWAY [88, 89] est un algorithme de recherche efficace pour la reconnaissance de la parole continue grand vocabulaire développé par S. Renals (Université de Sheffield). Cet algorithme utilise les probabilités a posteriori fournies par un réseau de neurones pour effectuer un élagage efficace au niveau du phonème afin de réduire le temps de calcul nécessaire à la génération d'une hypothèse.

Pour tout type de méthode utilisé en reconnaissance de la parole continue grand vocabulaire, les mêmes problèmes doivent être résolus. Il faut en effet trouver la séquence de mots la plus probable étant donné une séquence de vecteurs acoustiques, un modèle acoustique et un modèle de langage et ce, quasiment en temps réel. Ce problème est très complexe car chaque mot du dictionnaire peut être actif à chaque trame acoustique. Ainsi, plus la taille du vocabulaire est grande, plus le problème est compliqué. Noway (ainsi que la plupart des autres systèmes de reconnaissance de la parole grand vocabulaire) utilise différentes méthodes d'élagage et d'organisation des données pour réduire la taille effective de l'espace de recherche.

Deux types de recherche ont été utilisés par les logiciels de reconnaissance de la parole continue grand vocabulaire :

- les algorithmes Viterbi synchrones [38, 78].
- les algorithmes de recherche asynchrones du type A* [79, 88].

Les algorithmes de décodage du type Viterbi synchrones deviennent beaucoup plus complexes lorsque des modèles de langage du type N-gammes (avec $N \geq 3$) sont utilisés. De nombreuses techniques ont été testées pour résoudre ces problèmes comme l'utilisation de structures dynamiques de recherche ou des techniques de recherche en plusieurs passes. Steve Renals a plutôt choisi l'approche A* qui possède de nombreux avantages par rapport au décodage Viterbi.

- Le modèle de langage est découplé du modèle acoustique et n'est pas utilisé pour générer de nouvelles hypothèses.
- Il est facile d'incorporer des sources de connaissances autres que Markovienne sans étendre de manière drastique l'espace de recherche.

L'un des désavantages de cette approche est sa sensibilité au choix de l'estimation du meilleur score possible dans l'extension de l'hypothèse. En effet, un mauvais choix peut amener à une explosion de la taille de l'espace de recherche. A la base, Noway a été écrit pour tirer avantage du fait que des réseaux de neurones (MLP ou RNN) sont utilisés pour estimer les probabilités a posteriori $p(q_k|X)$. Il va de soi que l'algorithme est aussi applicable à n'importe quel système HMM standard. Un avantage supplémentaire de ce type d'algorithme est le fait que le critère de Viterbi ou le critère du Maximum de vraisemblance (Voir chapitre 2) peuvent être utilisés indifféremment car l'algorithme de recherche n'utilise à aucun moment l'approximation Viterbi.

6.2 Dictionnaire

Le dictionnaire contenant la liste des mots du vocabulaire est modélisé par l'intermédiaire d'un arbre où chaque noeud de l'arbre correspond à une unité phonétique de base. Ainsi la modélisation de quelques mots est visible sur la figure 6.1.

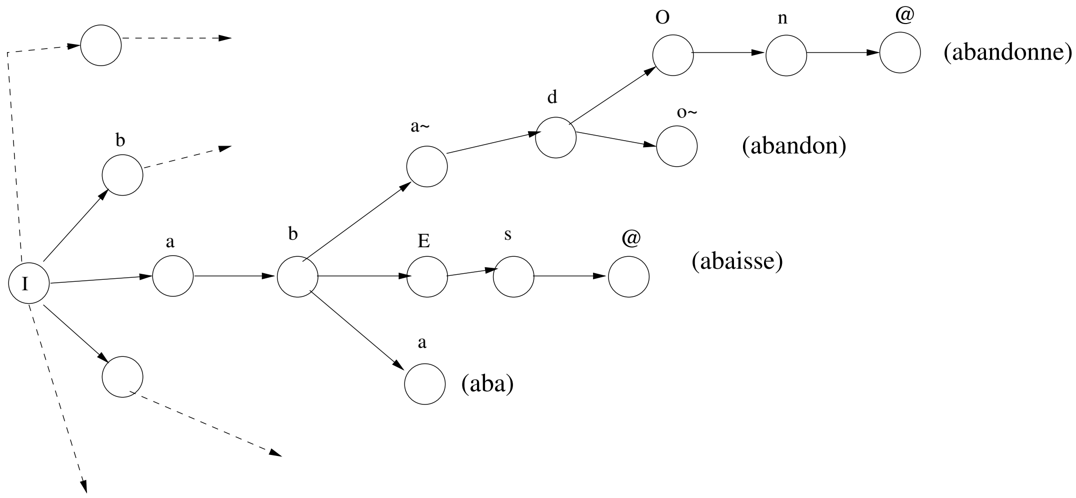

FIG. 6.1 - *Exemples de structure en arbre pour la construction d'un dictionnaire.*

Cette structure en arbre est utilisée par la plupart des outils de reconnaissance de la parole grand vocabulaire car elle permet le partage en mémoire et en temps de calcul pour des mots ayant des prononciations similaires (du moins pour les premiers phonèmes). Dans l'exemple de la figure 6.1, il n'est en effet pas nécessaire d'effectuer la programmation dynamique deux fois pour les mots "abandonne" et "abandon" (comme c'est le cas dans la structure en liste sur la figure 6.2).

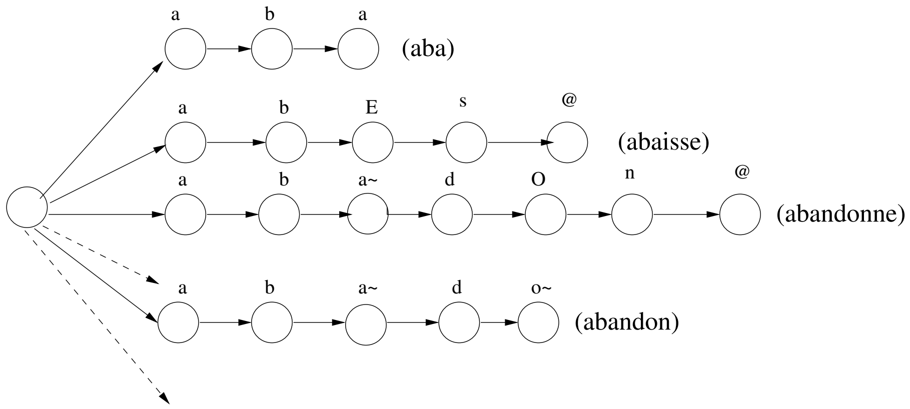

FIG. 6.2 - *Exemples de structure en liste pour la construction d'un dictionnaire.*

Il suffit de le faire 1 fois pour la séquence de phonèmes commune aux deux mots. Les gains au niveau mémoire et temps de calcul obtenus pour ce type de modélisation sont importants. Dans le cas d'un vocabulaire de 60000 mots, si cette structure en arbre n'était pas utilisée nous devrions effectuer les calculs pour les 60000 mots en parallèle (du moins si aucune méthode d'élagage n'est appliquée) alors qu'il n'est nécessaire de travailler que sur les 35 entrées correspondantes au nombre de phonèmes de la langue dans le cas d'une modélisation en arbre.

Cette modélisation du dictionnaire est moins efficace pour la représentation de mots à transcriptions multiples (par exemple le mot "épate" transcrit "e p a t" et "E p a t") ou pour les mots avec des différences mineures entre les prononciations (par exemple les mots "atèle" et "appelle"). Dans le deuxième cas, seul le premier phonème est commun aux deux mots, la programmation dynamique est réalisée pour les deux mots en parallèle alors qu'ils ne diffèrent que d'un unique phonème. Néanmoins pour de grands vocabulaires cette modélisation en arbre du dictionnaire est adoptée par tous les outils de reconnaissance de la parole grand vocabulaire.

Le système de reconnaissance mots isolés, grands vocabulaires de STRUT, construit un graphe (arbre) regroupant en un seul HMM toutes les transcriptions possibles pour tous les mots du vocabulaire. La programmation dynamique à l'intérieur d'un HMM unique est beaucoup plus rapide que celle associée à une liste de HMM (un pour chaque mot). Ce changement de configuration a diminué d'un facteur trois la vitesse du processus de reconnaissance. En outre, l'implémentation de techniques d'élagage couplée avec la représentation en arbre a permis d'effectuer une reconnaissance temps réel de mots isolés pour un vocabulaire d'environ 8000 mots. La mise en oeuvre de chacune de ces méthodes a permis d'obtenir un système de reconnaissance mots isolés, grand vocabulaire, fonctionnant en temps réel sur une simple station SUN-ULTRA-SPARC.

6.3 Algorithme de recherche

L'unité de base de l'algorithme de recherche est la pile. Cette pile supporte la plupart des opérations classiques comme l'insertion, le remplacement, la fusion, le tri ... d'éléments. Chaque élément de la pile est une hypothèse contenant un temps de référence t_h, une séquence de mots W couvrant l'espace temporel $[0..t_h]$ et le log de la probabilité (ou le score) que la séquence de mots W ait été émise étant donné les informations acoustiques, les modèles HMMs et le langage.

Dans la programmation A* classique (voir paragraphe 4.4), les hypothèses de la pile sont ordonnées en fonction de leurs scores respectifs. La probabilité d'une hypothèse h doit inclure, en plus de la probabilité cumulée de la phrase jusqu'à l'instant t_h, une estimation de la probabilité du chemin restant (de l'instant t_h à la fin de la phrase). Ainsi le critère A* usuel calcule le score de l'hypothèse h selon :

$$f_h(t_h) = P_h + b_h^*(t_h) \tag{6.1}$$

où $f_h(t_h)$ est le score d'une hypothèse h avec un temps de référence t_h et un score P_h et $b_h^*(t_h)$ une estimation du score de l'extension de l'hypothèse courante jusqu'à la fin de la phrase.

Cette approche nécessite donc une sorte de "lookahead" et n'est pas complètement synchrone. Il faut estimer ce score. Plutôt que d'utiliser une méthode en plusieurs passes pour estimer le facteur $b_h^*(t_h)$ (voir paragraphe 4.4), Noway approxime le critère A* usuel en calculant la différence entre la probabilité (ou le score) P_h de l'hypothèse h jusqu'au temps t_h et le score de l'hypothèse la plus probable à l'instant t_h (lub). Ainsi dès que l'extension partielle d'une hypothèse possède un score S tel que $S > lub$, alors lub prend la nouvelle valeur du score de l'hypothèse ($lub = S$). Cette approximation du critère permet de comparer des hypothèses ayant le même temps de référence. La valeur de lub est initialisée en utilisant une technique similaire à celle de la reconnaissance de mots-clés. On calcule la moyenne des n meilleures probabilités (excluant la meilleure) et cette moyenne est propagée au travers des trames acoustiques pour générer une valeur initiale de lub. Il va de soi que cette approximation du critère A* ne doit en aucun cas jouer sur l'optimalité de la procédure de recherche. Renals s'est basé sur l'étude de Nilsson [79] qui a montré que tant que $b_h^*(t_h)$ est une borne supérieure du score à l'instant t (c'est à dire qu'il ne sous-estime jamais le score ou ne surestime jamais le coût) alors l'algorithme est optimal, c'est à dire qu'il fournira l'hypothèse optimale [1].

Cette méthode d'estimation de $b_h^*(t_h)$ et donc du critère A* a le mérite de ne requérir aucun calcul supplémentaire, mais elle est bien entendu loin d'être exacte. En particulier, les contraintes de prononciations multiples du dictionnaire et du modèle de langage ne sont pas du tout utilisées pour obtenir une meilleure estimation. Ce type d'estimation du critère ne fonctionne que si elle est couplée à des techniques d'élagage efficaces (telles que celles utilisées par Noway). En effet, sans aucun élagage, le nombre d'hypothèses à traiter serait beaucoup trop important. Renals a développé une méthode un peu moins triviale d'estimation du facteur $b_h^*(t_h)$. Il se base sur les chemins obtenus par backtracking sur toutes les extensions d'un nouveau mot d'une hypothèse. Cette méthode requiert un peu plus de temps de calcul, mais elle assure que l'estimation n'est réalisée qu'avec des chemins correspondant à des extensions complètes de mots. Ces techniques ont prouvé leur efficacité sur de nombreuses bases de données [54, 61, 88, 89].

6.4 Reconnaissance de la parole continue

Nous rapportons dans ce chapitre les tests qui ont été effectués sur une base de données de parole continue grand vocabulaire en français : BREF-80 (Voir Annexe D). Cette base de données n'étant pas segmentée, nous décrirons une méthode originale basée sur un alignement temporel du signal de parole avec un signal synthétique pour générer la première segmentation utile pour initialiser le processus d'entraînement. Nous avons réalisé des expériences de reconnaissance de phonèmes et les premières expériences en reconnaissance de la parole continue pour des vocabulaires de 1000, 3000, 13000 et enfin 65000 mots sont très encourageantes [31]. Les systèmes hybrides HMM/ANN ont déjà été testés avec succès en Anglais américain et Anglais britannique, mais c'est à notre connaissance la première fois qu'ils sont utilisés en français. Nous verrons que certaines particularités de ce langage posent des problèmes difficiles à résoudre.

1. Comparé au critère A* usuel décrit dans le chapitre 4.4, surestimer le score (log P) est équivalent à sous-estimer la distance (-log(P)).

6.4.1 Paramètres utilisés

Les vecteurs acoustiques utilisés dans les expériences décrites dans ce chapitre sont les log RASTA-PLP et les PLP [52], les cepstres avec soustraction spectrale (CMS) [48] et les MFCC (voir paragraphe 2.3). Ces paramètres sont calculés tous les 10 ms sur une fenêtre d'analyse de 30 ms. L'ordre de l'analyse LPC a été fixé à 10. Ainsi le vecteur acoustique fourni à l'entrée de notre MLP est composé de 26 paramètres (les paramètres cepstraux, leurs dérivées premières ainsi que les dérivées première et seconde de l'énergie). Nous avons choisi de fournir 9 trames acoustiques de contexte au MLP (sachant que ce contexte donne généralement les meilleurs résultats), ce qui donne 234 entrées au MLP. Le lecteur trouvera une description détaillée de la définition des données d'entraînement et de test sélectionnées en annexe D. Pour rappel, les données d'entraînement consistent en 3737 phrases énoncées par 56 locuteurs (environ 9 heures de parole) et la partie test consiste en 144 phrases énoncées par 8 locuteurs (4 hommes et 4 femmes non présents dans la partie entraînement).

6.4.2 La procédure d'entraînement et de segmentation

Dans tous les cas, nous avons utilisé un entraînement Viterbi "embedded". Ce type d'entraînement requiert une segmentation phonétique de la base de données. Comme aucune segmentation n'est fournie avec BREF [2], nous avons généré une segmentation initiale pour initialiser le processus d'entraînement par une nouvelle méthode [71] basée sur l'utilisation d'un synthétiseur vocal de bonne qualité (MBROLA [36]). Ce synthétiseur est en fait utilisé pour générer (à partir de la transcription phonétique) un signal de parole synthétique qui est aligné sur le signal de parole réel. Ainsi le processus d'alignement est uniquement réduit à un simple algorithme DTW ("Dynamic Time Warping"). L'ensemble du processus est décrit sur la figure 6.3; le lecteur pourra trouver des renseignements complémentaires dans [30, 71].

En comparaison avec les méthodes classiques HMMs, cette méthode perd apparemment l'indépendance par rapport au locuteur puisqu'une seule voix est utilisée pour générer le signal acoustique. Mais la connaissance précise des limites entre les phonèmes suffit largement à contrebalancer cet effet. En pratique, nous avons observé que l'utilisation de deux voix distinctes (une pour les femmes et une pour les hommes) permettait d'obtenir une bonne segmentation. Fabrice Malfrère et Thierry Dutoit [71] ont rapporté un taux d'erreur

2. Et que le laboratoire ne possédait alors aucun système de reconnaissance en français.

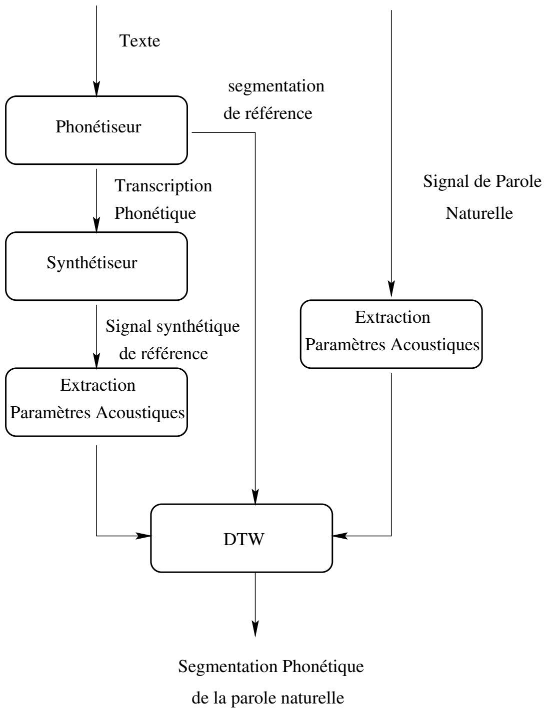

FIG. 6.3 - *Alignement Phonétique avec un TTS.*

de segmentation d'environ 8% (si on considère qu'une déviation temporelle de plus de 50 ms implique que le segment est faux). L'avantage principal de cette méthode réside dans le fait qu'aucune base d'entraînement n'est requise. Il suffit d'avoir la transcription phonétique. La segmentation obtenue par cet algorithme est indubitablement bien plus réaliste qu'une segmentation linéaire (les phrases de BREF sont en moyenne assez longues). Il y a fort à parier que le processus d'entraînement basé sur une segmentation linéaire aurait eu de grandes difficultés à converger. Cette segmentation initiale (facilement et rapidement obtenue) a été utilisée pour initialiser le processus d'entraînement. Nous avons entraîné dans un premier temps un système HMM à densités d'observations continues; et après quelques itérations, une assez bonne segmentation a été obtenue. Le système à densités d'observations continues utilise des matrices de covariances diagonales et le nombre de gaussiennes par état a été fixé à 16. Ce système a donc été utilisé pour obtenir une segmentation initiale qui a servi à l'entraînement d'un MLP. Comme le prouvent les taux de reconnaissance au niveau de la trame dans le tableau 6.1, la procédure suivie pour cette base de données nous a permis d'entraîner rapidement un bon système en français.

	Trames	log-RASTA	PLP	CMS	MFCC
Train	2400000	79.6%	82.0%	82.4%	82.4%
Cross	270000	77.0%	80.2%	79.6%	79.8%

TAB. 6.1 - *Taux de reconnaissance sur BREF au niveau de la trame acoustique avec un système hybride HMM/ANN entraîné sur des paramètres log-RASTA-PLP, PLP, CMS et MFCC.*

Ainsi l'utilisation d'un synthétiseur vocal de bonne qualité est un moyen efficace de générer une première segmentation d'une base de données qui peut être utilisée comme segmentation initiale dans le processus d'entraînement des systèmes de reconnaissance [30].

6.4.3 Reconnaissance de phonèmes sur BREF-80

Nous avons réalisé une expérience de reconnaissance de phonèmes sur cette base de données particulière afin de tester notre modèle hybride. Le système de base utilise 35 phonèmes indépendants du contexte qui sont décrits dans le tableau en annexe A (SAMPA [70]). Chaque modèle HMM est composé de 3 états identiquement distribués.

Le tableau 6.2 montre que les taux de reconnaissance obtenus sont assez impressionnants. Un taux de reconnaissance de 75.0% a été observé, ce qui correspond à notre connaissance à l'un des meilleurs taux jamais publiés sur cette base de données [43].

log-RASTA	PLP	CMS	MFCC
69.2%	72.9%	63.8%	75.0%

TAB. 6.2 - *Taux de reconnaissance de phonèmes utilisant un modèle hybride HMM/MLP et différents types de paramètres.*

6.4.4 Reconnaissance de la parole continue

Le système de base étant performant au niveau de la trame acoustique et du phonème, nous avons décidé de le tester au niveau du mot. Nous avons utilisé 4 dictionnaires différents :

- Un dictionnaire de 1.000 mots contenant les mots uniquement présents dans les phrases de tests.

- Deux dictionnaires de 3.000 et 13.000 mots contenant les mots du test set complétés avec les mots les plus courants
- Un dictionnaire de 65.000 mots.

Ces dictionnaires ont été obtenus par l'intermédiaire d'un phonétiseur par règles. Les résultats obtenus avec notre système hybride HMM/ANN de base sont rapportés dans les tableaux 6.3 et 6.4. Il est à noter que nous avons utilisé deux modèles de langage (bigramme et trigramme) estimés sur les textes du journal "Le Monde" (années 90-91-92[3], soit environ 80 millions de mots) en utilisant les outils de modélisation du CMU-Cambridge [23]. Nous avons pris garde à ce que les textes des données de tests ne figurent pas dans les textes utilisés pour la construction des modèles de langage. Les perplexités respectives de ces modèles de langage sont de 151.6 pour le bigramme et 94.4 pour le trigramme. Pour le dictionnaire de 65000 mots, le taux de mots hors vocabulaire (OOVs) est 1.49%. Nous ne rapportons ici que les meilleurs résultats qui ont été obtenus au niveau du mot, c'est pourquoi uniquement les résultats avec les paramètres acoustiques PLP sont donnés. Nous décrirons l'ensemble des résultats obtenus avec les deux modèles de langage décrit dans ce paragraphe (bigramme perplexité 151.6 et trigramme perplexité 94.4).

Il faut noter que les taux d'erreur reportés ici ne tiennent pas compte des erreurs par homophonie entre deux mots. Ainsi, une erreur du type "troyes" confondu avec "trois" n'est pas compté comme erreur dans les taux reportés dans les tableaux 6.3 et 6.4[4].

Dictionnaire	Ins.	Sup.	Subs.	Taux Erreur
1K	2.9%	5.0%	12.9%	20.8%
3K	3.4%	5.2%	14.0%	22.5%
13K	3.1%	6.2%	17.4%	26.7%
65K	2.7%	6.3%	18.9%	28.0%

TAB. 6.3 - *Taux d'erreur au niveau du mot pour des vocabulaires de 1K, 3K, 13K, 65K avec un modèle hybride HMM/ANN et un modèle de langage bigramme estimé sur les articles du journal "Le Monde" (perplexité 151.6).*

3. Ces années correspondent à la période d'enregistrement du corpus.
4. Ce choix a été fait pour pouvoir effectuer des comparaisons avec d'autres systèmes.

Dictionnaire	Ins.	Sup.	Subs.	Taux d'erreur
1K	2.3%	4.1%	9.5%	15.9%
3K	3.0%	4.6%	13.8%	19.8%
13K	3.1%	5.6%	15.3%	24.1%
65K	2.5%	6.0%	16.5%	25.0%

TAB. 6.4 - *Taux d'erreur au niveau du mot pour des vocabulaires de 1K, 3K, 13K, 65K avec un modèle hybride HMM/ANN et un modèle de langage trigramme estimé sur les textes du monde (perplexité 94.4).*

6.5 Conclusion

Nous avons décrit dans ce chapitre l'algorithme de reconnaissance de la parole continue grand vocabulaire développé par Renals. Cet algorithme utilise une programmation A* classique en recourant une approximation du critère A* usuel garantissant l'optimalité du processus de recherche de la meilleure séquence de mots. Le calcul du score de chaque hypothèse ne nécessite aucun "lookahead" et a prouvé son efficacité (du moins pour la reconnaissance de grands vocabulaires) si il est couplé à des techniques d'élagage évoluées. Cet algorithme a d'ailleurs été utilisé pour tester notre système de reconnaissance de la parole continue en français sur la base de données BREF-80. Les résultats obtenus sont les premiers avec un modèle hybride HMM/ANN sur cette base de données. Il est à noter que nous avons été confrontés à de nombreux problèmes dus au langage utilisé (le français). Les systèmes de reconnaissance souffrent beaucoup des liaisons entre les mots (ce qui arrive fréquemment en français) et du fait que certains phonèmes sont par moment prononcés par certains locuteurs et omis par d'autres. Ce problème a été résolu en utilisant pour chaque mot des prononciations multiples (modélisant au mieux les différentes prononciations dans le cas d'omission de phonèmes ou de liaisons). De plus en français, les homophones sont un réel problème. Par exemple les phrases "c'est à peine" et "sept à peine", possédant des transcriptions phonétiques identiques et étant toutes les deux sémantiquement valides, occasionneraient des erreurs lors du calcul des taux de reconnaissance. Gauvain [43] rapporte un taux d'homophone de 30% en français ce qui est largement supérieur à par exemple celui de l'anglais (3% pour TIMIT). Une étude détaillée des erreurs commises lors de la reconnaissance montre que les principales erreurs sont dues à ce type de problème et que l'acoustique des phrases reconnues est vraiment très bonne.

Chapitre 7

Contexte et suppression de trames

7.1 Introduction

Nous avons vu dans les chapitres précédents que la technologie la plus couramment utilisée en reconnaissance de la parole est basée sur les modèles de Markov cachés (HMM). Néanmoins, la mise en oeuvre de ces modèles nécessite quelques hypothèses simplificatrices dont nous avons déjà largement parlé. Les modèles hybrides HMM/ANN tentent de limiter le nombre des hypothèses nécessaires à l'utilisation des modèles HMMs et se sont avérés au moins aussi efficaces que les HMMs les plus sophistiqués, voire même supérieurs. L'un des avantages des modèles hybrides est leur capacité d'introduire une information contextuelle. De nombreux laboratoires ont montré qu'il existait un optimum relatif du taux de reconnaissance pour 9 trames de contexte à l'entrée du MLP (avec les dérivées premières). Initialement les dérivées des vecteurs acoustiques ont été introduites pour ajouter une information contextuelle au modèle de Markov HMM classique [41, 67]. Cette information s'accompagnait d'ailleurs d'une amélioration significative des taux de reconnaissance sur la plupart des bases de données utilisées en reconnaissance de la parole.

Les réseaux de neurones permettent maintenant d'introduire plus directement cette information contextuelle. Nous étudierons dans ce chapitre l'influence de cette information contextuelle sur les taux de reconnaissance. De plus, il a déjà été montré qu'il était possible (du moins pour les HMMs classiques) de supprimer des trames acoustiques sans dégrader de manière significative les taux de reconnaissance [22, 75, 113]. Nous étudierons donc l'influence de la suppression automatique de trames acoustiques sur les taux d'erreur pour les modèles hybrides. Nous espérons ainsi pouvoir accélérer le processus de reconnaissance de la parole par modèle hybride qui est bien entendu fonction de la complexité du système et du nombre de trames à traiter. Il sera possible ainsi de répondre à une des principales critiques sur l'utilisation des modèles hybrides en reconnaissance de la parole: le temps de calcul.

7.2 Influence du contexte

Classiquement en reconnaissance de la parole, la plupart des laboratoires utilisent les dérivées des vecteurs acoustiques pour ajouter une certaine information contextuelle aux modèles utilisés. Ainsi, il est très courant pour des modèles HMMs classiques (discrets ou continus) d'utiliser en plus des coefficients cepstraux, leurs dérivées premières et secondes ainsi que les dérivées première et seconde de l'énergie. Même pour les modèles hybrides, il est assez courant d'utiliser 9 trames de contexte à l'entrée du MLP complétées par leurs dérivées premières et les dérivées première et seconde de l'énergie. Ainsi les systèmes HMMs discrets ou continus utilisent un vecteur de 38 composantes (cepstres, Δcepstres, $\Delta\Delta$cepstres, Δénergie et $\Delta\Delta$énergie). Les modèles hybrides utilisent eux couramment un vecteur acoustique de 26 composantes et un contexte de 9 trames acoustiques (cepstres, Δcepstres, Δénergie et $\Delta\Delta$énergie).

Dans le cas des HMMs classiques, l'ajout des dérivées constitue une méthode simple de prise en compte de l'information contextuelle et de la dynamique du signal. En revanche, pour les modèles hybrides il est possible d'utiliser plus directement l'information contextuelle. Nous allons montrer qu'il est possible d'utiliser un contexte plus large sans dérivées pour obtenir des résultats semblables à ceux obtenus avec un contexte de 9 trames acoustiques plus les dérivées.

Le calcul des dérivées[1] est effectué par combinaison linéaire des trames voisines. Ainsi le calcul de la dérivée première de la trame X_i notée ΔX_i est effectué par combinaison linéaire des trames voisines selon la formule[2]:

$$\Delta X_i = \sum_{k=-2}^{k=2} kX_{i+k} = -2X_{i-2} - X_{i-1} + 0.X_i + X_{i+1} + 2X_{i+2} \quad (7.1)$$

Nous allons dans cette expérience comparer deux réseaux sur les bases de données PHONEBOOK et BREF-80[3]:

- le réseau de base, utilisé par la plupart des laboratoires de recherche en reconnaissance de la parole, possède 9 trames de contexte basées sur un vecteur cepstral de 26 composantes composé des coefficients cepstraux (log RASTA-PLP) ainsi que leurs dérivées premières complétées par les dérivées première et seconde de l'énergie.
- Un réseau utilisant 15 trames de contexte sur un vecteur cepstral de 12 composantes sans aucune dérivée.

Nous couvrons, en utilisant les deux bases de données citées précédemment, deux types de reconnaissance (isolée et continue) et deux types de canaux de transmission (téléphone et microphone). Les résultats obtenus sur chacun des corpus sont reportés dans les tableaux 7.1 et 7.2.

1. Dans les cas HMMs classiques et modèles hybrides.
2. Dans toutes les expériences reportées dans ce travail.
3. Voit Annexe D.

	75 mots	600 mots	8000 mots
9 trames + dérivées	2.7%	7.0%	22.0%
15 trames sans dérivées	2.5%	7.8%	23.0%

TAB. 7.1 - *Taux d'erreur sur Phonebook dictionnaires 75, 600 et 8000 mots pour deux systèmes hybrides utilisant des coefficients log RASTA-PLP. Le premier utilise 9 trames de contexte + les dérivées, le second 15 trames de contexte sans aucune dérivée.*

1K	Ins.	Sup.	Subs.	Taux d'erreur
9 trames + dérivées	2.7%	4.1%	9.3%	16.2%
15 trames sans dérivées	3.6%	3.5%	10.4%	17.5%
3K	Ins.	Del.	Subs.	Taux d'erreur
9 trames + dérivées	3.2%	4.1%	11.6%	18.9%
15 trames sans dérivées	3.3%	3.9%	12.3%	19.5%
13K	Ins.	Del.	Subs.	Taux d'erreur
9 trames + dérivées	3.0%	5.3%	15.0%	23.4%
15 trames sans dérivées	2.9%	5.2%	15.6%	23.8%
65K	Ins.	Del.	Subs.	Taux d'erreur
9 trames + dérivées	2.5%	5.7%	15.8%	24.0%
15 trames sans dérivées	2.5%	5.9%	16.4%	24.8%

TAB. 7.2 - *Taux d'erreur sur BREF-80 dictionnaires 1k, 3k, 13k et 65k pour deux systèmes hybrides utilisant des coefficients log RASTA-PLP.*

Nous pouvons donc remarquer qu'il est possible d'avoir recours à un réseau avec un contexte plus important, à la place des réseaux classiques utilisés par la plupart des laboratoires en reconnaissance de la parole (9 trames de contexte et les dérivées), sans observer de réelles détériorations des taux de reconnaissance. Le fait de ne pas calculer les dérivées permet de gagner un peu de temps lors du processus de reconnaissance. Il est à noter que nous n'avons utilisé aucune information sur l'énergie du signal pour le réseau sans dérivées. Cela peut expliquer en partie les légères dégradations des résultats observées sur les différents corpus.

7.3 Suppression de trames acoustiques

Nous étudions dans ce paragraphe l'influence de la suppression de trames sur le taux de reconnaissance. Cette idée fut introduite pour la première fois par Tapper [113]. D'autres chercheurs ont aussi montré qu'il était possible de supprimer une partie de la phrase sans dégrader (et même quelques fois améliorer) le taux de reconnaissance [22, 75]. Nous allons voir dans ce paragraphe si les mêmes conclusions peuvent être tirées avec le système hybride. Les trames sont supprimées automatiquement afin d'observer l'influence sur le taux de reconnaissance, les réseaux de base sont du même type que ceux utilisés précédemment (voir tableau 7.3 pour les résultats sur le corpus PHONEBOOK et la figure 7.1 pour le corpus BREF-80.). Les 1/2, 1/3, ... correspondent au nombre de trames supprimées. Par exemple, 1/2 signifie qu'une trame sur deux a été supprimée de manière automatique.

Dictionnaire	Sans Sup.	1/2	2/3	3/4	4/5
75 mots	2.7%	2.7%	2.7%	2.9%	2.7%
600 mots	7.4%	7.5%	7.4%	8.1%	8.2%
8000 mots	22.0%	22.1%	22.3%	23.3%	24.5%

TAB. 7.3 - *Taux d'erreur sur Phonebook (dictionnaires 75, 600 et 8000 mots) avec différents nombres de trames supprimées automatiquement et des coefficients log RASTA-PLP.*

Selon les résultats visibles sur les figures 7.1 et 7.2, il est possible de conclure :

- la suppression d'environ 1 trame sur deux et même 2 sur 3 dégrade légèrement le taux de reconnaissance. Le même type d'observation avait été fait par Le Cerf et Murveit pour les HMMs traditionnels dans [75].
- La suppression de trames n'a jamais impliqué (pour cette expérience) une amélioration du taux de reconnaissance. Ce qui était somme toute assez prévisible car la suppression automatique de trames implique nécessairement une perte d'informations.

Remarquons quand même (voir figure 7.1) que l'influence est beaucoup plus marquée dans le cas de la reconnaissance de la parole continue grand vocabulaire. Dans ce cas précis, le gain de temps observé avec la suppression du calcul des probabilités est négligeable. En fait, pour des vocabulaires de taille importante, c'est le décodage qui prend le plus de temps et non l'estimation des probabilités.

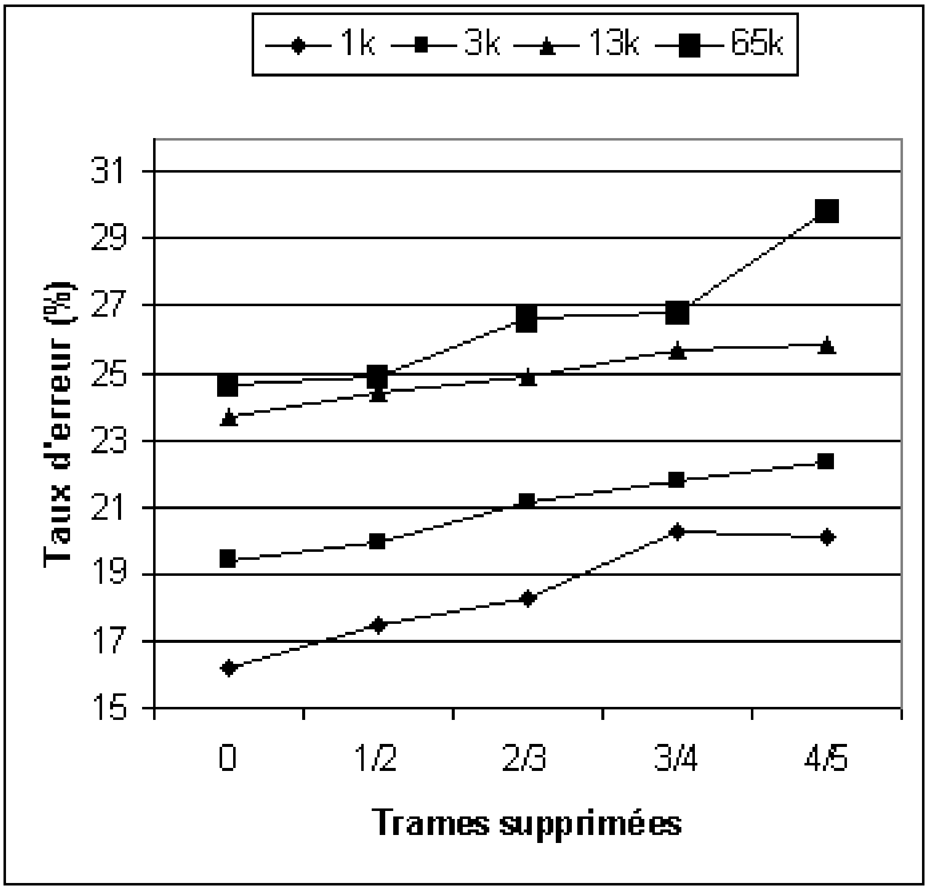

FIG. 7.1 - *Taux d'erreur sur BREF pour différents vocabulaires et nombres de trames supprimées.*

Il est donc possible d'accélérer le processus de reconnaissance en supprimant automatiquement le calcul des probabilités pour 1 trame sur 2 et même 2 trames sur 3, particulièrement dans le cas d'une reconnaissance de mots isolés. Au lieu d'éliminer de manière automatique les trames, voici quelques méthodes qui pourraient être utilisées pour déterminer plus précisemment quelles trames il faut supprimer.

– Calculer la corrélation entre les trames afin de supprimer celles correspondant aux régions stationnaires. La distance Euclidienne a déjà été utilisée pour évaluer cette corrélation mais aucune amélioration du taux de reconnaissance n'a été observée.

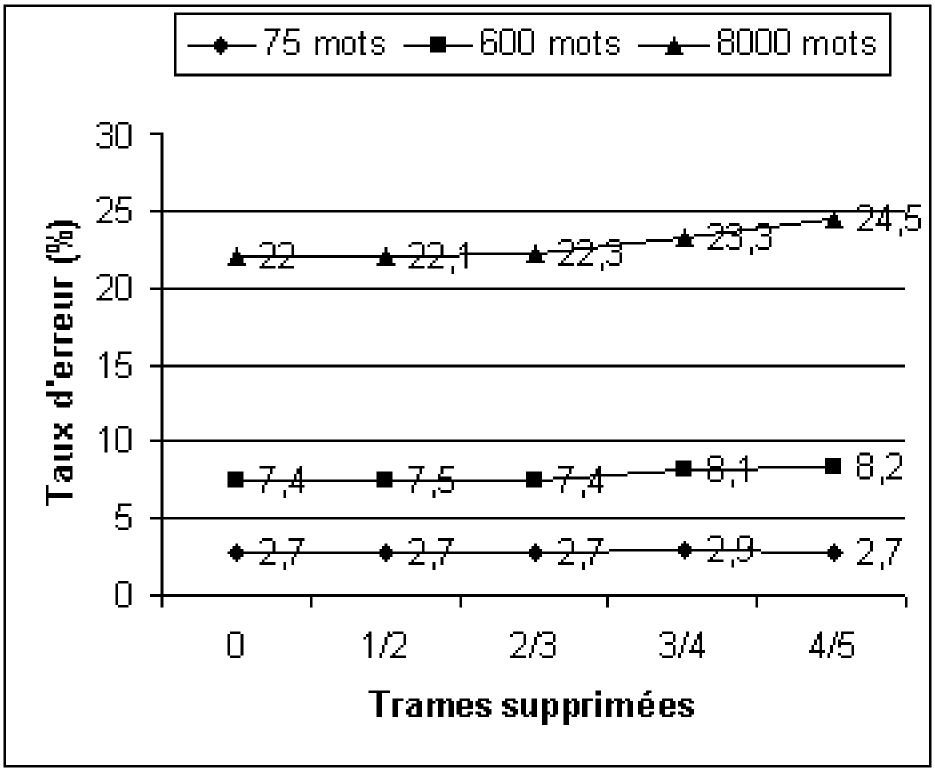

FIG. 7.2 - *Taux d'erreur sur Phonebook pour différents vocabulaires et nombres de trames supprimées.*

– Dans [22], le critère utilisé pour supprimer des trames est le suivant : Si la norme de la dérivée des paramètres cepstraux pour une trame est inférieure à un seuil fixé, la trame est éliminée.

Ces méthodes permettraient d'améliorer sensiblement la vitesse du processus de reconnaissance mais malheureusement pas le taux de reconnaissance. C'est pourquoi elles n'ont pas fait l'objet d'investigations supplémentaires.

7.4 Conclusion

Il est possible de remplacer le calcul des dérivées dans le cas de modèles hybrides par un contexte acoustique plus large sans observer de détériorations des taux de reconnaissance. De plus, il est possible d'accélérer fortement le processus de reconnaissance (particulièrement en mode mots isolés) en supprimant de manière automatique le calcul des probabilités pour 1 trame sur

2 et même 2 trames sur 3 sans observer de dégradation au niveau des taux de reconnaissance. Cette technique de suppression automatique des trames est d'ailleurs employée pour les démonstrations du laboratoire TCTS, permettant d'utiliser les modèles hybrides en temps réel pour un vocabulaire d'environ 600 mots sur un simple PC (133 MHz). Notons que cette technique de suppression de trames peut aussi être appliquée lors de l'entraînement des réseaux MLPs. Des expériences ont montré qu'il est possible d'accélérer fortement le processus d'entraînement des MLPs sans détérioration dans l'estimation des paramètres des modèles hybrides. Cette technique répond à l'un des principaux désavantages du modèle hybride : le temps de calcul qui lui est associé.

Chapitre 8

Modèles dépendants du contexte

8.1 Introduction

Les réseaux de neurones ont démontré leur capacité à estimer, s'ils sont entraînés comme classificateur, la probabilité a posteriori d'une classe étant donné l'entrée fournie au réseau. Ces 10 dernières années [4, 12, 32, 105], de nombreux chercheurs ont montré l'intérêt d'utiliser des modèles dépendants du contexte pour modéliser au mieux la variabilité du signal de parole. Ces modèles sont plus adaptés parce que les effets de coarticulation sont explicitement modélisés, produisant ainsi un recouvrement moins important des densités de probabilités pour les différentes classes phonétiques. Il est donc utile de s'intéresser au couplage de ces deux méthodes pour améliorer les performances des systèmes de reconnaissance. Le problème lié aux MLPs vient du nombre de paramètres à estimer. En effet, en moyenne on utilise au minimum 1000 unités pour modéliser des phonèmes dépendants du contexte du type diphone et jusqu'à plusieurs milliers pour des triphones. Prenons l'exemple de 5000 sorties et 1000 noeuds cachés, il faudrait estimer 5 millions de paramètres. Ceci est bien entendu difficile compte tenu de la taille des bases de données utilisées pour l'entraînement (voir Annexe D, typiquement quelques millions de trames acoustiques) et du temps nécessaire pour l'entraînement du réseau par un algorithme de "rétro-propagation" classique[1]. Nous allons décrire dans ce chapitre les développements qui ont été menés sur ce problème et les solutions auxquelles nous sommes arrivés. Nous décrirons l'état de l'art des méthodes utilisées pour modéliser les phonèmes dépendants du contexte, puis nous étendrons ces méthodes au cas particulier des modèles hybrides, enfin nous développerons une méthode basée sur une quantification des transitions entre les phonèmes. Nous montrerons que ces méthodes permettent d'obtenir une amélioration sensible des taux de reconnaissance sans augmentation importante du temps de calcul.

8.2 Modèles de phonèmes dépendants du contexte et HMMs standards

L'un des principaux problèmes liés à l'utilisation de modèles dépendants du contexte est le nombre de paramètres à estimer. Nous avons décidé dans un premier temps de tester les méthodes de bases utilisées en reconnaissance de la parole pour les HMMs classiques et de les étendre aux modèles hybrides. Les tests effectués pour l'étude de ces modèles ont été réalisés sur le corpus PHONEBOOK. Nous nous sommes, dans un premier temps, limités

1. Notons que le même problème existe pour les HMMs standards à une moindre échelle.

à l'étude des diphones dans un contexte gauche. Rappelons qu'un diphone est un modèle de phonème prenant en considération les contextes gauches ou droits immédiats. Toutes les méthodes décrites dans ce chapitre peuvent évidemment être étendues aux triphones (un triphone est un modèle prenant en considération les contextes gauches et droits).

Sachant que pour ce corpus, il y a 47 phonèmes différents pour décrire l'ensemble des mots du vocabulaire, il existe a priori un ensemble de $47 * 47$ diphones possibles. En fait, après un simple comptage sur les données d'entraînement de la base de données, nous avons dénombré environ 1100 unités.

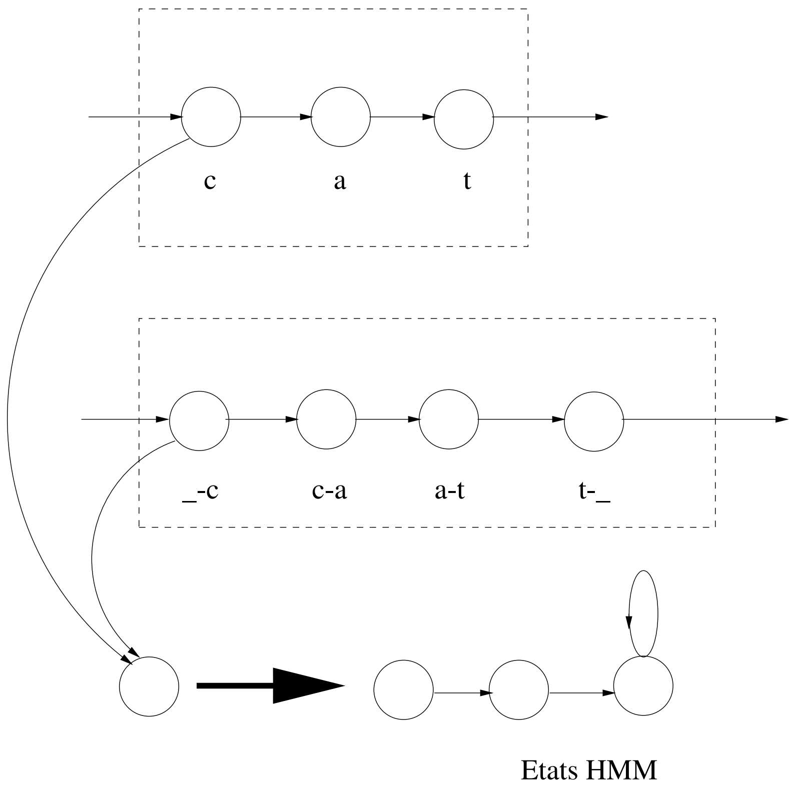

FIG. 8.1 - *Définition sur un exemple précis (le mot "cat") des modèles indépendants et dépendants du contexte.*

La figure 8.1 permet de visualiser plus précisément en quoi consistent ces modèles. Dans le cas d'une modélisation CI le mot "cat" est construit par une concaténation des phonèmes "c", "a", et "t" [2]. Dans le cas d'une modélisation

2. Une définition de l'alphabet phonétique utilisé est visible en Annexe A.

dépendant du contexte gauche, le mot "cat" est construit par concaténation des différents modèles "_-c" "c-a", "a-t", et "t-_" (rappelons que le phonème "_" correspond au silence). Ces phonèmes correspondent respectivement au phonème "c" dans le contexte silence suivi du phonème "a" dans le contexte "c" ...

L'entraînement de ces modèles a été effectué de la manière suivante :

– Initialisation de la segmentation en diphones à partir de celle des phonèmes indépendants du contexte. Il suffit d'assigner au label correspondant au phonème Y celui correspondant au diphone $X - Y$ [3] dans le cas particulier (phonème dans un contexte gauche) de la segmentation présentée sur la figure 8.2. Durant l'entraînement, les paramètres du modèle de diphone $X - Y$ sont estimés directement à partir des données disponibles pour le phonème Y dans ce contexte particulier X.

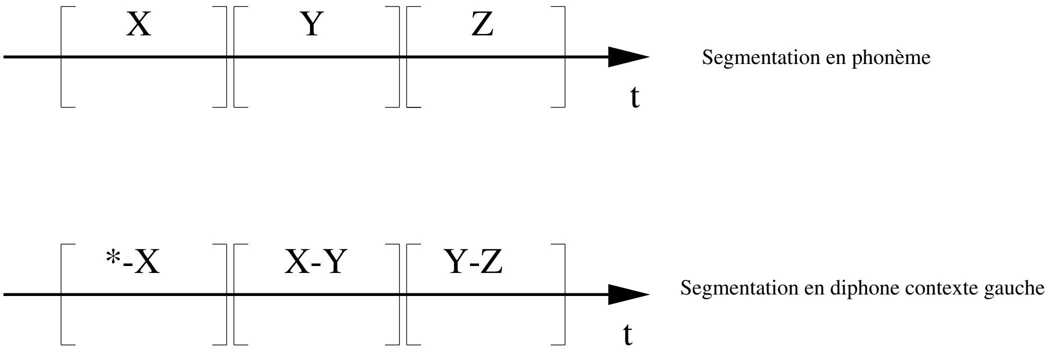

FIG. 8.2 - *Passage d'une segmentation phonèmes indépendants du contexte à phonèmes dépendants du contexte.*

– Entraînement des modèles basé sur l'algorithme E-M (Expectation-Maximisation) et itération du processus jusqu'à convergence.

En général, afin de réduire le nombre de paramètres à estimer, on utilise des modèles de phonèmes dépendants du contexte généralisés. C'est à dire qu'une quantification est réalisée pour regrouper les unités possédant des caractéristiques similaires. Il est possible de réduire le nombre de paramètres à estimer en partageant les distributions des différents modèles, c'est-à-dire en regroupant, sous un même groupe, les distributions ayant des propriétés

3. "X-Y" signifie phonème "Y" dans le contexte gauche "X".

similaires. La méthode utilisée pour effectuer cette quantification est assez couramment employée en reconnaissance de la parole.

1. A partir d'une segmentation des données d'entraînement en diphones, entraîner une gaussienne pour chaque état,

2. Lorsque ces gaussiennes sont entraînées, il suffit pour les quantifier de regrouper celles pour lesquelles la perte d'information liée à la quantification est minimale. Nous utilisons pour cela une distance[4] entre deux gaussiennes définie selon :

$$d[i,j] = [\frac{1}{V}\sum_{k=0}^{V}\frac{(\mu_{ik}-\mu_{jk})^2}{(\sigma_{ik}.\sigma jk)^2}]^{\frac{1}{2}} \tag{8.1}$$

   où μ_i et σ_i sont classiquement le vecteur moyen et le vecteur variance associés à la gaussienne i.

Il est ainsi possible de quantifier facilement l'ensemble des diphones sans connaissance a priori sur les phonèmes et leurs caractéristiques. Une autre possibilité souvent utilisée dans la littérature consiste à utiliser des arbres de décision pour classer les diphones. La méthode décrite dans ce paragraphe ne nécessitant pas de connaissances linguistiques, et les résultats associés étant assez bons, nous avons décidé de l'utiliser dans la plupart de nos expériences sur les modèles dépendants du contexte. Un exemple de classification obtenue sur la base de données PHONEBOOK peut être observé sur le tableau 8.1. Le lecteur appréciera facilement la logique de la classification effectuée. La première ligne nous permet d'observer que l'algorithme de quantification a regroupé les diphones du type $\{iy, ey, oy\} - er$ sous une même classe. Or d'un point de vue purement linguistique, cette classification a un sens puisque $\{iy, ey, oy\}$ sont trois voyelles proches. Il apparaît donc logique de regrouper en une seule classe les trois diphones $iy - er$, $ey - er$ et $oy - er$.

L'entraînement des différents modèles associés aux classes s'effectue de manière classique par l'algorithme E-M. Nous avons testé plusieurs valeurs du nombre de classes utilisées pour la classification (de 600 à 1000). L'entraînement a été réalisé sur toutes les données d'entraînement (79900 phrases soit 11 millions de trames acoustiques). En effet, nous devions nous assurer que tous les contextes possibles définis précédemment se trouvaient en nombre suffisant dans la base d'entraînement. Le corpus PHONEBOOK a d'ailleurs été utilisé parce que nous étions assurés d'y trouver l'ensemble des phonèmes dans différents contextes. Les résultats sont largement meilleurs

4. Distance de Mahalanobis.

Classes	diphones
1	iy-er ey-er oy-er
2	m-pcl n-dcl m-bcl ng-pcl m-tcl
3	ch-iy t-iy g-iy jh-iy d-iy k-iy hh-iy
4	d-ow t-ow y-ao jh-oy ch-ao
5	k-bcl t-bcl k-pcl p-gcl k-gcl p-pcl
6	ah-dh ah-pcl ao-pcl aw-bcl aa-pcl aa-v aa-bcl ao-bcl
7	n-ih m-ey m-ae n-ae n-ey n-eh
..	..

TAB. 8.1 - *Classification des différents diphones par la méthode basée sur les gaussiennes*

que ceux obtenus avec les HMMs classiques et les modèles de phonèmes indépendants du contexte (voir tableau 8.2). Ils sont maintenant comparables et même meilleurs que ceux obtenus avec notre système hybride de base. Les paramètres acoustiques utilisés sont les log RASTA-PLP (voir paragraphe 2.3.3). Le système multigaussien utilise des matrices de covariances diagonales et le nombre de gaussiennes par état a été fixé à 16 (généralement un bon compromis entre la charge de calcul et la validité des modèles).

Nb classes	75 mots	600 mots	8000 mots
600	2.2%	6.4%	18.7%
700	2.3%	6.1%	18.2%
800	1.9%	6.0%	18.1%
1000	1.9%	5.6%	16.8%
tous	1.8%	5.2%	15.4%
mlp de base	2.7%	7.0%	22.0%
HMM CI	8.2%	13.4%	38.0%
R. T. E.	33%	25.7%	30%

TAB. 8.2 - *Taux d'erreur au niveau du mot pour un dictionnaire de 75, 600 et 8000 mots et les différents nombres de classes. Paramètres log Rasta-PLP, modèle HMM, 16 gaussiennes par état, matrice de covariance diagonale.*

Nous avons donc montré qu'il était possible, en utilisant des modèles tenant compte du contexte, d'obtenir de très bons taux de reconnaissance même comparés aux modèles hybrides de base. Néanmoins, il faut noter que le temps de calcul associé et le nombre de paramètres utilisés par rapport au modèle CI sont beaucoup plus importants.

8.3 Extension de la méthode

Nous allons présenter dans le reste de ce chapitre les différentes méthodes utilisant à la fois les modèles de phonèmes dépendants du contexte et les modèles hybrides en reconnaissance de la parole. Le problème revient à estimer la probabilité $p(X|q_k, c_j)$ où q_k et c_j sont respectivement l'état HMM et le contexte associé. Nous allons tester les différentes méthodes qui ont été utilisées pour estimer cette probabilité. Un des moyens les plus directs pour l'estimation de cette probabilité consiste à utiliser simplement les unités phonétiques pré-définies pour les modèles CD-HMM continus du paragraphe précédent et d'entraîner un réseau qui estime directement la probabilité $p(q_k, c_j|X)$ pour chacune des classes[5]. Ainsi, en utilisant la segmentation en diphones généralisés, nous avons entraîné un réseau avec N sorties où N est égal au nombre de classes utilisées pour modéliser les prononciations des différents mots du vocabulaire de PHONEBOOK. Ainsi le réseau possède pour l'expérience qui a été menée, 600 sorties correspondant aux 600 classes de diphones généralisés (ou 1098 dans le cas où aucune quantification n'est réalisée). Un contexte classique de 9 trames de vecteurs acoustiques et une couche cachée de 1000 noeuds ont été choisis. Les résultats obtenus avec ce type de réseau ont largement dépassé nos espérances puisque, comme on le constate sur le tableau 8.3, l'amélioration du taux de reconnaissance par rapport au modèle hybride de base, et même par rapport aux modèles dépendants du contexte HMM, est plus que conséquente.

Nb classes	75 mots	600 mots	8000 mots
CD-MLP-600	1.2%	4.1%	15.1%
CD-MLP-tous	0.9%	3.2%	13%
CI-MLP	2.7%	7.0%	22.0%
CD-HMM	1.8%	5.2%	15.4%
CD-HMM-600	2.2%	6.4%	18.7%

TAB. 8.3 - *Taux d'erreur au niveau du mot pour un dictionnaire de 75, 600 et 8000 mots et les différents nombres de classes. Paramètres log Rasta-PLP, MLP 234-1000-X, où X est le nombre de classes.*

Les résultats obtenus simplement avec un unique MLP estimant les probabilités $p(q_k, c_j|X)$ sont donc réellement impressionnants. Nous avons réussi à réduire le taux d'erreur de plus de 66%, 54% et 41% respectivement pour des vocabulaires de 75, 600 et 8000 mots par rapport au système hybride

5. $p(q_k, c_j|X)$ et $p(X|q_k, c_j)$ sont liées par la loi de Bayes (voir équation(8.4))

de base. De plus, le temps de calcul est certes plus important comparé au réseau de base CI, mais il reste encore largement inférieur au temps nécessaire à l'utilisation des phonèmes dépendants du contexte et des HMMs continus. Il est d'ailleurs possible d'appliquer les techniques de suppressions de trames acoustiques du chapitre 7 pour réduire encore davantage le temps de calcul. De même, en comparaison avec la méthode HMM continus et modèles dépendants du contexte développés dans le paragraphe précédent, nous observons une réduction significative des taux d'erreur (50% pour 75 mots, 38% pour 600 mots et 18% pour 8000 mots). Une des explications possibles des résultats obtenus par cette méthode est que le réseau MLP entraîné sur les classes estime de manière efficace la probabilité $p(q_k, c_j|X)$. En effet, il ne faut pas oublier que le réseau est discriminant au niveau de la trame acoustique, et donc probablement plus à même d'estimer cette probabilité que la modélisation multigaussienne.

8.4 La méthode proposée par ICSI

Hervé Bourlard [16, 18] a développé une méthode permettant de calculer, sans aucune simplification, les probabilités associées aux modèles dépendants du contexte à l'aide de plusieurs réseaux de tailles sensiblement égales à celle du MLP de base utilisé pour la modélisation des phonèmes indépendamment du contexte (CI). En notant c_j^l et c_l^r les contextes gauche et droit de l'état q_k, il est possible d'estimer $p(X|q_k, c_j^l, c_l^r)$ selon :

$$p(X|q_k, c_j^l, c_l^r) = \frac{p(q_k, c_j^l, c_l^r|X).P(X)}{p(q_k, c_j^l, c_l^r)} \tag{8.2}$$

ce qui peut encore s'écrire :

$$p(X|q_k, c_j^l, c_l^r) = \frac{p(c_j^l|q_k, c_l^r, X).P(c_l^r|q_k, X).P(q_k|X).P(X)}{P(c_j^l|q_k, c_l^r).P(c_l^r|q_k).P(q_k)} \tag{8.3}$$

Or chacune des probabilités présentes dans cette équation peut être estimée par l'intermédiaire de MLPs.

– $p(c_j^l|q_k, c_l^r, X)$ peut être estimé par le MLP de la figure 8.3 dans lequel les unités de sorties sont associées au phonème gauche du triphone et pour lequel l'entrée est constituée par le vecteur acoustique X, l'état courant et le contexte droit du triphone

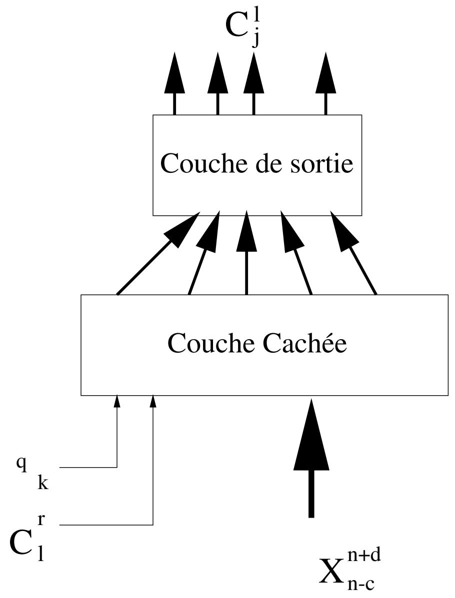

FIG. 8.3 - *Le réseau permettant l'estimation de* $p(c_j^l|q_k, c_l^r, X)$.

- $p(c_l^r|q_k, X)$ peut être estimé par le MLP de la figure 8.4 pour lequel la sortie est associée au phonème droit et pour lequel l'entrée est constituée par le vecteur acoustique X (et son contexte) et l'état courant.
- $P(q_k|X)$ peut être estimé par un MLP classique déjà décrit de nombreuses fois dans cette étude.
- $P(c_j^l|q_k, c_l^r)$, $P(c_l^r|q_k)$ et $P(q_k)$ peuvent être estimés par comptage sur les données d'entraînement.

Ainsi, en entraînant différents réseaux de neurones et en utilisant leurs sorties, il est possible d'estimer sans aucune hypothèse simplificatrice la probabilité $p(X|q_k, c_j^l, c_l^r)$ utilisée pour modéliser les probabilités des triphones fournies aux HMMs. Cette procédure permet de réduire le nombre de sorties (donc de paramètres) d'un unique MLP ($K * J * J$ où K est le nombre de CI phonèmes et J le nombre de contextes) en trois réseaux plus petits avec K, J et J sorties pour modéliser les triphones. Le problème avec cette méthode apparaît en phase de reconnaissance où le temps nécessaire au calcul de ces probabilités est beaucoup plus important. En effet, le calcul de $p(c_l^r|q_k, X)$ doit être effectué pour tous les états q_k. Le réseau doit donc être lancé K fois pour chaque trame acoustique pendant la phase de reconnaissance puisque la probabilité de sortie dépend d'une hypothèse sur la classe courante (q_k). Ainsi

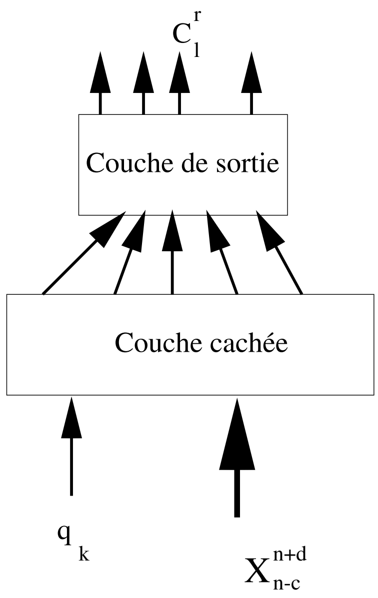

FIG. 8.4 - *Le réseau permettant l'estimation de $p(c_l^r|q_k, X)$.*

d'un point de vue pratique, la formulation précédente ne peut être appliquée directement. Une solution proposée par H. Bourlard consiste à effectuer une petite restriction sur la topologie du réseau de neurones permettant le précalcul des contributions de la partie phonétique contextuelle à la sortie du MLP. Ce calcul peut être effectué à la fin de l'entraînement et avant la reconnaissance. Il suffit de scinder la topologie du réseau afin que la couche cachée ne reçoive pas à l'entrée, à la fois, la contribution des données de parole et la contribution du contexte phonétique. Il est alors possible de calculer, pour toutes les combinaisons de contextes droits et gauches possibles, la contribution correspondante à la sortie. Ces contributions sont sauvegardées dans des tables. Lors de la reconnaissance, les valeurs résultant des données de la parole peuvent être calculées par une passe pour chaque trame de parole. Pour chaque modèle de triphone, ces contributions des données d'entrée peuvent être additionnées à la contribution contextuelle correspondante de la table. Ainsi après entraînement du réseau, une table de $K * J * J$ éléments est calculée en lançant la partie correspondant aux classes phonétiques dépendantes du contexte (sans entrées correspondant aux données de parole) $K * J * J$ fois pour toutes les unités de sorties possibles et pour toutes les combinaisons possibles des contextes phonétiques. Le chargement de cette table est négligeable par rapport au temps de calcul nécessaire pour l'ensemble des probabilités. Lors de la reconnaissance, il suffit de lancer les trois réseaux précédemment décrits une seule fois pour obtenir la contribution de la donnée présentée à l'entrée pour chaque sortie q_j. Pour chaque triphone, cette contribution doit être ajoutée à la contribution correspondant au contexte comme un biais dépendant du contexte.

8.5 Méthode proposée par Stanford

Horacio Franco [39] a aussi présenté une méthode visant à utiliser les modèles dépendants du contexte et les modèles hybrides. Cette méthode a été testée sur la base de données RESOURCE MANAGEMENT et une amélioration sensible du taux de reconnaissance a été observée (passage de 5.4% d'erreur à 4.7% sur feb89). Nous allons décrire dans ce paragraphe cette méthode et la procédure d'entraînement des différents réseaux utilisés. La méthode développée à Stanford est basée sur la décomposition suivante de la probabilité $P(X|q_k, c_j)$.

$$p(X|q_k, c_j) = \frac{P(q_k, c_j|X).p(X)}{P(q_k, c_j)} \tag{8.4}$$

soit en utilisant l'équation (2.15) :

$$p(X|q_k, c_j) = \frac{P(q_k|c_j, X).P(c_j|X).p(X)}{P(q_k, c_j)} \tag{8.5}$$

Le principal problème vient toujours de l'estimation de $p(q_k|X, c_j)$. Franco propose dans [39] de calculer cette probabilité en réduisant les données fournies à l'entrée du MLP à celles correspondant au contexte c_j. Ainsi, si M est le nombre total de classes, cette application utilise une série de M MLPs similaires à ceux utilisés dans le cas des phonèmes indépendants du contexte, excepté le fait que chaque MLP est entraîné uniquement avec des données correspondant à un contexte fixé c_j. Cette méthode possède l'avantage d'utiliser la même architecture et le même processus d'entraînement que les réseaux CI. Chaque MLP réalise une classification plus simple que dans le cas CI, car pour un contexte donné, le recouvrement des différentes classes phonétiques est moins important, ce qui implique une erreur de classification moindre. Comme pour la méthode développée à l'ICSI, le facteur $p(c_j|X)$ de (8.5) peut être estimé par l'intermédiaire d'un MLP classique avec les sorties correspondant aux classes contextuelles au lieu des phonèmes.

Les différents CD-MLP ont leurs poids initialisés avec ceux du réseau CI, puis chaque vecteur d'entraînement (vecteurs acoustiques avec leurs étiquettes phonétiques et la classe phonétique associée) est présenté uniquement au réseau correspondant. En supposant qu'il y a assez de données d'entraînement et que le nombre d'itérations est suffisant, chaque réseau contextuel devrait "asymptotiquement" converger vers la probabilité $p(q_k|X, c_j)$. L'initialisation des poids des MLPs, avec ceux du réseau CI, permet de réaliser une sorte de lissage non linéaire entre les paramètres associés aux modèles indépendants du contexte et les paramètres associés aux modèles dépendants du contexte. De plus, le temps nécessaire à l'entraînement devrait être considérablement réduit par rapport à une initialisation au hasard des poids. Pour cette méthode, il est aussi possible de définir de nombreuses classes phonétiques basées par exemple sur la catégorie des phonèmes voisins ou sur le lieu de l'articulation. Lors de la reconnaissance, les premiers états des HMMs de phonèmes sont associés aux sorties du MLP selon la classe à laquelle appartient le phonème gauche. De même, les derniers états des modèles HMMs sont associés aux sorties du réseau dépendant du contexte selon la classe phonétique à laquelle appartient le phonème droit. La reconnaissance est effectuée par l'algorithme Viterbi qui requiert le calcul des probabilités d'observation associées à chaque état HMM. Pour cela, il est nécessaire de convertir les probabilités a posteriori issues des réseaux en utilisant la formule suivante [39] :

$$P(X|q_k, c_j) = P(q_k|X, c_j).\{\frac{\alpha_j^k}{P(q_k)} + \frac{(1-\alpha_j^k).p(c_j|X)}{p(q_k|c_j)p(c_j)}\} \quad (8.6)$$

avec

$\alpha_j^k = \frac{N_{ci}(j)}{N_{ci}(j)+b.N_{cd}(j,k)}$

où

- $N_{ci}(j)$ est le nombre d'occurrences du phonème j
- $N_{cd}(j,k)$ est le nombre d'occurrences du phonème j pour le réseau dépendant du contexte correspondant à la classe k.

La constante b est optimisée sur des données de développement. Ce type de modèle a permis d'obtenir de très bons résultats sur la base de données RESOURCE MANAGEMENT (voir Annexe D) [6].

6. Il faut toutefois noter que le système final donnant les meilleurs résultats sur le corpus RM n'utilise qu'un unique MLP pour estimer les probabilités $p(q_k|X, c_j)$ (comme notre système développé dans le paragraphe 8.3).

8.6 Modèles hybrides dépendants du contexte

Toujours dans la même optique d'utiliser les systèmes hybrides HMM/ANN pour modéliser les phonèmes dépendants du contexte, nous définissons dans ce paragraphe un nouveau type de modèle basé sur les transitions entre les phonèmes [35]. Ces modèles seront utilisés dans ce paragraphe pour, à la fois, modéliser un peu mieux la dynamique des segments phonétiques, et garder un nombre de paramètres relativement faible. En fait, les transitions entre phonèmes sont modélisées par des distributions généralisées alors que les unités phonétiques de base sont toujours des modèles de phonèmes indépendants du contexte entraînés sur la partie la plus stable (généralement celle correspondant au milieu) du phonème. Les mots sont ainsi modélisés par des séquences de distribution de probabilité représentant alternativement la partie centrale du phonème et les transitions entre les phonèmes.

8.6.1 Description du système

Dans la plupart des systèmes de reconnaissance de la parole actuels, utilisant des modèles de phonèmes dépendants du contexte, les modèles de mots sont composés d'une séquence de phonèmes dépendants du contexte (voir paragraphe 8.2). Dans notre cas, les modèles dépendants du contexte sont uniquement utilisés pour modéliser les transitions de phonèmes à phonèmes. Ainsi les modèles indépendants du contexte sont toujours utilisés pour modéliser la partie la moins coarticulée (souvent la partie centrale) des unités phonétiques utilisées. Un exemple de modèle de mot peut être visualisé sur la figure 8.5.

Le nombre de transitions entre les phonèmes étant très élevé, il est nécessaire de quantifier ces transitions en un nombre fini limité de transitions généralisées pour réduire le nombre de paramètres à entraîner. Dans les expériences reportées ici, nous n'utiliserons qu'un unique réseau (MLP) pour estimer à la fois les probabilités associées aux modèles indépendants du contexte et celles associées aux phonèmes dépendants du contexte généralisés représentant les transitions entre phonèmes (voir figure 8.6).

8.6.2 Les différents types de quantification utilisés

Deux stratégies différentes ont été utilisées pour quantifier l'ensemble des transitions entre phonèmes en un nombre restreint de classes. La première consiste à définir des classes de transitions basées sur la classe phonétique voisine des phonèmes droits et gauches. Ainsi les transitions généralisées sont caractérisées par la classe phonétique des phonèmes droits et gauches. Nous avons bien entendu supposé que les phonèmes gauches et droits d'une même

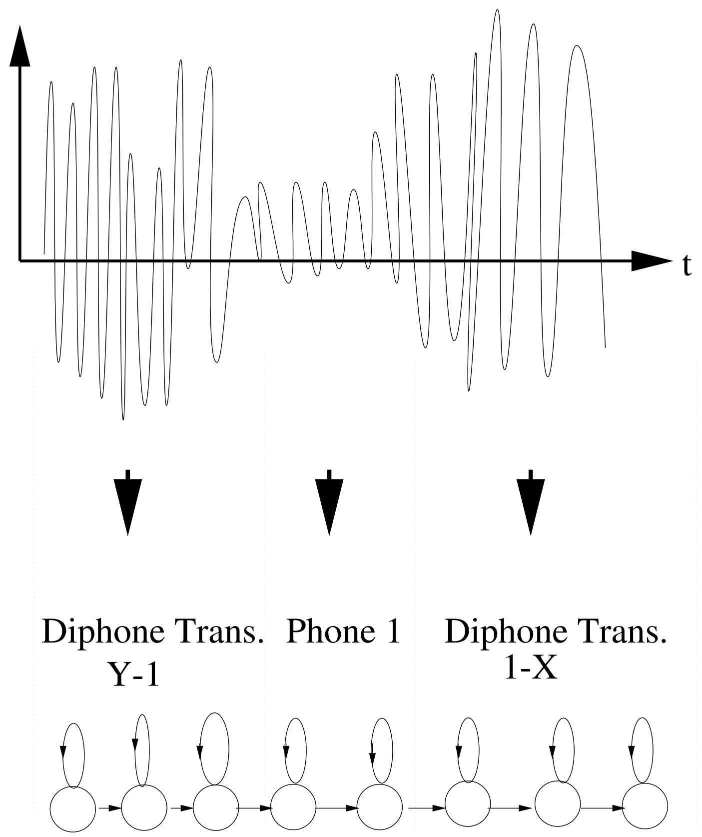

FIG. 8.5 - *Exemple de concaténation de modèles de phonèmes (Phone1) et de modèles de phonèmes généralisés (Diphone Trans Y-1 et 1-X).*

classe phonétique avaient des effets similaires sur la réalisation acoustique de la transition. Nous avons considéré deux critères différents en vue de la définition des classes phonétiques :

1. le silence + 9 classes basées sur la catégorie des phonèmes voisins. Ainsi, selon les phonèmes droits ou gauches de la transcription d'un mot, la classe phonétique est choisie selon l'appartenance à une catégorie définie dans le tableau 8.4

2. silence + 9 classes basées sur le lieu de l'articulation. Ces classes sont décrites dans le tableau 8.5

La seconde stratégie est celle déjà développée dans ce chapitre dans le paragraphe 8.2. Elle est basée sur une quantification automatique de toutes

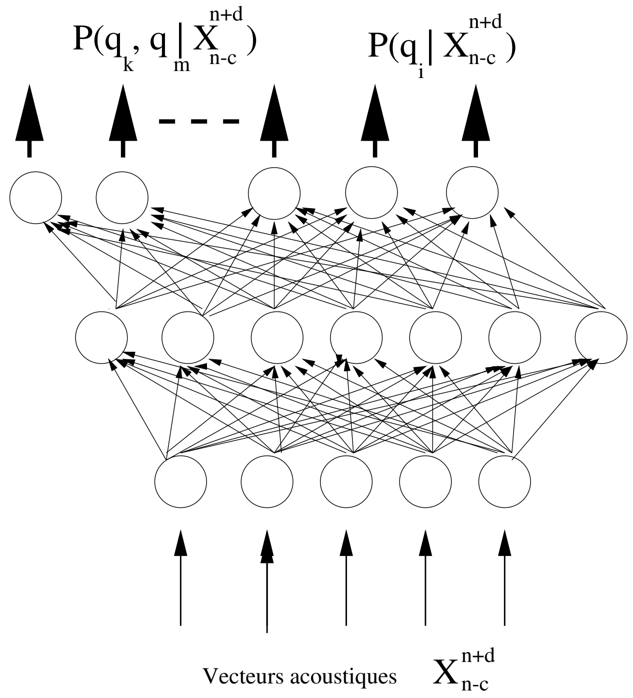

FIG. 8.6 - *Le réseau permettant l'estimation combinée des probabilités pour les transitions généralisées et les phonèmes indépendants du contexte.*

les transitions. Le nombre de classes de transitions sélectionné est 81[7].

Cette nouvelle approche est supposée donner de meilleurs résultats dans le cas particulier de la base de données PHONEBOOK. Les données d'entraînement contiennent en général les unités phonétiques dans un nombre limité de contextes gauches et droits. Ainsi les systèmes usuels utilisent cette information pour améliorer les performances en utilisant par exemple des phonèmes dépendants du contexte. Même dans le cas particulier des modèles de phonèmes indépendants du contexte qui sont couramment utilisés avec les modèles hybrides, les modèles de phonèmes capturent implicitement cette information contextuelle (d'autant plus qu'on fournit au MLP un contexte acoustique). Pourtant, dans le cas particulier où les dictionnaires des mots de tests et d'entraînement sont complètement différents, l'utilisation de l'information contextuelle peut entraîner une baisse significative du taux de re-

7. Ce nombre de classe peut bien évidemment être plus important, nous avons choisi 81 pour pouvoir comparer avec les deux approches précédentes basées sur la classe phonétique ou le lieu de l'articulation des phonèmes droits et gauches.

classes phonétiques	TIMIT phonèmes
voiced stops	b, bcl, d, dcl, g, gcl
unvoiced stops	p, pcl, t, tcl, k, kcl
affricates	jh, ch
unvoiced fricatives	s, sh, f
voiced fricatives	z, zh, th, v, dh
nasals	m, n, ng
semivowels	l, r, w, y
whisper	hh
vowels	iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy, ow, uh, uw, er

TAB. 8.4 - *Classes phonétiques voisines.*

classes phonétiques	TIMIT phonèmes
closures	bcl, dcl, gcl, pcl, tcl, kcl
front consonants	b, p, f, th, v, dh, m
middle consonants	d, t, jh, ch, s, sh, z, zh, n, l
back consonants	g, k, ng, hh
front vowels	y, iy, ih, eh, ae, ey, ay, oy
middle vowels	r, er, aa, ah, ao
back vowels	uh, uw
middle diphtongs	w, aw, ow

TAB. 8.5 - *Classes phonétiques basées sur le lieu de l'articulation.*

connaissance car les modèles ne sont plus réellement appropriés. En utilisant des phonèmes dépendants du contexte généralisés, nous espérons limiter cet effet car nous espérons que toute l'information contextuelle sera capturée par ces états de transitions, alors que les modèles de phonèmes indépendants du contexte vont se focaliser sur la partie stable de chaque segment phonétique.

8.6.3 Résultats

Les résultats obtenus avec ces nouveaux modèles sont visibles dans le tableau 8.6 où :

- CI correspond au système hybride HMM/MLP utilisant les phonèmes indépendants du contexte.
- CI&CD (I) correspond aux transitions généralisées basées sur les classes phonétiques des phonèmes droits et gauches. Le réseau MLP possède 46 (phonèmes indépendants du contexte) + 10∗10 transitions généralisées, soit 146 sorties.

– CI&CD (II) correspond aux transitions généralisées basées sur le lieu de l'articulation des phonèmes droits et gauches. Le réseau MLP possède 46 (phonèmes indépendants du contexte) + 8∗8 transitions généralisées, soit 127 sorties.

– CI&CD(III) correspond à la quantification automatique décrite au paragraphe 8.2. Le réseau MLP possède 46 (phonèmes indépendants du contexte) + 81 classes de transitions générées automatiquement, soit 127 sorties.

type	75 mots	600 mots	8000 mots
CI	2.7%	7.0%	22.0%
CI&CD(I)	1.9%	5.7%	18.9%
CI&CD(II)	1.5%	5.0%	17.0%
CI&CD(III)	1.7%	5.5%	18.3%

TAB. 8.6 - *Taux d'erreur pour une reconnaissance de mots isolés sur* PHONEBOOK *avec des dictionnaires de 75, 600 et 8000 mots avec des systèmes hybrides et des paramètres Log RASTA-PLP. Comparaison entre les modèles de phonèmes indépendants du contexte et les trois méthodes de quantification des transitions de phonèmes à phonèmes.*

Les modèles de phonèmes dépendants du contexte modélisant les transitions entre phonèmes donnent donc de meilleurs résultats, et ce de manière significative, que notre système hybride de base. Trois types de quantification ont été testés et ont donné des améliorations sensibles des taux de reconnaissance. En outre, la charge de calcul entraînée par l'utilisation de ces modèles est légèrement supérieure à celle du système de base (et largement inférieure à n'importe quel autre système dépendant du contexte). Enfin l'utilisation d'un unique réseau de neurones en vue de l'estimation des probabilités correspondant aux transitions entre phonèmes et celles correspondant aux phonèmes indépendants du contexte permet un entraînement plus discriminant (au niveau de la trame acoustique).

8.7 Test des différentes méthodes en reconnaissance de la parole continue

Nous avons montré l'efficacité des méthodes développées dans ce chapitre sur un corpus de parole téléphonique mots isolés. Nous voulions connaître l'efficacité de ces méthodes sur un corpus de parole continue. Afin de pouvoir comparer nos méthodes à celles d'autres laboratoires (voir paragraphes 8.4 et 8.5) nous avons testé les diverses méthodes décrites dans ce paragraphe sur le corpus RESOURCE MANAGEMENT. Nous avons sélectionné feb89 comme données de test, et bien entendu les 3990 phrases d'entraînement officielles de ce corpus. Les résultats sont donnés avec la grammaire wordpair de perplexité 60. Le tableau 8.7 résume l'ensemble des résultats obtenus sur ce corpus pour les méthodes décrites dans ce chapitre. Nous avons sélectionné le type de quantification des transitions ayant fourni les meilleurs taux de reconnaissance sur le corpus PHONEBOOK, c'est-à-dire les transitions généralisées basées sur le lieu de l'articulation ou sur la classe phonétique voisine des phonèmes droits et gauches.

type	Taux d'erreur
FPMS	
CI-HMM	12.4%
CI-MLP	5.4%
CD-MLP place	4.5%
CD-MLP classe	4.8%
CD-MLP all	6.5%
Cohen Franco Morgan	
CI-HMM	14.1%
CI-MLP	5.4%
CD-MLP	4.7%
CD-HMM	5.0%

TAB. 8.7 - *Taux d'erreur pour une reconnaissance de la parole continue sur* RM *avec des systèmes hybrides. Comparaison entre les modèles de phonèmes indépendants du contexte et les différentes méthodes de modélisation des phonèmes dépendants du contexte.*

Les résultats obtenus sur ce corpus indiquent qu'il est possible d'utiliser des modèles dépendants du contexte et des modèles hybrides pour améliorer sensiblement les taux de reconnaissance par rapport au système de base. Nous avons observé, en utilisant des modèles correspondant aux transitions généralisées basées sur le lieu de l'articulation des phonèmes droits et gauches, une diminution du taux d'erreur de 20%. Pour les transitions généralisées basées sur les classes phonétiques des phonèmes droits et gauches, une réduction relative de 13% du taux d'erreur a été observée. Le taux d'erreur de 4.5% (et même 4.8%) sur les données de test correspondant à la campagne d'enregistrement de février 89 (feb89) nous place parmi les meilleurs systèmes. Nous pouvons néanmoins noter que la méthode basée sur l'estimation des probabilités de tous les diphones par un unique MLP n'a pas donné les meilleurs résultats comme ce fût le cas pour le corpus PHONEBOOK. Ceci peut s'expliquer par le nombre restreint des données d'entraînement que nous possédons. En effet, RESOURCE MANAGEMENT ne possède que 3990 phrases d'entraînement (environ 1.4 M de trames) ce qui est largement inférieur aux nombres de données que nous possédions pour le corpus PHONEBOOK. Il y a fort à parier que le réseau s'est spécialisé sur les données d'entraînement et n'est plus capable de généraliser. Nous pouvons néanmoins conclure que la méthode des transitions généralisées basée sur le lieu de l'articulation ou sur les classes phonétiques des phonèmes droits et gauches fonctionne aussi très bien pour la parole continue, même dans le cas où le nombre des données d'entraînement reste faible.

8.8 Conclusion

Nous avons présenté dans ce chapitre quelques méthodes visant à utiliser l'information contextuelle avec les modèles hybrides. Nous avons observé une amélioration sensible des taux de reconnaissance par rapport aux modèles indépendants du contexte sans une augmentation importante de la charge de calcul. L'ensemble des résultats obtenus avec ces méthodes est visible sur la figure 8.7 pour le corpus PHONEBOOK où :

- CI MLP : représente la méthode hybride de base, phonèmes indépendants du contexte.
- CD HMM (600 $\rightarrow$ 1000) : représente la méthode HMM à distribution multi-gaussiennes (16 gaussiennes par états, matrice de covariance diagonale) et des phonèmes dépendants du contexte (diphones) après quantification en 600 $\rightarrow$ 1000 classes.
- CD HMM (diphones) : représente la méthode HMM à distribution multi-gaussiennes (16 gaussiennes par états, matrice de covariance diagonale) et des phonèmes dépendants du contexte (diphones) sans quantification.
- CD MLP (600) : représente la méthode où un réseau MLP a été entraîné sur les 600 classes issues de la méthode CD HMM (600) (9 trames de contexte, 1000 noeuds cachés et 600 sorties).
- CD MLP (diphones): représente la méthode où un réseau MLP a été entraîné sur les diphones de la méthode CD HMM (diphones) (9 trames

de contexte, 1000 noeuds cachés et 1098 sorties correspondant aux diphones présents dans le corpus PHONEBOOK).

- CI&CD (I) correspond aux transitions généralisées basées sur les classes phonétiques des phonèmes droits et gauches.
- CI&CD (II) correspond aux transitions généralisées basées sur le lieu de l'articulation des phonèmes droits et gauches.
- CI&CD(III) correspond à la quantification automatique décrite au paragraphe 8.2.

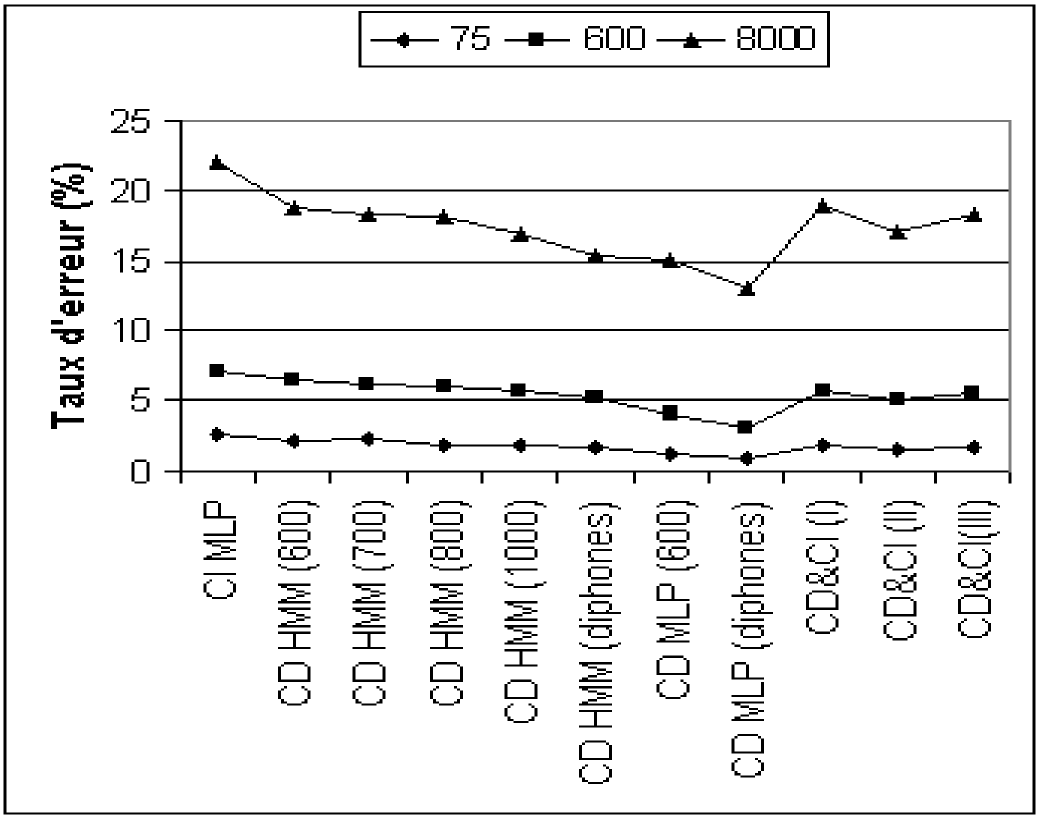

FIG. 8.7 - *Résumé des taux obtenus pour les différentes méthodes sur le corpus* PHONEBOOK.

Les meilleurs résultats ont été obtenus avec la méthode CD MLP (diphones). Nous avons observé une réduction significative du taux d'erreur par rapport au modèle hybride de base pour les trois dictionnaires (66.6%

d'amélioration pour 75 mots, 54% pour 600 mots et 41% pour 8000 mots). Il va de soi que la charge de calcul est plus importante car le réseau possède beaucoup plus de paramètres (par rapport au modèle CI). Pourtant, la charge de calcul reste inférieure à celle occasionnée par l'utilisation de modèles dépendants du contexte à distribution multi-gaussiennes (méthode CD HMM ($N = 600..all$)). De plus, d'après l'étude du chapitre 7, il est possible d'utiliser des méthodes de suppression automatique de trames pour accélérer le processus de reconnaissance sans détériorer de manière significative les taux de reconnaissance. Les mêmes méthodes ont été testées sur le corpus RESOURCE MANAGEMENT, et nous avons observé que :

– La méthode CD MLP (diphones) détériore les taux de reconnaissance par rapport au modèle hybride de base. Ceci est lié au fait que le nombre des données d'entraînement est faible. Il y a probablement eu surentraînement, le réseau n'étant plus capable de généraliser. Le principal inconvénient de la méthode est donc la nécessité de posséder une base de données suffisamment grande pour effectuer un bon entraînement du réseau (il faut en effet assez d'occurrences des différents diphones pour estimer les probabilités). Son principal avantage réside dans les performances qui sont observées lorsque le réseau est bien entraîné. Une solution possible consiste à quantifier les diphones pour réduire ainsi le nombre de paramètres à entraîner.

– La méthode correspondant aux transitions généralisées basées sur le lieu de l'articulation ou sur les classes phonétiques des phonèmes droits et gauches fonctionne particulièrement bien dans les deux cas (parole isolée et continue). Le nombre de paramètres à estimer étant restreint, nous n'avons pas observé de problèmes à l'entraînement pour le corpus RESOURCE MANAGEMENT.

Chapitre 9

La méthode de fusion de données

9.1 Introduction

Comme nous l'avons remarqué dans les chapitres précédents, les modèles hybrides HMM/ANN sont très efficaces pour la reconnaissance de la parole. Néanmoins l'un des principaux défauts liés à ces modèles 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 HMMs continus) lorsque le nombre des données d'entraînement et/ou de paramètres est fortement augmenté. Le tableau 9.1 reporte les résultats obtenus sur le corpus PHONEBOOK avec deux réseaux MLP entraînés sur environ 3.5 millions de trames acoustiques (26000 phrases) pour le premier et environ 11 millions de trames (790000 phrases) pour le second[1].

Taille	MLP (3.5 M)	MLP (11 M)
75 mots	2.7%	2.6%
600 mots	7.0%	7.1%
8000 mots	22.0%	22.3%

TAB. 9.1 - *Taux d'erreur au niveau du mot pour les trois dictionnaires de 75, 600, et 8000 mots et deux modèles hybrides entraînés avec 3.5 et 11 millions de trames acoustiques (paramètres log RASTA-PLP).*

Les résultats reportés sur le tableau 9.1 montrent que l'augmentation du nombre des données d'entraînement 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 3.5 millions de trames et donc l'augmentation du nombre des données d'entraînement n'a aucun effet bénéfique sur le système. Nous proposons dans ce chapitre 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, particulièrement dans le domaine de la reconnaissance optique de caractères [46].

A la base, l'hypothèse est que, si les modèles sont entraînés sur différentes parties du fichier d'entraînement, 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

1. Correspondant aux données d'entraînement du système de base et à l'ensemble des données d'entraînement du corpus PHONEBOOK (voir Annexe D).

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 utilisée [44]. Nous allons décrire dans ce chapitre une procédure simple permettant de diviser les données d'entraînement de manière efficace afin qu'après recombinaison des modèles, une amélioration sensible des taux de reconnaissance soit observée.

9.2 La procédure de fusion

Cette procédure vise à éclater les données d'entraînement 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 la base de données PHONEBOOK pour des vocabulaires de 75, 600 et 8000 mots. N'ayant pas observé d'amélioration significative du taux de reconnaissance en utilisant un réseau MLP entraîné sur la totalité des données d'entraînement (11 millions de trames acoustiques) par rapport à celui entraîné sur un plus petit ensemble d'entraînement (voir tableau 9.1), 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 MLP classique estimant les probabilités a posteriori de phonèmes indépendants du contexte (CI) sur une petite partie des données d'entraînement (ce réseau sera par la suite dénommé MLP1). Ce réseau est alors utilisé pour filtrer le reste des données d'entraînement pour un second réseau. Les données conservées pour entraîner le second réseau sont celles pour lesquelles le premier réseau MLP1 s'est trompé dans la classification. Ainsi pour toutes les données d'entraînement, nous comparons pour chaque trame acoustique, les sorties du réseau MLP1 (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'entraînement, sinon elle est gardée. Dans un esprit de simplification et pour éviter la suppression de trames acoustiques correspondant à un même phonème sur une grosse partie de la base d'entraînement, nous avons décidé, pour chaque phrase de la base d'entraînement, 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, la phrase est gardée pour l'entraînement du deuxième réseau; sinon, elle n'est plus prise en compte. Il va de soi que plus le seuil fixé est élevé, moins il y aura de phrases gardées et donc moins la subdivision des données d'entraînement 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 MLP1) et pour s'assurer de la validité de l'entraînement des différents réseaux. Nous avons ainsi filtré une première fois les données d'entraînement pour créer un nouvel ensemble d'entraînement qui sera utilisé pour entraîner un second réseau (noté MLP2). Enfin, le reste des données d'entraînement est passé au travers des deux réseaux MLP1 et MLP2. Si ces deux réseaux sont en désaccord sur la classification des exemples présentés à l'entrée, cet exemple est alors ajouté aux données d'entraînement du troisième réseau. En revanche, si les deux réseaux sont d'accord, l'exemple est écarté. Là encore, un taux d'erreur par phrase est calculé et comparé à un seuil pour décider ou non de l'insertion dans les données d'entraînement du troisième réseau (noté MLP3). Le processus de division des données d'entraînement 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'entraînement ait été utilisé. L'entraînement des réseaux est réalisé de manière classique par propagation arrière du gradient de l'erreur quadratique (voir chapitre 3). Ces réseaux sont ensuite combinés par différentes méthodes pour estimer les probabilités utilisées par les HMMs.

9.3 Description du système

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 MLP1, MLP2, MLP3 est composé de 47 sorties (correspondant aux 47 phonèmes utilisés dans le dictionnaire), 1000 noeuds cachés et 234 noeuds en entrée correspondant à 9 trames acoustiques de 26 paramètres. Le système utilisé lors de la reconnaissance est décrit sur la figure 9.1.

Ainsi, après extraction des paramètres acoustiques classiques (log RASTA-PLP pour les expériences décrites ci-après) et passage par chacun des trois réseaux MLP1, MLP2 et MLP3, nous disposons pour chaque trame acoustique de 3 séries de probabilités qu'il nous faut combiner. Il existe plusieurs techniques pour combiner les probabilités; nous allons en décrire quelques unes qui ont été testées sur la base de données PHONEBOOK.

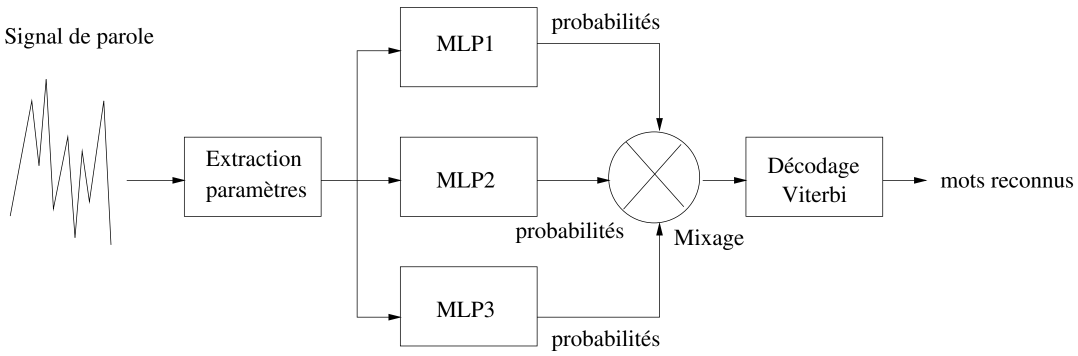

FIG. 9.1 - *Le processus de reconnaissance pour la méthode de fusion.*

9.4 Les différentes méthodes de combinaison

Dans ce paragraphe, nous allons décrire chacune des méthodes utilisées pour combiner les probabilités issues des trois réseaux entraînés par la méthode de fusion.

9.4.1 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 :

$$OUT[i] = \frac{1}{N} \sum_{j=1}^{N} Out_j[i] \tag{9.1}$$

où

- N est le nombre d'experts (nombre de réseaux) utilisés (3 dans notre cas),
- $Out_j[i]$ est la composante i du vecteur observé à la sortie de l'expert j (ici $out_j[i] = P_{MLP_j}[q_i|X]$).

Le vecteur de sortie fourni aux HMMs est donc égal à la moyenne des probabilités des trois MLPs.

9.4.2 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 $out_j[i] = log[P_{MLP_j}[q_i|X]]$. Ce type de combinaison a déjà été utilisé avec succès dans des recombinaisons de systèmes à bandes multiples [34], ou pour le calcul du score du "garbage" [2] en reconnaissance de mots clés par modèles hybrides [13].Le vecteur de sortie fourni aux modèles HMMs est donc la moyenne des log probabilités de chacune des sorties des réseaux.

9.4.3 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 MLP1 et MLP2 s'accordent sur la classification de la trame acoustique (même sortie ayant la meilleure probabilité), alors le vecteur de sortie du réseau MLP1 est utilisé comme vecteur de sortie fourni aux HMMs. Si les deux réseaux sont en désaccord, alors le vecteur de sortie du réseau MLP3 est utilisé comme vecteur de probabilités fournit aux HMMs. Cette méthode est appelée méthode du vote et est couramment utilisée pour la combinaison de plusieurs modèles.

9.4.4 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 MLP1 et MLP2. 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 :

$$Entropie = -\sum_{k=1}^{Noutputs} p(q_k|X).log(p(q_k|X)) \qquad (9.2)$$

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

2. Le "garbage" est un modèle qui prend en compte l'ensemble des mots prononcés par un locuteur qui n'appartiennent pas au lexique utilisé.

autres), l'entropie est alors nulle. Dans le cas où le réseau n'est pas capable de se décider (même probabilité $\frac{1}{Noutputs}$ pour chacune des sorties), l'entropie vaut alors $-log(\frac{1}{Noutputs}) > 0$.

9.4.5 Combinaison par l'intermédiaire d'un MLP

Ce type de combinaison fournit généralement des résultats assez bons. Il suffit d'entraîner un réseau MLP classique avec en entrée, un vecteur composé des probabilités de sortie de chacun des trois MLPs et un contexte acoustique quelconque. Ainsi dans les expériences menées sur la base de données PHONEBOOK, nous avons entraîné un réseau MLP avec 3*47 composantes en entrée, correspondant aux probabilités des 3 réseaux et un contexte de 3 trames acoustiques, soit 423 noeuds d'entrée, 1000 noeuds cachés et 47 sorties.

9.4.6 Conclusion

Il existe de nombreuses méthodes pour combiner les probabilités a posteriori des trois réseaux MLP1, MLP2 et MLP3. Nous avons sélectionné celles qui nous paraissaient les plus simples et les plus efficaces en nous basant sur les quelques expériences déjà effectuées sur l'approche multi-bandes [34]. D'autres méthodes du type "mixture of experts" (mélange d'experts) peuvent aussi être utilisées [56, 58].

9.5 Expériences et Résultats

La procédure décrite dans ce chapitre a été testée sur la base de données PHONEBOOK. Les paramètres utilisés sont les log RASTA-PLP calculés toutes les 10 ms sur des fenêtres d'analyse de 30 ms. L'ordre de l'analyse LPC est fixé à 10. Sur les 79777 phrases composant les données d'entraînement (environ 11 millions de trames acoustiques) les trois réseaux ont été entraînés sur les différentes parties décrites dans le tableau 9.2.

MLP	Trames (train)	Trames (cross)
1	2.570.000	256.000
2	2.975.000	331.000
3	4.417.000	510.000
réseau de base	2.500.000	950.000

TAB. 9.2 - *Nombres de trames acoustiques utilisées pour chacun des réseaux.*

A titre de comparaison, nous mettrons dans chacun des tableaux de résultats ceux correspondant au modèle hybride HMM/ANN de base dans les mêmes conditions. Les premiers tests effectués sur cette méthode ont été réalisés au niveau de la trame acoustique. Vous trouverez, pour l'ensemble d'entraînement utilisé pour le modèle de base, les taux de reconnaissance au niveau de la trame acoustique dans le tableau 9.3 pour chacune des méthodes de combinaisons décrites dans le paragraphe précédent.

	linéaire	log linéaire	vote	Entropie	MLP	MLP de base
Train	75.2%	75.1%	74.7%	73.4%	75.8%	73.9%
Cross	73.7%	72.7%	73.3%	72.0%	74.2%	68.8%

TAB. 9.3 - *Taux de reconnaissance au niveau de la trame acoustique pour les différentes méthodes de combinaisons et les paramètres log RASTA-PLP.*

Les premiers résultats au niveau de la trame acoustique sont fortement encourageants puisque nous observons une réduction d'environ 16% du taux d'erreur. Observons maintenant les performances au niveau du mot pour des vocabulaires de 75, 600 et 8000 mots sur le tableau 9.4.

Taille	linéaire	log linéaire	vote	Entropie	MLP	MLP de base
75 mots	1.5%	1.5%	1.7%	1.7%	1.8%	2.7%
600 mots	5.8%	6.0%	6.0%	5.8%	5.9%	7.0%
8000 mots	19.0%	18.9%	19.2%	19.1%	18.1%	22.0%

TAB. 9.4 - *Taux d'erreur au niveau du mot pour les trois dictionnaires de 75, 600, et 8000 mots et les différentes méthodes de combinaisons avec paramètres log RASTA-PLP.*

Nous observons, là encore, une réduction significative du taux d'erreur pour les trois types de vocabulaire (44% pour 75 mots, 17% pour 600 mots et 14% pour 8000 mots).

9.6 Conclusion

Nous avons défini une méthode permettant de diviser en plusieurs parties l'ensemble d'entraînement et d'entraîner plusieurs MLPs sur chacune de ces parties. Nous espérions ainsi tirer profit de l'entraînement 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 vote
- la combinaison par le critère entropique
- la combinaison par un MLP

Une réduction significative du taux d'erreur a pu être observée en utilisant la méthode de fusion décrite dans ce chapitre (44% pour 75 mots, 17% pour 600 mots et 14% pour 8000 mots). Cette procédure nous a permis de tirer au mieux parti des nombreuses données d'entraînement dont nous disposions. Cette amélioration, obtenue dans le cadre d'une reconnaissance de mots isolés, devrait aussi être constatée pour un système de reconnaissance de la parole continue. Dans cette optique, la même procédure décrite dans ce chapitre pourra être appliquée et le même système utilisé. 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'entraînement. Ne possédant pas de corpus parole continue de taille suffisante, cette méthode n'a pas pu être testée pour la parole continue.

Il semble que la méthode de combinaison des sorties des MLPs la plus efficace (du moins pour l'expérience décrite ici) consiste à combiner linéairement les sorties des trois MLPs utilisés. Cette combinaison linéaire est somme toute très facile à mettre en oeuvre.

Chapitre 10

Conclusions

Nous avons décrit dans cette thèse les modèles statistiques les plus utilisés en reconnaissance de la parole : les HMMs. Ces modèles sont à la base de la presque totalité des outils de reconnaissance de la parole présents sur le marché. Néanmoins, la mise en oeuvre de ces modèles nécessite des hypothèses contraignantes qui peuvent être adoucies en utilisant conjointement les modèles de Markov cachés et un réseau de neurones. Ces modèles sont appelés modèles hybrides et permettent d'obtenir de meilleures performances que les modèles HMMs classiques. Les avantages dans l'utilisation cumulée des modèles de Markov cachés et des réseaux de neurones se reflètent clairement dans les expériences que nous avons réalisées sur des corpus de parole isolée et continue pour de petits et grands vocabulaires tout au long de ce travail.

10.1 Réestimation des N meilleures séquences de mots

De nombreux algorithmes peuvent être utilisés pour générer les N meilleures séquences de mots reconnues au lieu de la plus probable uniquement. Nous avons décrit les avantages et inconvénients de ces méthodes. L'algorithme que nous avons sélectionné dans le cas de vocabulaire de taille moyenne est basé sur l'algorithme A*. C'est une méthode de recherche en deux passes :

1. Estimation de la probabilité d'émettre les trames le long du meilleur chemin par un algorithme de programmation dynamique classique (Viterbi).
2. Utilisation de l'algorithme A* (algorithme de recherche dans un graphe) pour générer les N meilleures séquences de mots.

Dans le cas de très grands vocabulaires, nous avons utilisé un algorithme développé par S. Renals (Noway) basé sur une approximation du critère A*. Cet algorithme utilise la méthode A* en une seule passe. Le critère d'évaluation du score est différent. Au lieu d'utiliser les scores acoustiques calculés dans une première passe, le score correspondant à la partie non encore traitée est estimé directement dans la première passe sans "lookahead". Noway permet aussi de générer un treillis de mots pour chaque phrase prononcée. Les treillis permettent une représentation des phrases reconnues sous formes de graphes. Un treillis est représenté par un ensemble de noeuds et de liens. Chaque lien porte un mot correspondant à une hypothèse et le score acoustique correspondant et chaque noeud correspond à un instant donné. Ce treillis est utilisé pour générer les N meilleures séquences de mots dans le cas de vocabulaires de taille importante. Notre algorithme basé sur la méthode A* et Noway, permet donc de générer les N meilleures séquences de mots pour des vocabulaires de tailles moyennes[1] et pour de grands vocabulaires. Il faut donc maintenant trouver des méthodes efficaces pour sélectionner parmi les N meilleures séquences de mots, celle correspondant à ce qui a été effectivement énoncé.

Nous avons proposé une première tentative de réestimation des N meilleures séquences par l'intermédiaire du modèle indexé temporellement. Ce modèle utilise une information supplémentaire à l'entrée du réseau MLP : la durée depuis laquelle on est présent dans le phonème. Cette durée est facilement calculable lors de l'entraînement à partir de la segmentation Viterbi. Nous avons montré que pour peu que l'on fournisse à l'entrée du MLP le bon index temporel (celui calculé après un alignement des phrases de tests), ce modèle avait des performances largement supérieures à celles des modèles classiques (HMM ou HMM+MLP). Le problème réside dans l'estimation de cet index temporel. Nous avons proposé d'utiliser ce modèle en vue d'une réestimation des N meilleures séquences de mots. En effet, il est possible d'obtenir la segmentation Viterbi associée à chacune des N meilleures séquences de mots. Or à partir de cette segmentation, il est facile de calculer la durée depuis laquelle on est dans le phonème et donc cet index temporel. Il nous paraissait donc possible d'utiliser ce réseau facilement pour recalculer les scores de ces N meilleures séquences de mots. Bien qu'attractive, cette méthode n'a pas fourni les résultats escomptés. Une explication possible est qu'il existe une différence notoire entre les segmentations obtenues après génération des N meilleures séquences et celles utilisées pour l'entraînement du réseau time-index. Une analyse un peu plus fine des résultats a montré que les scores obtenus après réestimation des N meilleures séquences étaient sensiblement les mêmes. Nous espérions en utilisant ce modèle écarter certaines segmentations (donc solutions) peu probables. Il serait peut être nécessaire pour améliorer le système d'utiliser des contre-exemples pour l'entraînement du réseau.

En revanche, l'utilisation de modèles dépendants du contexte pour réestimer les N meilleures séquences de mots a fourni de très bons résultats. Nous bénéficions en fait de la connaissance des séquences de mots pour modéliser précisément les transitions entre les mots (autrement qu'en modifiant l'algorithme de programmation dynamique pour étendre avec le bon modèle dépendant du contexte correspondant à la transition entre deux mots). Nous

1. Noway n'est pas adapté au traitement des petits vocabulaires.

avons observé avec ce type de méthode une réduction du taux d'erreur de l'ordre de 20% par rapport au modèle hybride classique sur la base de données RESOURCE MANAGEMENT.

De même, nous avons montré qu'il était possible de recalculer les scores des N meilleures séquences avec des modèles de langage d'ordre supérieur. L'introduction des probabilités des modèles de langage d'ordre élevé ($3 \leq N$) dans la programmation dynamique classique Viterbi est difficile. Une solution possible à ce problème (outre l'utilisation de techniques A*) consiste à générer avec un modèle de langage simple (bigramme) les N meilleures séquences de mots et à les réestimer avec ces modèles de langage d'ordre supérieur. Nous avons observé une réduction du taux d'erreur de l'ordre de 7% sur BREF-80 en recalculant les scores des 200 meilleures phrases à l'aide d'un 5-gramme estimé sur les articles du journal "Le Monde". L'amélioration observée est moins importante que celle obtenue avec la méthode précédente. Ceci peut s'expliquer par le fait que le 5-gramme utilisé pour la réestimation n'a été entraîné que sur très peu de données.

10.2 Développement d'un système de reconnaissance de la parole continue en français

Noway a aussi été utilisé pour tester notre système de base en français pour des vocabulaires de 1K, 3K, 13K et 65K mots. Le système a été entraîné sur le corpus BREF-80. Cette base de données n'étant pas manuellement segmentée, nous avons utilisé un synthétiseur vocal de bonne qualité (MBROLA) afin de générer une segmentation pour initialiser le processus d'entraînement. Les résultats obtenus au niveau du phonème sont parmi les meilleurs jamais publiés à notre connaissance sur ce corpus particulier (75% de reconnaissance). En revanche les résultats au niveau du mot sont moins bons que nous ne l'espérions. Les principales erreurs commises au niveau du mot sont dues au modèle de langage utilisé. L'acoustique des phrases reconnues est en revanche bonne.

10.3 Contexte et suppression de trames

Nous avons montré qu'il était possible de remplacer le calcul des dérivées dans le cas de modèles hybrides par un contexte acoustique plus large sans détérioration significative des taux de reconnaissance. De plus, il est possible d'accélérer fortement le processus de reconnaissance (particulièrement en mode mots isolés) en supprimant de manière automatique le calcul des probabilités pour 1 trame sur 2 et même 2 trames sur 3 sans observer de dégradation au niveau des taux de reconnaissance. Cette technique de suppression automatique des trames est d'ailleurs employée pour les démonstrations du laboratoire TCTS, permettant d'utiliser les modèles hybrides en temps réel pour un vocabulaire d'environ 600 mots sur un simple PC (pentium 133 MHz). Cette technique répond à l'un des principaux désavantages du modèle hybride : le temps de calcul qui lui est associé.

10.4 Modèles dépendants du contexte

La parole est un signal possédant une très grande variabilité. La plupart des outils de reconnaissance "grand vocabulaire" utilisent les phonèmes comme unité linguistique de base. Pourtant, les phonèmes sont des unités très petites et leur réalisation acoustique est extrêmement variable et influencée par le contexte dans lequel ils apparaissent. De nombreux systèmes de reconnaissance de la parole utilisent des modèles de phonèmes dépendants du contexte (diphone ou triphone) avec des HMMs à distribution multigaussienne. Mais l'utilisation de ces modèles dépendants du contexte couplés avec les modèles hybrides est beaucoup moins fréquente.
Nous nous sommes intéressés au couplage de ces deux méthodes pour améliorer les performances des systèmes de reconnaissance. Le problème vient du nombre de paramètres à estimer. En effet, en moyenne on utilise un minimum de 1000 unités pour modéliser des phonèmes dépendants du contexte du type diphone et jusqu'à plusieurs milliers pour les triphones. Ceci correspond à plusieurs millions de paramètres à estimer; ce qui est difficile vue la taille des corpus et le temps nécessaire pour l'entraînement du réseau. Nous avons présenté dans cette thèse plusieurs méthodes pour entraîner les modèles dépendants du contexte et estimer la probabilité $P(q_k, c_j|X)$ (probabilité d'être dans l'état q_k et le contexte c_j étant donné la séquence de vecteurs acoustiques X) à l'aide d'un ou plusieurs réseaux de neurones.

La première méthode que nous avons testée consiste à estimer directement cette probabilité par l'intermédiaire d'un réseau MLP entraîné de la même manière que les modèles de phonèmes indépendants du contexte. Les résultats obtenus avec ce type de méthodes sont réellement impressionnants puisque nous avons observé une réduction significative des taux d'erreur sur le corpus PHONEBOOK (66% pour 75 mots de vocabulaire, 54% pour 600 mots et 41% pour 8000 mots) par rapport au modèle indépendant du contexte de base. Cette méthode possède néanmoins un désavantage certain: il est nécessaire d'avoir un corpus d'entraînement important pour éviter un surentraînement du réseau. Il faut un nombre suffisant d'occurrences des différents diphones pour estimer les probabilités. C'est probablement pourquoi nous avons observé une détérioration des taux de reconnaissance avec cette méthode sur le corpus RESOURCE MANAGEMENT. La charge de calcul associée est un peu plus importante mais elle reste inférieure à celle occasionnée par l'utilisation de modèles CD HMMs à distribution multigaussienne. De plus, il est possible d'utiliser des méthodes de quantification pour réduire le nombre de paramètres. Nous avons utilisé une méthode de quantification automatique ne nécessitant pas de connaissance linguistique. Les résultats obtenus sont également très bons. Il est aussi possible d'utiliser des méthodes de suppression automatique de trames pour réduire encore la charge de calcul.

L'ICSI utilise une décomposition de la probabilité que l'on cherche à estimer en plusieurs termes pouvant être calculés à l'aide de plusieurs réseaux de tailles sensiblement égales à celle du MLP de base (CI). Standford utilise aussi le même type de décomposition que l'ICSI, mais propose de calculer le terme $p(q_k|X, c_j)$ en réduisant les données fournies à l'entrée du MLP à celles correspondant au contexte c_j. Ainsi en notant M le nombre de contextes, cette méthode utilise une série de M MLPs similaires à ceux utilisés dans le cas des phonèmes indépendants du contexte, excepté le fait que chaque MLP est entraîné uniquement avec des données correspondantes à un contexte c_j. Horacio Franco rapporte un taux d'erreur de 4.7% sur feb89 avec cette méthode[2].

Enfin, dans la même optique d'utiliser les systèmes hybrides HMM/ANN pour modéliser les phonèmes dépendants du contexte, nous avons défini un nouveau type de modèle basé sur les transitions entre les phonèmes. Ces modèles sont utilisés pour, à la fois, modéliser un peu mieux la dynamique des segments phonétiques, et garder un nombre de paramètres relativement faible. En fait, les transitions entre phonèmes sont modélisées par des distributions généralisées alors que les unités phonétiques de base sont toujours des modèles de phonèmes indépendants du contexte entraînés sur la partie la plus stable (généralement celle correspondant au milieu) du phonème. Les mots sont ainsi modélisés par des séquences de distributions de probabilité représentant alternativement la partie centrale du phonème et les transitions entre les phonèmes. Le nombre de transitions entre les phonèmes étant très élevé, il est nécessaire de quantifier ces transitions en un nombre fini limité de transitions généralisées pour réduire le nombre de paramètres à entraîner. Deux stratégies différentes ont été utilisées pour quantifier l'ensemble des transitions entre phonèmes en un nombre restreint de classes. La première consiste à définir des classes de transitions basées sur la classe phonétique

2. Il faut remarquer quand même que le système final utilise un unique réseau pour estimer les différentes probabilités.

voisine des phonèmes droits et gauches. Nous avons considéré deux critères différents en vue de la définition des classes phonétiques :

1. le silence + 9 classes basées sur la catégorie des phonèmes voisins.

2. silence + 9 classes basées sur le lieu de l'articulation.

La seconde stratégie est basée sur une quantification automatique de toutes les transitions. Nous avons testé ces modèles sur le corpus PHONEBOOK et nous avons observé une réduction du taux d'erreur de 44% pour 75 mots, 28% pour 600 mots et 23% pour 8000 mots par rapport au système hybride indépendant du contexte de base sur le corpus PHONEBOOK. De plus, nous avons aussi testé ces méthodes de modélisation du contexte sur la base de données de parole continue (RESOURCE MANAGEMENT) pour pouvoir comparer les méthodes mises au point dans ce travail avec celles d'autres laboratoires et vérifier leur validité dans d'autres conditions. Les résultats observés sur le corpus RESOURCE MANAGEMENT (feb89) sont bons puisque nous obtenons, en utilisant des transitions généralisées entre phonèmes basées sur le lieu de l'articulation des phonèmes droits et gauches, un taux d'erreur de 4.5% (réduction du taux d'erreur de 20% par rapport au modèle hybride CI). Ce taux est comparable et même légèrement meilleur que la plupart des résultats publiés à notre connaissance par d'autres laboratoires utilisant les modèles hybrides sur ce corpus.

Quelle que soit la méthode utilisée pour modéliser l'information contextuelle avec les modèles hybrides, nous avons observé une amélioration sensible du taux de reconnaissance par rapport aux modèles indépendants du contexte. Il apparaît donc que l'utilisation de modèles dépendants du contexte et de modèles hybrides améliore énormément les taux de reconnaissance (comme c'est le cas pour les modèles HMMs classiques).

10.5 Méthode de fusion

Nous avons défini une méthode permettant de diviser en plusieurs parties les données d'entraînement et de construire un système basé sur plusieurs MLPs entraînés sur chacune de ces parties. Lors de la reconnaissance, plusieurs critères de recombinaison ont été testés (combinaisons linéaires, linéaires dans le domaine logarithmique, par vote, par le critère entropique ou par un MLP). Là encore, nous avons observé une réduction significative du taux d'erreur sur le corpus PHONEBOOK par rapport au modèle hybride de base pour les trois dictionnaires (44%, 17% et 14% pour 75, 600 et 8000 mots). Cette procédure nous a permis de tirer parti des nombreuses données d'entraînement dont nous disposions. La méthode de combinaison des MLPs la plus efficace (du moins pour le corpus PHONEBOOK) consiste à combiner linéairement les sorties des 3 MLPs utilisés. Cette combinaison est extrêmement facile à mettre en oeuvre. Ce type de technique nécessite un nombre de données d'entraînement suffisant pour être efficace, c'est pourquoi nous l'avons testé uniquement sur la base de données PHONEBOOK.

10.6 Perspectives

L'une des méthodes testées sur le corpus de parole continue RESOURCE MANAGEMENT a donné des résultats moins bons que ceux obtenus sur le corpus PHONEBOOK. Nous avons avancé comme explication le fait que la taille des données d'entraînement était trop faible pour pouvoir entraîner le réseau MLP servant à estimer les probabilités des diphones. Il serait peut être intéressant de vérifier cette conclusion sur un autre corpus pour valider complètement la méthode développée.

De plus, les premiers résultats obtenus avec la méthode de fusion de données sont très prometteurs. Nous prévoyons de l'appliquer à la reconnaissance de la parole continue grand vocabulaire dès que nous posséderons un corpus suffisant pour réaliser l'entraînement des différents réseaux[3]. Nous pensons appliquer cette technique aux modèles dépendants du contexte tels que ceux qui ont été développé dans ce travail. En outre, il serait intéressant de travailler encore sur les méthodes qui peuvent être utilisées pour recombiner plusieurs systèmes (notamment des méthodes du type mélange d'experts "mixture of expert").

10.7 Conclusion Finale

La génération des N meilleures séquences de mots permet d'en effectuer la réestimation par l'intermédiaire de différentes techniques développées dans ce travail. Nous avons observé des réductions de 20% et 7% des taux d'erreur en utilisant des modèles dépendants du contexte et des modèles de langage d'ordre supérieur.

De plus, les travaux menés dans cette thèse pour étendre les méthodes hybrides aux phonèmes dépendants du contexte ont amené une réduction si-

3. Par exemple le corpus BREF entier (65000 phrases) au lieu de BREF-80 (5300 phrases).

gnificative des taux d'erreur par rapport aux modèles de base. Une réduction relative du taux d'erreur de 50% pour 75 mots, 38% pour 600 mots et 18% pour 8000 mots sur le corpus PHONEBOOK et de 20% sur le corpus RESOURCE MANAGEMENT a été observée dans le cas d'une modélisation des transitions entre phonèmes basée sur le lieu de l'articulation. Nous avons aussi observé un baisse relative du taux d'erreur de 66% pour 75 mots, 54% pour 600 mots, 41% pour 8000 mots sur PHONEBOOK, en utilisant un unique MLP pour estimer les probabilités des diphones dans le contexte gauche. Les résultats obtenus sur le corpus Phonebook sont parmi les meilleurs jamais publiés à notre connaissance. Ces techniques de modélisation devraient être appliquées dans les diverses démonstrations du laboratoire TCTS.

La dernière méthode développée dans cette thèse, basée sur la fusion de données a permis d'obtenir des réductions relatives du taux d'erreur de 44%, 17% et 14% pour des vocabulaires de 75, 600 et 8000 mots sur le corpus PHONEBOOK. Elle peut aussi facilement être utilisée dans le cadre d'une évaluation sur des corpus officiels (où on dispose de suffisamment de données). Nous pensons utiliser cette technique pour l'évaluation AUPELF des systèmes de reconnaissance de la parole en français. En outre, il serait intéressant de travailler encore sur les méthodes qui peuvent être utilisées pour recombiner plusieurs systèmes (notamment des méthodes du type mélange d'experts "mixture of expert").

Annexe A

L'ensemble des phonèmes utilisés

Le tableau suivant décrit l'ensemble des phonèmes utilisés avec la base de données PHONEBOOK. Il comprends 45 phonèmes différents.

TIMIT phonemes	Exemple	Transcription	Phonemes	Exemple	Transcription
b , bcl	bee	BCL B iy	r	ray	R ey
d, dcl	day	DCL D ey	w	way	W ey
g, gcl	gay	GCL G ey	y	yacht	Y aa tcl t
p, pcl	pea	PCL P iy	hh	hay	HH ey
t, tcl	tea	TCL T iy	iy	beet	bcl b IY tcl t
k, kcl	key	KCL K iy	ih	bit	bcl b IH tcl t
jh	joke	DCL JH ow kcl k	eh	bet	bcl b EH tcl t
ch	choke	TCL CH ow kcl k	ey	bait	bcl b EY tcl t
s	sea	S iy	ae	bat	bcl b AE tcl t
sh	she	SH iy	aa	bott	bcl b AA tcl t
z	zone	Z ow n	aw	bout	bcl b AW tcl t
zh	azure	ae ZH er	ay	bite	bcl b AY tcl t
f	fin	F ih n	ah	but	bcl b AH tcl t
th	thin	TH ih n	ao	bought	bcl b AO tcl t
v	van	V ae n	oy	boy	bcl b OY
dh	then	DH e n	ow	boat	bcl b OW tcl t
m	mom	M aa M	uh	book	bcl b UH kcl k
n	noon	N uw N	uw	boot	bcl b UW tcl t
ng	sing	s ih NG	er	bird	bcl b ER dcl d
l	lay	L ey	h#	Sil	h#

TAB. A.1 - *Liste des phonèmes utilisés pour Phonebook.*

Le tableau suivant décrit l'ensemble des phonèmes utilisés avec la base de données BREF. Il comprends 35 phonèmes différents.

Vowels		Fricatives	
i	idée	f	feu
e	ému	s	soeur
E	maison	S	chanter
a	alarme	v	vent
O	obstacle	z	zéro
o	beau	Z	jardin
u	coupable	Plosives	
y	punir	p	patte
2	creuser	t	tarte
9	peur	k	coup
@	petite	b	robe
Nasal Vowels		d	dire
e~	lundi	g	gauche
o~	rondeur	Nasals	
a~	temps	m	madame
Semi Vowels		n	nous
j	piétiner	N	campagne
w	soir	Others	
H	huile	G	boeing
l	long	_	silence
R	rond		

TAB. A.2 - *Les 35 phonèmes utilisés pour BREF.*

Annexe B

Calcul de $P(X|M)$

Nous rappelons en détail ici le calcul de la probabilité $P(X|M)$ afin de mettre en évidence les hypothèses simplificatrices qui doivent être mise en oeuvre pour utiliser les modèles de Markov cachés.

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

où

- $P(X|M)$ est la vraisemblance de la séquence d'observations X étant donné le modèle M,
- $P(M)$ est la probabilité a priori du modèle,
- $P(X)$ la probabilité a priori de la séquence de vecteurs acoustiques.
- **H1** On suppose que $P(M)$ peut être calculé indépendamment. 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.

Nous devons donc calculer :

$$P(X|M) = \sum_{\Gamma} P(C, X|M) = \sum_{\Gamma} P(Q_1^N, X|M)$$

où

- Γ = ensemble des chemins autorisés C dans le modèle M.
- $Q_1^N = \{q_I = q^0, q^1, ..., q^N, q^{N+1} = q_F\}$ est une séquence ordonnée de N états[1],
- q^n représentant l'état du HMM au temps n,
- q_k^n signifie que l'état q_k est visité au temps n.

Il vient alors :

$$\forall \{q_{l_1}^1, ..., q_{l_n}^n\} \in \Gamma$$
$$P(X|M) = \sum_{l_1=1}^{L} ... \sum_{l_n=1}^{L} P(q_{l_1}^1, ..., q_{l_n}^n, X|M)$$

1. Les états finaux q_F et initiaux q_I ne peuvent pas émettre de vecteurs acoustiques. Ils sont généralement utilisés pour initialiser les équations récurrentes.

Comme les événements q_k^n sont mutuellement exclusifs, cette probabilité s'écrit encore,

$$\forall n \in [1, ..., N] \quad P(X|M) = \sum_{l=1}^{L} P(q_l^n, X|M) \tag{B.2}$$

où $P(q_l^n, X|M)$ est la probabilité que la séquence de vecteurs acoustiques X ait été produite par M pendant qu'il était dans l'état q_l au temps n (L est le nombre d'états du modèle).

En posant $X_m^n = \{x_m, ..., x_n\}$ il est possible de décomposer l'équation (B.2) en appliquant $P(a, b|c) = P(a|b, c).P(b|c)$ selon :

$$P(q_l^n, X|M) = P(q_l^n, X_1^n|M).P(X_{n+1}^N|q_l^n, X_1^n, M) \tag{B.3}$$

- $P(q_l^n, X_1^n|M)$ est la probabilité d'avoir généré la séquence d'observation partielle X_1^n et d'être arrivé à l'état q_l au temps n, étant donné le modèle M. Elle est aussi appelée *Forward Probability ou probabilité avant.*

- $P(X_{n+1}^N|q_l^n, X_1^n, M)$ représente la probabilité d'observer la séquence partielle X_{n+1}^N étant donné que la séquence d'observation X_1^n a déjà été observée pendant qu'on est arrivé dans l'état q_l au temps n. Cette probabilité est aussi appelée *Backward Probability ou probabilité arrière.*

Ces deux probabilités peuvent être calculées de manière récurrente. La première probabilité peut être calculée selon l'équation suivante :

$$P(q_l^n, X_1^n|M) = \sum_{k=1}^{L} P(q_k^{n-1}, X_1^{n-1}|M).p(q_l^n, x_n|q_k^{n-1}, X_1^{n-1}, M) \tag{B.4}$$

Remarque : P(.) représente une probabilité globale ou cumulée alors que p représente une vraisemblance (contribution locale).

Comme $P(a, b|c) = P(a|b, c).P(b|c)$, nous pouvons écrire :

$$p(q_l^n, x_n|q_k^{n-1}, X_1^{n-1}, M) = p(q_l^n|q_k^{n-1}, X_1^{n-1}, M).p(x_n|q_l^n, q_k^{n-1}, X_1^{n-1}, M) \tag{B.5}$$

On retrouve ici dans l'équation (B.5) la propriété que les modèles de Markov peuvent être décrits comme deux processus Markoviens corrélés:

- un correspondant à la séquence d'états par l'intermédiaire de la probabilité $p(q_l^n|q_k^{n-1}, X_1^{n-1}, M)$,

- un autre correspondant à la séquence de vecteurs acoustiques par l'intermédiaire de la probabilité $p(x_n|q_l^n, q_k^{n-1}, X_1^{n-1}, M)$.

D'autres hypothèses interviennent dans l'utilisation des HMMs pour faciliter le calcul de la probabilité de l'équation (B.4).

- **H3** Les modèles de Markov sont supposés du premier ordre, ainsi la probabilité que la chaîne de Markov soit dans l'état q_l au temps n dépend uniquement de l'état de la chaîne de Markov au temps n-1 et est indépendant du passé.

  Ainsi on peut écrire

$$p(q_l^n|q_k^{n-1}, X_1^{n-1}, M) = p(q_l^n|q_k^{n-1}, M) \tag{B.6}$$

  $p(q_l^n|q_k^{n-1}, M)$ est encore appelé probabilité de transition et représente la probabilité d'aller de l'état q_k à q_l dans un modèle de Markov M.

- **H4** Les vecteurs acoustiques ne sont pas corrélés, la probabilité qu'un vecteur acoustique soit émis au temps n dépend uniquement de la transition de l'état q_k^{n-1} à q_l^n et est indépendante du passé. Ainsi il vient

$$p(x_n|q_l^n, q_k^{n-1}, X_1^{n-1}, M) = p(x_n|q_l^n, q_k^{n-1}, M) \tag{B.7}$$

  $p(x_n|q_l^n, q_k^{n-1}, M)$ est encore appelée probabilité d'émission sur transition, et représente la probabilité d'observer le vecteur acoustique x_n pendant une transition de l'état q_k à q_l. Cette hypothèse est réellement limitative car il est clair que les vecteurs acoustiques sont en réalité fortement corrélés.

- **H5** Bien que la probabilité d'émission sur transition soit quelquefois utilisée comme telle, elle est supposée dépendante uniquement de l'état courant pour réduire le nombre de paramètres.

$$p(x_n|q_l^n, q_k^{n-1}, M) = p(x_n|q_l) \tag{B.8}$$

  $p(x_n|q_l)$ est la probabilité d'émission.

Il vient donc l'équation récurrente de base des modèles cachés de Markov :

$$P(q_l^n, X_1^n|M) = [\sum_{k=1}^{L} P(q_k^{n-1}, X_1^{n-1}|M).p(q_l|q_k, M)].p(x_n|q_l) \tag{B.9}$$

De même il est possible de montrer que la seconde probabilité de l'équation (B.3) peut être calculée selon :

$$P(X_{n+1}^{N}|q_l^n, X_1^n, M) = \sum_{k=1}^{L} p(q_k|q_l).p(x_{n+1}|q_k).P(X_{n+2}^{N}|q_l^{n+1}, X_1^{n+1}, M) \tag{B.10}$$

Les équations (B.2) (B.3) (B.9) (B.10) permettent donc le calcul de $P(X|M)$, la probabilité que le modèle M ait généré la séquence de vecteurs acoustiques X à l'aide des paramètres du HMM.

En posant :

$$\forall l \in [1, L] \quad \alpha_n(l|M) = P(q_l^n, X_1^n|M) \tag{B.11}$$

il vient alors d'aprés (B.9):

$$\forall l \in [1, L] \quad \alpha_{n+1}(l|M) = [\sum_{k=1}^{L} \alpha_n(k|M).p(q_l|q_k, M)].p(x_{n+1}|q_l) \tag{B.12}$$

avec $\alpha_1(l) = p(x_1|q_l).p(q_l|q_1)$

De même posons :

$$\forall l \in [1, L] \quad \beta_n(l|M) = P(X_{n+1}^{N}|q_l^n, X_1^n, M) \tag{B.13}$$

il vient alors d'aprés (B.10):

$$\forall l \in [1, L] \quad \beta_n(l|M) = \sum_{k=1}^{L} p(q_k|q_l, M).p(x_{n+1}|q_k).\beta_{n+1}(k|M) \tag{B.14}$$

avec $\forall l \in [1, L] \quad \beta_N(l) = 1 \quad ou \quad 0$

Ainsi en utilisant ces deux dernières formulations il vient :

$$\forall n \in [1, N] \quad P(X|M) = \sum_{l=1}^{L} \alpha_n(l|M).\beta_n(l|M) \tag{B.15}$$

Où les termes $\alpha_n(l|M)$ et $\beta_n(l|M)$ peuvent être calculés de manière récurrente.

De plus en posant n = N , q_F état final et q_I état initial :

$$P(X|M) = \sum_{\forall l \in \Im} \alpha_N(l|M) = \alpha_{N+1}(q_F|M) = \beta_0(q_I|M) \tag{B.16}$$

C'est cette dernière formulation qui est trés souvent utilisée en reconnaissance de la parole, car elle ne nécessite que le calcul des termes α.

Annexe C

Probabilités a posteriori

Supposons qu'un réseau soit entraîné à classifier les exemples (X, q_l) où X correspond à l'entrée et q_l est la classe correcte associée[1]. Le réseau calcule l'activation de la k^{eme} sortie $(g_k(X))$. Les sorties valent δ_{lk} car elles sont imposées à 1 quand $l = k$ et 0 sinon. Il est alors possible de calculer l'erreur quadratique E :

$$E = \int X \sum_l p(X, q_l). \sum_k (\delta_{kl} - g_k(X))^2 dX \tag{C.1}$$

ce qui peut encore s'écrire :

$$E = \int X \sum_k \sum_l p(X, q_l).(\delta_{kl} - g_k(X))^2 dX \tag{C.2}$$

En décomposant en deux termes cette somme (un correspondant à $l = k$ et l'autre à $l \neq k$) il vient :

$$E = \int X \sum_k (p(X, q_k)(1 - g_k(X))^2 + \sum_{l \neq k} p(X, q_l).(0 - g_k(X))^2 dX \tag{C.3}$$

De plus comme

$$p(X) = \sum_k p(X, k) \tag{C.4}$$

Il vient

$$\sum_{l \neq k} p(X, q_l) = p(X) - p(X, q_k) \tag{C.5}$$

Soit en remplaçant dans (C.3) :

$$\begin{aligned} E = \int X \sum_k [p(X, q_k)(1 - 2.g_k(X) + g_k^2(X)) \quad + \\ (p(X) - p(X, q_k)).g_k^2(X)]dX \end{aligned} \tag{C.6}$$

De plus, en utilisant le fait que $P(a, b) = P(a|b).P(b) = P(b|a).P(a)$ il vient :

$$p(X, q_k) = p(q_k|X).p(X) = p(X|q_k)p(q_k) \tag{C.7}$$

Soit en remplaçant dans (C.3) :

1. Notons que le même vecteur d'entrée X peut appartenir à différentes classes q_l puisque les classes peuvent se chevaucher.

$$E = \int X \sum_k [p(q_k|X).p(X).(1 - 2.g_k(X) + g_k^2(X)) \quad + \\ (p(X) - p(X).p(q_k|X)).g_k^2(X)]dX \tag{C.8}$$

En mettant $p(X)$ en facteur,

$$E = \int X \sum_k p(X).[p(q_k|X) - 2.g_k(X).p(q_k|X) + g_k^2(X)]dX \tag{C.9}$$

Ce qui peut encore s'ecrire :

$$E = \int X \sum_k p(X).[p(q_k|X) + (p(q_k|X) - g_k(X))^2 - p^2(q_k|X)]dX \tag{C.10}$$

soit :

$$E = \int X \sum_k p(X).[(p(q_k|X) - g_k(X))^2 + (1 - p(q_k|X).p(q_k|X)]dX \tag{C.11}$$

L'équation (C.11) est minimum lorsque $g_k(X) = p(q_k|X)$, c'est à dire lorsque la sortie est égale à la probabilité a posteriori de la classe q_k étant donnné l'entrée X.

Hampshire et Pearlmutter [49] ont généralisé cette preuve montrant que la même conclusion peut être obtenue pour un réseau entraîné avec des critères d'erreurs du type moindes carrés ou entropiques.

Annexe D

Les Bases de données utilisées

D.1 Introduction

Nous allons décrire dans ce chapitre les différentes bases de données utilisées pour l'entraînement des modèles et les tests des différentes méthodes présentées dans cette étude. Comme nous l'avons précisé dans l'introduction, nous ne nous sommes pas cantonnés à tester les méthodes décrites dans ce travail sur des bases de données personnelles pour lesquelles des comparaisons avec d'autres systèmes n'auraient pas pu être effectuées. L'ensemble des bases de données utilisées couvre deux des principaux langages utilisés en reconnaissance de la parole (le français et l'anglais), les différents types de canaux de transmission (téléphone, microphone), le type d'élocution (parole continue, mots connectés, mots isolés), enfin les tailles des vocabulaires varient de 10 mots (chiffres connectés) à 65.000 mots (parole continue grand vocabulaire).

D.2 BDSONS

BDSONS [21, 33] est une base de données développée par le GRECO comportant 32 locuteurs.

Elle se divise en deux groupes :
Le premier groupe est appelé le groupe **Evaluation** et il contient pour chaque locuteur :

1. Un petit texte de cinq phrases.
2. Quatre cents chiffres isolés.
3. Deux cents suites de trois chiffres.
4. Cent suites de quatre chiffres.
5. Cent suites de cinq chiffres.
6. 432 lettres de l'alphabet français isolées.
7. 52 noms épelés en isolé.
8. 50 noms épelés en continu.

Le second groupe appelé **Acoustique** a été enregistré par 12 locuteurs et comporte :

- six cents groupements CVCV (consonne, voyelle, consonne, voyelle).

– cinquante phrases phonétiquement équilibrées.

La partie chiffres de la base de données a été l'objet de toute notre attention.

La base de données BDSONS a été échantillonnée à 16 kHz, le codage est réalisé sur 16 bits en linéaire.

Cette base de données sera le plus souvent utilisée pour des tests préliminaires de reconnaissance de chiffres isolés et connectés en français pour un vocabulaire très petit (10 mots).

D.3 Resource Management

La base de données Resource Management[1] [83] est un corpus de reconnaissance de parole continue en anglais américain pour un vocabulaire de 1000 mots. Ce vocabulaire de 1000 mots est spécifique à des conversations militaires navales. 160 locuteurs ont participé à l'élaboration de cette base de données couvrant ainsi un nombre important d'accents américains différents. L'enregistrement a été effectué sur microphone, le signal a été échantillonné à 16 kHz. Les données d'entraînement et de test officielles ont été définies pour permettre une comparaison des différents systèmes de reconnaissance. Nous possédons ainsi 5 différents types de données :

– l'entraînement composé de 3990 phrases.

– 4 fichiers tests composés de 300 phrases chacun (feb89, oct89, feb91 et sep92) correspondant à différentes campagnes d'enregistrement au cours de plusieurs années.

Avec le corpus, est fourni un modèle de langage de type grammaire word-pair de perplexité 60[2]. Cette base de données était couramment utilisée dans le monde de la parole pour comparer les différents algorithmes et méthodes utilisées en reconnaissance de la parole. Les taux de reconnaissance obtenus sur ce corpus étant maintenant bons, elle est beaucoup moins utilisée de nos jours.

D.4 Phonebook

Phonebook [82] est une base de données phonétiquement riche contenant des mots isolés énoncés sur ligne téléphonique. Elle contient plus de

1. Classiquement notée RM.
2. Voir le chapitre 2.

92.000 mots et le vocabulaire est constitué d'environ 8000 mots différents, avec une moyenne de 11 locuteurs pour chacun des mots (représentant environ 23 heures de parole). Chaque locuteur issu d'un groupe démographique représentatif de 1.300 personnes nées aux Etats Unis et parlant l'anglais effectue un enregistrement unique pendant lequel il énonce 75 mots.

La base de données contient 106 listes de mots, chacune composée de 75 ou 76 mots qui ont été prononcés par quelques locuteurs (typiquement 11).

Les locuteurs et les mots sont différents pour chacune des listes. Ces listes sont étiquetées $l_1 l_2$ avec

$$l_1 \in \{a, b, c, d, e\}$$

et

$$l_2 \in \{a, b, c, d, e, f, ..., x, y, z\}$$

(Sauf si $l_1 = e$ dans ce cas l_2 est égal à a ou b).

Il y a ainsi 106 listes de mots. Cette base de données étant très grande (23 heures de parole), elle a été scindée en deux lots pour l'entraînement, un lot pour la "cross-validation" [3] et un lot pour le test selon :

- un “petit” fichier d'entraînement correspondant à environ 5 heures de parole : tous les $*a$, $*h$, $*m$, $*q$, and $*t$ [4] listes de mots (21 listes de mots, soit environ 26000 mots).
- le grand fichier d'entraînement correspondant à environ 21 heures de parole : toutes les listes de mots exceptées celles présentes dans le fichier de "cross-validation" et de test (environ 80000 mots).
- le fichier “cross-validation” : toutes les listes $*o$,$*y$ (8 listes de mots, 8000 mots environ).
- le fichier “test” : tous les fichiers $*d$ et $*r$, c'est à dire 8 listes de mots (soit environ 6600 mots). Nous avons alors le choix entre reconnaître les 8 listes entièrement (correspondant à un dictionnaire de 600 mots) ou reconnaître chacune des listes indépendamment avec un dictionnaire de 75 mots. Dans le second cas, le taux de reconnaissance global sera la moyenne non pondérée des 8 taux de reconnaissance. Le vocabulaire global (parties entraînement et test) comprend environ 8000 mots.

3. Dans le cas où des modèles hybrides sont utilisés.
4. *a = aa,ba,ca,..,za.

Deux dictionnaires ont été utilisés pour obtenir la transcription phonétique des mots. Le premier est fourni avec la base PHONEBOOK et contient la transcription phonétique de tous les mots basée sur un fichier de 43 phonèmes. L'autre dictionnaire est le dictionnaire *CMU 0.4* contenant 110000 mots et utilisant 39 phonèmes (une partie du fichier de phonèmes de *TIMIT*). Certains des mots non présents dans le dictionnaire *CMU 0.4* ont été transcrits manuellement. Les résultats obtenus avec le dictionnaire *Phonebook* étant largement inférieurs à ceux obtenus avec le *CMU 0.4* la totalité des résultats présentés dans ce rapport utilise le dictionnaire *CMU 0.4* et le fichier de phonèmes décrit dans le tableau visible en Annexe 1.

D.5 BREF-80

Le corpus BREF [65] a été conçu pour disposer de suffisamment de parole pour développer et évaluer des systèmes de reconnaissance de parole continue (dépendants et indépendants du locuteur), mais aussi pour fournir un large corpus de parole continue pour l'acquisition de connaissances acoustico-phonétiques sur le français. Tous les textes enregistrés ont été extraits du journal "Le Mondë. Le sous-corpus BREF-80 a été spécialement sélectionné pour l'apprentissage de systèmes de reconnaissance indépendants du locuteur et du vocabulaire.

Le sous-corpus BREF-80 [65] contient 2 disques contenant chacun des enregistrements de 40 locuteurs. Seul le signal correspondant au microphone Shure SM10 est distribué sur ces deux disques. Au total, ces deux disques contiennent 5330 phrases prononcées par 80 locuteurs (i.e. une moyenne de 67 phrases par locuteur). Ce sous-corpus ne représente qu'une petite partie du corpus BREF puisque plus de 66500 phrases ont été enregistrées pour 120 locuteurs. Les phrases du sous-corpus BREF-80 ont cependant été sélectionnées pour couvrir la plupart des textes d'apprentissage de BREF, et ainsi conserver le plus grand nombre de contextes phonétiques avec le minimum de données. Les 5330 phrases sélectionnées correspondent en fait à 3747 textes différents, couvrant la presque totalité des 3877 textes d'apprentissage de BREF. Les données ont été divisées par nos soins en deux parties distinctes pour l'entraînement et le test des systèmes de reconnaissance. La partie entraînement est constituée de 3737 phrases énoncées par 56 locuteurs (environ 9 heures de parole) et la partie test est constituée de 144 phrases énoncées par 8 locuteurs (4 hommes et 4 femmes non présents dans la partie entraînement)[5]. Nous disposons en outre d'un autre corpus de test, plus

5. Ce petit corpus de test a été défini pour tester les différentes méthodes de reconnaissance et valider les modèles utilisés. Il correspond à un corpus de développement

important, regroupant 1046 phrases énoncées par 16 locuteurs[6] (8 hommes et 8 femmes).

D.6 Conclusion

Ces bases de données ont été utilisées pour l'entraînement et le test des systèmes de reconnaissance de la parole développés dans le laboratoire. Elles nous ont permis de couvrir assez largement les différents types de reconnaissance (parole continue, connectée, isolée ...), différents types de canaux de transmission (microphone, téléphone), et même deux langages différents (le français et l'anglais américain) et enfin différentes tailles de vocabulaire (de quelques mots à quelques dizaines de milliers de mots). L'utilisation des ces bases de données nous assure d'éviter de tirer des conclusions erronées du fait de la faible taille des données utilisées et de pouvoir comparer les résultats obtenus avec ceux d'autres laboratoires.

---

généralement utilisé pour régler les paramètres.

6. N'ayant pas participé à l'entraînement.

Bibliographie

[1] A. Anastasakos, R. Schwartz, and H. Shu, "Duration modeling in large vocabulary speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 1, pp. 628–631, IEEE, may 1995.

[2] B. Atal, "Effectiveness of linear prediction characteristics of a speech wave for automatic speaker identification and verification," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 305–308, 1991.

[3] X. Aubert, Haeb-Umbach, and H. .Ney, "Improvement in connected digit recognition using linear discriminant analysis and mixture densities," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. 648–651, 1993.

[4] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, and R. L. Mercer, "Further results on the recognition of a continuously read natural corpus," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, April 1980.

[5] L. R. Bahl, S. Balakrishnan-Aiyer, J. Bellegarda, M. Franz, P. Gopalakrishnan, D. Nahamoo, M. Novak, M. Padmanabhan, M. Picheny, and S. Roukos, "Performance of the IBM large vocabulary continuous speech recognition system on the ARPA wall street journal task," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Detroit, MI), pp. 41–44, may 1995.

[6] L. R. Bahl, P. F. Brown, P. V. de Souza, and R. L. Mercer, "Speech recognition with continuous-parameter hidden markov models," *Computer Speech and Language*, vol. 2, pp. 219–234, 1987.

[7] L. R. Bahl, F. Jelinek, and R. L. Mercer, "A maximum likelihood approach to continuous speech recognition," *IEEE Trans. on Patt. Anal. and Machine Intell.*, vol. PAMI-5, pp. 179–190, 1983.

[8] J. K. Baker, "The dragon system – an overwiew.," *IEEE Transactions on Acoustics Speech and Signal Processing*, vol. 1, pp. 24–29, February 1975. ASSP-23.

[9] R. Bakis, "Continuous speech word recognition via centisecond acoustic states," *In 91st Meeting of the Acoustical Society of America*, April 1976.

[10] L. E. Baum, "An inequaly and associated maximization technique in statistical estimation of probabilistic functions of markov processes.," *Inequalities 3*, pp. 1–8, 1972.

[11] J. R. Bellegarda and D. Nahamoo, "Tied mixture continuous parameter modeling for speech recognition.," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 38, pp. 2033–2045, December 1990.

[12] P. Beyerlein, M. Ullrich, and P. Wilcox, "Modeling and decoding of crossword context dependent phones in the philips large vocabulary continuous speech recognition system," in *Proceeding of the European Conference On Speech Communication and Technology.*, (Rhodes, Greece), pp. 1163–1166, Sept. 1997.

[13] J. M. Boite, H. Bourlard, B. D'hoore, S. Accaino, and J. Vantiegen, "Task independent and dependent training : Performance and comparison of hmm and hybrid hmm/mlp approaches," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 1, pp. 617–620, April 1994.

[14] R. Boite and M. Kunt, *Traitement de la parole.* Presses Polytechniques Romandes, 1987.

[15] H. Bourlard and N. Morgan, "A continuous speech recognition system embedding MLP into HMM," in *Advances in Neural Information Processing Systems 2* (D. Touretzky, ed.), pp. 186–193, Morgan Kaufmann, 1990.

[16] H. Bourlard and N. Morgan, *Connectionist Speech Recognition - A Hybrid Approach.* Kluwer Academic Publisher, 1994.

[17] H. Bourlard, N. Morgan., and C. Wellekens, "Statistical inference in multilayer perceptrons and hidden markov models with applications in

continuous speech recognition," *Neurocomputing Algorithms ,Architectures and Applications*, pp. 217–226, 1990. F. FogelmanSoulie and J. Herault (eds), NATO ASI Series.

[18] H. Bourlard, N. Morgan, C. Wooters, and S. Renals, "Cdnn : A context dependent neural network for continuous speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 2, pp. 349–352, March 1992.

[19] H. Bourlard and C. J. Wellekens, "Multilayer perceptrons and automatic speech recognition," in *Proceedings of IEEE First International Conference on Neural Networks, San Diego* (M. Caudill and C. Butler, eds.), pp. 407–416, IEEE, 1987.

[20] J. S. Bridle, D. M. Brown, and R. M. Chamberlain, "An algorithm for connected word recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 2, May 1982.

[21] R. Carré, R. Descourt, M. .Eskénozi, J. Mariani, and M. Rossi, "The french language database : defining, planning and recording a large database," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1984.

[22] P. L. Cerf and D. V. Compernolle, "A new variable frame rate analysis method for speech recognition," *IEEE Signal Processing Letters*, vol. 2, pp. 185–187, December 1994.

[23] P. Clarkson and R. Rosenfeld, "statistical language modeling using the cmu-cambridge toolkit," *Proceeding of the European Conference On Speech Communication and Technology.*, pp. 2707–2710, 1997.

[24] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash, "Hybrid neural network/hidden markov model continuous-speech recognition," in *icslp*, 1992.

[25] M. Cohen, H. Franco, N. Morgan, D. Rumelhart, and V. Abrash, "Multiple state context dependent phonetic modeling with mlps," *Proceeding of Speech Research Symposium 7*, June 1992.

[26] R. A. Cole, J. Mariani, H. Uszkoreit, A. Zuenen, and V. Zue, "Survey of the state of the art in human language technology," *Center for Spoken Language Understanding*, pp. 35–42, November 1995. Oregon Graduate Institute.

[27] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 357–366, August 1980.

[28] S. Deligne and F. Bimbot, "Inference of variable length linguistic and acoustic units by multigrams," tech. rep., Ecole National Superieure des Telecomunications, 97.

[29] A. P. Demster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm.," *Journal Of the Royal Statistical Society*, 1977. Series B 34.1-38.

[30] O. Deroo, F. Malfrere, and T. Dutoit, "Comparison of speech synthesis based vs. hybrid hmm/ann for text-to-speech alignment," *European Association for Signal Processing Conference*, September 1998. Rhodes, Greece.

[31] O. Deroo, C. Ris, S. Dupont, V. Fontaine, J. M. Boite, and H. Leich, "Hybrid hmm/ann systems for speaker independent continuous speech recognition system," *ProRisk 97*, November 1997.

[32] A.-M. Derouault, "Context-dependent phonetic Markov models for large vocabulary speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Dallas, TX), pp. 360–63, Apr. 1987.

[33] R. Descourt, J.-F. Serignat, O. Cervantes, and R. Carré, "Bdsons : Une base de données des sons du français," *12 th International Congress on Acoustics*, July 1986. Toronto, Canada.

[34] S. Dupont, H. Bourlard, and C. Ris, "Multi-stream speech recognition," *Tech Rep IDIAP-RR 96-07*, 1996. IDIAP, Martigny.

[35] S. Dupont, C. Ris, O. Deroo, and V. Fontaine, "Context independent and context dependent hybrid hmm/ann systems for vocabulary independent tasks," *Proceeding of the European Conference On Speech Communication and Technology.*, pp. 1947–1950, September 1997. Patras, Greece.

[36] T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O. V. D. Vreken, "The mbrola project : Toward a set of high quality speech synthesizers free of use for non commercial purposes," *International Conference on Speech and Language Processing*, vol. 3, pp. 1393–1396, 1996. Philadelphia.

[37] V. Fontaine, C. Ris, and J. Boite, "Non linear discriminant analysis for improved speech recognition," *Proceeding of the European Conference On Speech Communication and Technology.*, vol. 4, pp. 2071–2075, 1997.

[38] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, pp. 268–278, March 1973.

[39] H. Franco, M. Cohen, N. Morgan, D. Rumelhart, and V. Abrash, "Context-dependent connectionist probability estimation in a hybrid hidden markov model-neural net speech recognition system," *Computer Speech and Language*, vol. 8, pp. 211–222, JULY 1994.

[40] F. Freitag, E. Monte, and J. M. Salavedra, "Predictive neural networks applied to phoneme recognition," in *Proc. Eurospeech '97*, (Rhodes, Greece), pp. 2831–2834, Sept. 1997.

[41] S. Furui, "Speaker independent isolated word recognition based on emphasized spectral dynamics.," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 1991–1994, April 1986. Tokyo, Japan.

[42] S. Furui, "Speaker independent isolated word recognizer using dynamic features of speech spectrum.," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 34, pp. 52–59, April 1986. Tokyo, Japan.

[43] J. L. Gauvain and L. F. Lamel, "Speaker-independent phone recognition using bref," in *Proceedings of the DARPA Speech and Natural Language Workshop, Arden House, NY*, pp. 344–349, 1992.

[44] J. L. Gauvain, L. F. Lamel, G. Adda, and Adda-Decker, "Speaker independent continuous speech dictation," *Proc. Speech Communication*, November 1994.

[45] E. P. Giachin, "Phrase bigrams for continuous speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Detroit, MI), pp. 225–228, may 1995.

[46] B. Gosselin, *Application de réseaux de neurones artificiels à la reconnaissance optique de caractères manuscrits.* PhD thesis, Faculty Polytechnic of Mons, 1997.

[47] R. Haeb-Umbach, D. Geller, and H. Ney, "Linear discriminant analysis for improved large vocabulary continuous speech recognition.," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 1, pp. 13–16, 1992.

[48] V. Hamme, G. Gallopyn, V. Springel, W. D'Hoore, B. Butnaru, and H. Bourlard, "Acoustic features comparison and robustness tests of a realtime recognizer on a hardware telephone line simulator," *Proceedings of the International Conference on Spoken Language Processing.*, pp. –, 1994.

[49] J. Hampshire and B. Pearlmutter, "Equivalence proofs for multi-layer perceptron classifiers and the bayesian discriminant funcion.," in *Proceedings of the 1990 Connectionist Models Summer School.*, Morgan Kaufmann, 1990.

[50] H. Hermansky, "Perceptual linear predictive analysis of speech," *Journal of The Acoustic Soc. Am.*, vol. 87, 1990.

[51] H. Hermansky, A. Bayya, N. Morgan, and P. Khon, "Compensation for the effect of the communication channel in perceptual linear predictive (plp) analysis of speech," *Proceeding of the European Conference On Speech Communication and Technology.*, pp. 1367–1370, 1991. Genova, Italy.

[52] H. Hermansky and N. Morgan, "Rasta processing of speech.," *IEEE Trans. on Speech and Audio Processing*, vol. 2, pp. 578–589, 1994.

[53] M. Hochberg, G. D. Cook, S. Renals, A. J. Robinson, and R. S. Schechtman, "The 1994 abbot hybrid connectionist-hmm large vocabulary recognition system," *Spoken Language Systems Technology Wokshop*, pp. 170–176, January 1995.

[54] M. Hochberg, S. Renals, A. Robinson, and G. D. Cook, "Recent improvements to the abbot large vocabulary csr system," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, (Detroit, MI), pp. 69–72, May 1995.

[55] E.-F. Huang, F. K. Soong, and H. Wang, "The use of tree-trellis search for large vocabulary mandarin polysyllabic word speech recognition," *Computer Speech and Language*, vol. 8, January 1994.

[56] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, “Adaptative mixtures of local experts,” *Neural Computation*, vol. 3, no. 1, pp. 79–87, 1991.

[57] F. Jelinek, “Continuous speech recognition by statistical methods,” *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–536, 1976.

[58] M. I. Jordan and R. A. Jacobs, “Hierarchical mixtures of experts and the em algorithm,” *Neural Computation*, vol. 6, no. 2, pp. 181–214, 1994.

[59] B. H. Juang and L. R. Rabiner, “Mixture autoregressive hidden markov models for speech signals,” *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 33, no. 6, pp. 1404–1413, 1985.

[60] B. H. Juang, L. R. Rabiner, and J. G. Wilpon, “On the use of bandpass liftering in speech recognition,” in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Tokyo, Japan), pp. 765–768, Apr. 1986.

[61] D. Kershaw, T. Robinson, and S. Renals, “The 1995 abbot lvcsr system for multiple unknown microphones,” in *Proceedings of the International Conference on Spoken Language Processing*, vol. 3, (Philadelphia, PA), pp. 1325–1328, Oct. 1996.

[62] Y. Koenig, *REMAP: Recursive Estimation and Maximization of A Posteriori Probabilities in Transition-based Speech Recognition.* PhD thesis, University of California at Berkeley, 1996.

[63] G. Kuhn, “Connected word recognition with a recurrent network,” *Speech Communication*, vol. 9, no. 2, 1990.

[64] N. Kumar and A. G. Andreou, “On generalizations of linear discriminant analysis,” Tech. Rep. JHU/ECE-96-07, Electrical and Computer Engineering, John Hopkings University, 96.

[65] L. Lamel, J. Gauvain, and M. Eskenazi, “Bref, a large vocabulary spoken corpus for french,” in *Proceeding of the European Conference On Speech Communication and Technology.*, vol. 2, pp. 505–508, IEEE, 1991.

[66] K. J. Lang and A. H. Waibel, “A time-delay neural network architecture for isolated word recognition,” *Neural Networks*, vol. 3, pp. 23–43, 1990.

[67] K.-F. Lee, *Automatic Speech Recognition: Development of the SPHINX System.* Kluwer Academic Publisher, 1989. forward by D. Raj Reddy.

[68] K.-F. Lee, "Context dependent phonetic hidden markov models for speaker independent continuous speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 38, April 1990.

[69] R. P. Lippmann, "Review of neural networks for speech recognition," *Neural Computation*, vol. 1, pp. 1–38, 1989.

[70] U. C. London, "Multilingual speech input / output assesment methodology and standardisation," 1988. Esprit Project 1541, Definition Phase Final Report.

[71] F. Malfrère and T. Dutoit, "High-quality speech synthesis for phonetic speech segmentation," *Proceeding of the European Conference On Speech Communication and Technology.*, vol. 2, pp. 2631–2634, 1997.

[72] J. B. Marino and E. Monte, "Generation of multiple hypothesis in connected phonetic-unit recognition by a modified one-stage dynamic programming algorithm," *Proceeding of the European Conference On Speech Communication and Technology.*, pp. 1465–1472, December 1986.

[73] N. Morgan and H. Bourlard, "Continuous speech recognition using multilayer perceptrons with hidden markov models," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 413–416, april 1990.

[74] N. Morgan, H. Hermansky, H. Bourlard, P. Kohn, and C.Wooters, "Continuous speech recognition using plp analysis with multilayer perceptrons," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 49–52, may 1991.

[75] H. Murveit and R. Brodersen, "An integrated-circuit-based speech recognition system," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 2, September 1989.

[76] C. S. Myers and L. R. Rabiner, "A level building dynamic time warping algorithm for connected word recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 2, 1981.

[77] H. Ney, "The use of a one stage dynamic programming algorithm for connected word recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 1, 1984.

[78] H. Ney, D. Mergeland, A. Noll, and A. Paeseler, "A data driven organization of the dynamic programming beam search for continuous speech recognition," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 833–836, April 1987.

[79] N. Nilsson, *Problem-Solving Methods in Artificial Intelligence.* Mc Graw Hill, 1971.

[80] A. Paescler and H. Ney, "Continuous speech recognition using a stochastic language model," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 699–702, May 1989.

[81] D. B. Paul, "An efficient A^* stack decoder algorithm for continuous speech recognition with a stochastic language model," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (San Francisco, CA), pp. 25–28, Mar. 1992.

[82] J. Pitrelli, C. Fong, S. Wong, J. Spitz, and H. Leung, ""phonebook" : A phonetically-rich isolated-word telephone speech database.," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, 1995. Detroit, Michigan.

[83] P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallet, "The darpa 1000-word ressource management database for continuous speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 651–654, april 1998.

[84] L. R. Rabiner and B. Juang, "A tutorial on hidden markov models and selected applications in speech recognition.," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–285, 1989.

[85] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," *ASSP Magazine*, pp. 4–16, January 1986.

[86] L. R. Rabiner, J. G. Wilpon, and B. H. Juang, "A performance evaluation of a connected digit recognizer," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Dallas, TX), pp. 101–104, Apr. 1987.

[87] L. R. Rabiner and B.-H. Juang, *Fundamentals of Speech Recognition.* PTR Prentice Hall, 1993.

[88] S. Renals and M. Hochberg, "Efficient search using posterior phone probability estimates," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Detroit, MI), pp. 596–599, May 1995.

[89] S. Renals and M. Hochberg, "Efficient evaluation of the lvcsr search space using the noway decoder," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Atlanta, GA), pp. 149–152, May 1996.

[90] S. Renals, N. Morgan, and H. Bourlard, "Probability estimation by feed-forward networks in continuous speech recognition," Tech. Rep. TR-91-030, International Computer Science Institute, Berkeley, 91.

[91] S. Renals, N. Morgan, M. Cohen, and H. Franco, "Connectionist probability estimation is the Decipher speech recognition system," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 1, pp. 601–604, 1992.

[92] G. Riccardi, E. Bocchieri, and R. Pieraccini, "Non deterministic stokastic language models for speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 237–240, May 1995.

[93] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate bayesian a posteriori probabilities," *Neural Computation*, vol. 3, no. 4, pp. 461–483, 1991.

[94] A. J. Robinson, "An application of recurrent nets to phone probability estimation.," *Proceedings of the IEEE Transactions on Neural Network*, vol. 5, pp. 298–305, 1994.

[95] A. J. Robinson and F. Fallside, "Phoneme recognition from the timit database using reccurent error propagation networks," *Technical Report CUED/F-INFENG TR42*, 1990. Cambridge University UK.

[96] A. J. Robinson and F. Fallside, "A reccurent error propagation network speech recognition system," *Computer Speech and Language*, vol. 5, pp. 257–286, 1991.

[97] T. Robinson, "A real time reccurent error propagation network word recognition system," *Computer Speech and Language*, vol. 5, pp. 617–620, March 1992. New York, USA.

[98] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Parrallel distributed processing. exploration of microstructure of cognition," *Foundations D. E. Rumelhart and J. L. McClelland*, vol. 1, 1986. MIT Press.

[99] D. Rumelhart, G. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parrallel Distributed Processing. Exporation of the MicroStructure of Cognition*, vol. 1, 1986. MIT Press.

[100] M. J. Russel and A. E. Cook, "Experimental evaluation of duration modelling techniques for automatic speech recognition," *Proceedings International Conference on Acoustics Speech and Signal Processing*, pp. 2376–2379, April 1987.

[101] M. J. Russel, K. M. Ponting, S. M. Peeling, S. R. Browning, J. S. Briddle, R. K. Moore, I. Galiano, and P. Howell, "The arm continuous speech recognition system," *Proceedings International Conference on Acoustics Speech and Signal Processing*, pp. 69–72, April 1990.

[102] H. Sakoe, "Two level dp matching - a dynamic time warping based pattern matching algorithm for continuous speech recognition," *IEEE Transactions of the IECE of Japan*, vol. 3, 1979.

[103] L. Sauter, B. Flocon, and P. Lockwood, "Speaker independent speech recognition using demisyllable-based hidden markov models," *IEEE*, 1986.

[104] M. R. Schroeder, "Recognition of complex acoustic signal," *Life Science Research Report edited by T. H. Bullock*, 1977.

[105] R. Schwartz, Y. L. Chow, O. A. Kimball, S. Rouscos, M. Krasner, and L. Makhoul, "Context dependent modeling for acoustic-phonetic recognition of continuous speech," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 1, pp. 1205–1208, April 1985.

[106] R. Schwartz and Y. Chow, "The n-best algorithm : An efficient and exact procedure for finding the n most likely sentence hypotheses," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 1, April 1990.

[107] R. Schwartz and S. Austin, "A comparison of several approximate algorithms for finding multiple (n-best) sentence hypothesis," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 2, May 1991.

[108] S. A. Shamma, "Neural networks for speech processing and recognition," in *First International Conference on Neural Networks* (M. Caudill and C. Butler, eds.), (San Diego), IEEE, 1987.

[109] T. Shimizu, H. Yamamoto, H. Masataki, S. Matsunaga, and Y. Sagisaka, "Spontaneaous dialogue speech recognition using cross-word context-word context constrained word graphs," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, pp. 145–148, IEEE, may 1996.

[110] O. Siohan, "On the robustness of linear discriminant analysis as a preprocessing step for noisy speech recognition.," *Proceedings International Conference on Acoustics Speech and Signal Processing*, pp. 125–128, 1995.

[111] F. K. Soong and E. Huang, "A tree-trellis based fast search for finding the n best sentence hypotheses in continuous speech recognition," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Toronto, Canada), pp. 705–708, May 1991.

[112] V. Steinbis, "Sentence-hypotheses generation in a continuous speech recognition system," *Proceeding of the European Conference On Speech Communication and Technology.*, vol. 2, September 1989.

[113] C. Tapper and S. Das, "Memory and time improvements in a dynamic programming algorithm for matching speech patterns.," *Proceedings of the IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 583–586, 1978.

[114] J. Tebelskis, A. Waibel, B. Petek, and O. Schmidbauer, "Continuous speech recognition using linked predictive neural networks," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Toronto, Canada), pp. 61–64, May 1991.

[115] A. J. Viterbi, "Error bounds for convolution codes and an asymptotically optimum decoding algorithm," *Proceedings of the IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[116] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time–delay neural networks," *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, vol. 37, pp. 328–339, March 1989.

[117] R. L. Watrous, *Speech Recognition Using Connectionist Networks.* PhD thesis, University of Pennsylvania, Philadelphia, 1988.

[118] H. White, "Multilayer feedforward networks can learn arbitrary mappings : Connectionist non parametric regression with automatic and semi automatic determination of network complexity determination of network complexity," *Discussion Paper*, 1988. University of California, Department of Economics.

[119] J. G. Wilpon, B. H. Juang, and L. R. Rabiner, "An investigation on the use of acoustic sub-word units for automatic speech recognition," in *Proceedings of International Conference on Acoustics Speech and Signal Processing*, pp. 2071–2074, april 1987.

[120] P. C. Woodland, C. J. Leggetter, J. J. Odell, V. Valtchev, and S. Young, "The 1994 HTK large vocabulary speech recognition system," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing.*, (Detroit, MI), pp. 73–76, May 1995.

[121] G. Zavaliagkos, Y. Zhao, R. Schwartz, and J. Makhoul, "A hybrid segmental neural net/hidden markov model system for continuous speech recognition," *IEEE Transactions On Speech and Audio Processing*, vol. 2, pp. 151–160, January 1994.

[122] V. Zue, S. Seneff, and J. Glass, "Speech database development : Timit and beyond," *Speech Communication*, pp. 351–356, 90.