Reconnaissance automatique de la parole : progrès et tendances

Advances and trends in automatic speech recognition

Joseph MARIANI
LIMSI-CNRS
BP 133, 91403 ORSAY CEDEX, FRANCE

Joseph MARIANI a obtenu les titres de *Docteur-Ingénieur* en 1977, et de *Docteur ès Sciences* en 1982, de l'Université Paris VI.
Il a rejoint le LIMSI, laboratoire propre du CNRS à Orsay, comme *Chargé de Recherche* au CNRS en 1978, et est à présent *Directeur de Recherche*. Il a été responsable du groupe « Communication Parlée » du LIMSI de 1982 à 1985, World Trade Visiting Scientist à l'IBM T. J. Watson Research Center (Yorktown Heights, NY, USA) de 1985 à 1986, et est responsable, depuis 1987, du Département « Communication Homme-Machine » du LIMSI, qui regroupe les activités en Communication Parlée, Traitement du Langage Naturel, et Communication Non Verbale (Image, Vision et Geste). Il a été nommé Directeur du LIMSI en 1989.

J. Mariani a été lauréat du prix « Science et Défense » en 1985, et co-lauréat, en tant que responsable du groupe « Communication Parlée » du LIMSI, des prix « Systèmes de Communication Homme-Machine » de l'*Agence de l'Informatique* en 1984, et « Transfert Recherche-Industrie » du Ministère de l'Industrie en 1985. Il est membre de l'IEEE et de l'Association pour la Recherche Cognitive (ARC). Il a été vice-président du Groupe « Communication Parlée » de la Société Française d'Acoustique (SFA) de 1982 à 1986, et est Président de l'European Speech Communication Association (ESCA) depuis 1988.

Ses activités de recherche portent sur le domaine général de la communication homme-machine, et plus particulièrement sur la reconnaissance vocale.

RÉSUMÉ

Le but de cet article est de donner un aperçu des progrès récents obtenus dans le domaine de la reconnaissance automatique de la parole. Il traite essentiellement de la reconnaissance vocale, mais mentionne également les progrès réalisés dans d'autres domaines du Traitement Automatique de la Parole (Reconnaissance du Locuteur, Synthèse de Parole, Analyse et Codage), qui utilisent des méthodes voisines.

Ensuite, sont introduites les nouveautés méthodologiques qui ont permis des progrès suivant trois axes : des mots isolés vers la parole continue, de la reconnaissance monolocuteur vers la reconnaissance multilocuteur, et des petits vocabulaires vers les grands vocabulaires. Une mention spéciale est accordée aux améliorations qui ont été rendues possibles par les Modèles Markoviens, et, plus récemment, par les Modèles Connexionnistes. Ces méthodes ont conduit à des progrès obtenus concurremment suivant plusieurs axes, à des performances meilleures sur les vocabulaires difficiles, ou à des systèmes plus robustes. Quelques matériels spécialisés sont également décrits, ainsi que les efforts qui ont été consentis dans le but d'évaluer la qualité des systèmes de reconnaissance.

MOTS CLÉS

Communication Parlée. Technologies Vocales. Traitement Automatique de la Parole. Reconnaissance Vocale. Reconnaissance du Locuteur. Reconnaissance des Formes. Modèles Markoviens. Modèles Connexionnistes.

SUMMARY

This paper aims at giving an overview of recent advances in the domain of Speech Recognition. The paper mainly focuses on Speech Recognition, but also mentions some progress in other areas of Speech Processing (speaker recognition, speech synthesis, speech analysis and coding) using similar methodologies.

It first gives a view of what the problems related to automatic speech processing are, and then describes the initial approaches that have been followed in order to address those problems.

It then introduces the methodological novelties that allowed for progress along three axes : from isolated-word recognition to continuous speech, from speaker-dependent recognition to speaker-independent, and from small vocabularies to large vocabularies. Special emphasis centers on the improvements made possible by Markov Models, and, more recently, by Connectionist Models, resulting in progress simultaneously obtained along the above different axes, in improved performance for difficult vocabularies, or in more robust systems. Some specialised hardware is also described, as well as the efforts aimed at assessing Speech Recognition systems.

KEYWORDS

Speech Communication. Speech Technology. Automatic Speech Processing. Speech Recognition. Speaker Recognition. Markov Models. Connectionist Models.

1. Introduction

Le but de cet article est de donner un aperçu des progrès récents dans le domaine de la reconnaissance de la parole.

De façon très générale, on peut dire que, ces dernières années, la comparaison entre les *méthodes basées sur les connaissances* étendues d'experts humains, avec les heuristiques correspondantes, et les *méthodes auto-organisatrices,* utilisant des bases de données de parole et des algorithmes d'apprentissage automatique, avec peu de connaissances explicites, a tourné à l'avantage de ces dernières, qui ont obtenu des résultats nettement meilleurs lors d'essais comparatifs d'évaluation.

2. Les problèmes propres au traitement automatique de la parole

Plusieurs problèmes font que le traitement automatique de la parole est un domaine difficile, et non résolu actuellement :

A. Il n'y a pas de séparateurs, de silences entre les mots, comparables aux blancs dans le langage écrit.

B. Chaque son élémentaire (appelé également phonème) est modifié par con contexte (proches) : le phonème qui le précède, et celui qui lui succède. Cela est dû à la coarticulation : le fait que lorsqu'un phonème est prononcé, la prononciation du phonème suivant est préparée par un mouvement du conduit vocal. Cette cause donne à la parole un aspect « téléologique » [94]. D'autres modifications du signal correspondant à un phonème (mais de second ordre) seront dues au contexte plus large, comme la place du phonème dans la phrase.

C. Une très grande quantité de variabilité est présente dans la parole : variabilité intra-locuteur, due au mode d'élocution (voix chantée, criée, murmurée, enrhumée, enrouée, sous stress, bégaiement...) ; variabilité interlocuteur (timbres différents, voix masculines, féminines, voix d'enfants...) ; variabilité due au moyen d'acquisition du signal (type de microphone), ou à l'environnement (bruit, diaphonie...).

D. A cause de B et de C, il est nécessaire d'étudier, ou de traiter, une grande quantité de données si l'on veut découvrir, ou obtenir, ce qui fait un son élémentaire, en dépit des différents contextes, des différents modes d'élocution, des différents locuteurs, et des différents environnements. Un problème difficile pour le système est d'être capable de décider qu'un « a » prononcé par un adulte masculin est plus proche d'un « a » prononcé par un enfant, dans un mot différent, dans un environnement différent, et avec un autre microphone, qu'un « o » prononcé dans la même phrase par le même adulte masculin.

E. Le même signal renferme différents types d'informations (les sons eux-mêmes, la structure syntaxique de la phrase, sa signification, mais aussi l'identité du locuteur, et son état émotionnel (joyeux, en colère...)). Il faudra que le système se focalise sur le type d'information qui correspond à la tâche qu'il a à accomplir.

F. Il n'y a pas de règles précises actuellement pour formaliser ces connaissances aux différents niveaux du décodage (incluant la syntaxe, la sémantique, la pragmatique), ce qui fait qu'il est difficile de traiter la langue naturelle courante. De plus, ces différents niveaux semblent être étroitement imbriqués (la syntaxe et la sémantique par exemple). Heureusement, le problème mentionné en E signifie également que l'information dans le signal sera redondante, et que les différentes informations contenues dans le même signal coopéreront pour permettre la compréhension du signal, malgré les ambiguïtés et le « bruit » qui peuvent être trouvés à chaque niveau.

3. Premiers résultats sur un problème simplifié

Après quelques espoirs trop optimistes sur la difficulté de la reconnaissance vocale, similaires aux premières sous-estimations de la difficulté de la traduction automatique, une saine réaction à la fin des années 60 a été de constater l'importance du problème dans sa généralité, et d'essayer de résoudre tout d'abord un problème plus simple en introduisant des hypothèses simplificatrices. Au lieu d'essayer de reconnaître n'importe qui prononçant n'importe quoi, de n'importe quelle façon, et en parole courante, un premier problème a été isolé : reconnaître seulement une personne, utilisant un petit vocabulaire (de l'ordre de 20 à 50 mots), en lui demandant d'observer des courtes pauses entre les mots.

L'approche de base utilise deux phases : la phase d'apprentissage, et la phase de reconnaissance. Pendant la phase d'apprentissage, l'utilisateur prononce chacun des mots du vocabulaire. Le signal correspondant est traité au niveau dit « acoustique » ou « paramétrique », et l'information résultante, également appelée « image acoustique », « spectrogramme de parole », « sonogramme », « référence » ou « forme de référence », qui représente habituellement le signal en trois dimensions (le temps, la fréquence et l'amplitude), est conservée en mémoire, avec son étiquette correspondante. Pendant la phase de reconnaissance, un traitement similaire est fait au niveau « acoustique ». La forme correspondante est alors comparée avec toutes les formes de référence conservées en mémoire, en utilisant une distance appropriée. La référence pour laquelle la distance est la plus faible désigne le mot reconnu, et son étiquette peut alors être fournie comme un résultat. Si la distance est trop élevée, en fonction d'un seuil pré-défini, la décision de non-reconnaissance du mot prononcé est prise, qui permet ainsi de rejeter des mots qui n'appartiennent pas au vocabulaire.

Cette approche a conduit aux premiers systèmes commercialisés, qui sont apparus sur le marché au début des années 70, comme le VIP 100 de la société Threshold Technology Inc., qui a remporté un US National Award en 1972. A cause de ces simplifications, cette approche n'a pas à traiter les problèmes de segmentation de la parole

continue en mots (problème A), de coarticulation (puisqu'elle traite une forme complète correspondant à un mot toujours prononcé dans le même contexte — le silence (problème B)), de variabilité interlocuteur (problème C). Également, elle s'affranchit du problème relatif au traitement de la langue naturelle (problème F), dans la mesure où la faible taille du vocabulaire, et la prononciation par « mots isolés », empêche une parole naturelle ! Cependant, les problèmes dus à la variabilité intra-locuteur, à l'acquisition du signal, et à l'environnement demeurent entiers.

3.1. RECONNAISSANCE DE FORMES PAR PROGRAMMATION DYNAMIQUE

Dans la phase de reconnaissance, la distance entre la forme à reconnaître (la forme de test) et chacune des formes de références dans le vocabulaire doit être calculée. Chaque forme est représentée par une séquence de vecteurs régulièrement espacés dans le temps. Ces vecteurs peuvent être les valeurs de sortie d'un banc de filtres (analogiques, ou simulés par différents moyens, incluant la Transformée de Fourier Rapide [34]), les coefficients obtenus par un processus autorégressif comme la Prédic-

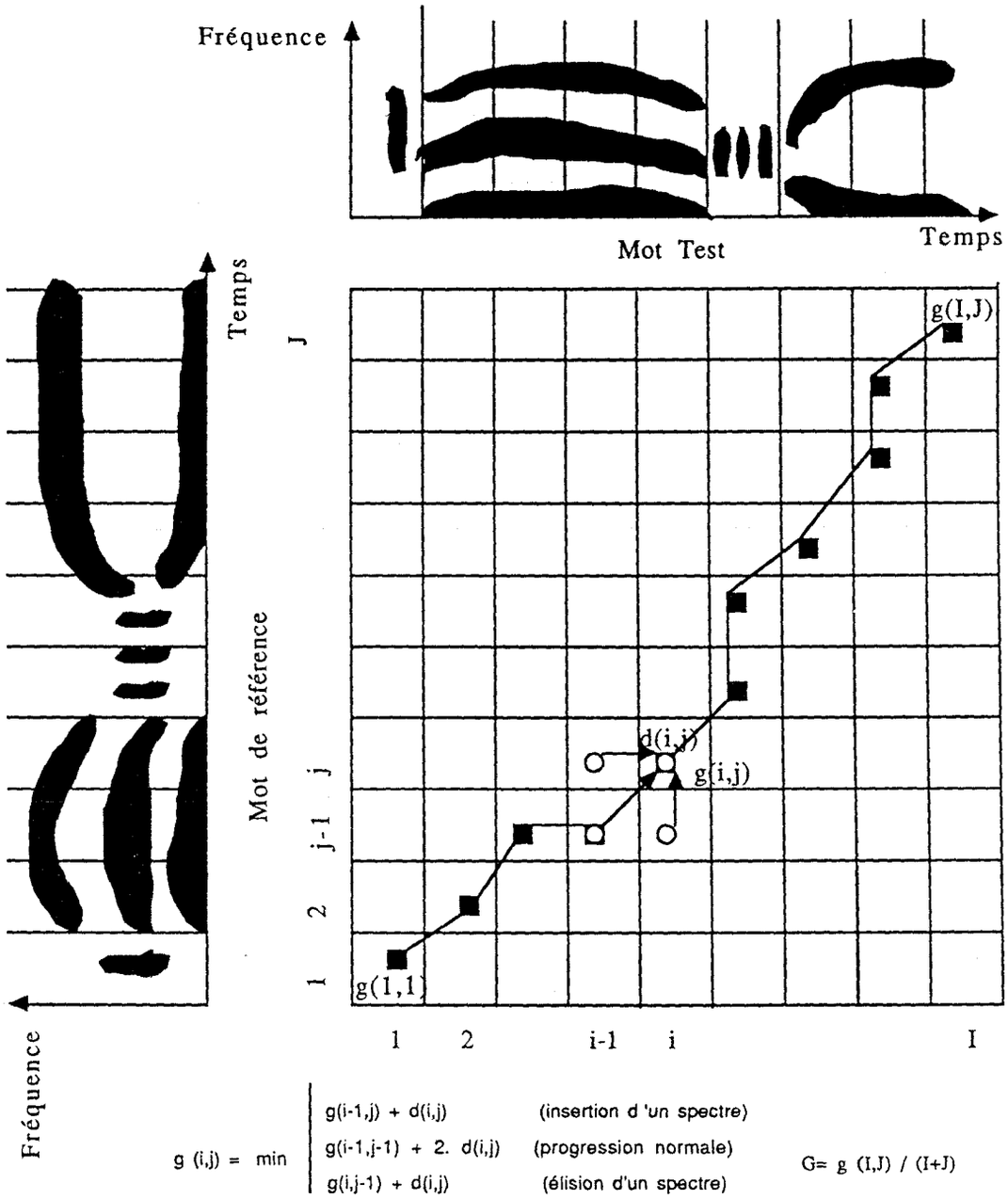

Figure 1. — Exemple d'Alignement Temporel Dynamique entre deux formes vocales (le mot « Paris » représenté par son spectrogramme schématisé).
G est la distance entre les deux prononciations du mot. $d(i,j)$ est la distance entre deux spectres de la référence et du test aux instants i et j. Un exemple d'équation locale de Programmation Dynamique est donné. Le chemin optimal est représenté par des carrés. Les distances locales qui interviennent dans le calcul de la distance cumulée $g(i,j)$ sont représentées par des cercles.

tion Linéaire (LPC) [162], ou des coefficients dérivés de ces méthodes, comme les coefficients cepstraux [19], ou être même obtenus par des modèles auditifs [52, 63, 99]. Typiquement, il s'agit de vecteurs de dimension 8 à 20 (également appelés spectres ou événements), obtenus toutes les 10 ms (pour des informations générales sur les techniques de traitement du signal vocal, consulter [122, 106, 134]).

Le problème est que lorsqu'un locuteur prononce deux fois un même mot, les spectrogrammes correspondants ne seront jamais exactement les mêmes. Il y aura des différences non linéaires dans le temps (rythme), la fréquence (timbre), et l'amplitude (intensité). Il est par conséquent nécessaire d'aligner les deux spectrogrammes de telle sorte que, lorsque la forme de test est comparée à la bonne forme de référence, les vecteurs correspondant aux mêmes sons dans les deux énonciations soient alignés. La distance entre les deux spectrogrammes sera calculée en fonction de cet alignement. Un alignement optimal peut être obtenu en utilisant la méthode de « Programmation Dynamique » (*fig.* 1). Si l'on considère la *matrice des distances* D obtenues en calculant les distance $d(i, j)$ (par exemple, la distance euclidienne) entre chaque vecteur de la forme test et la forme de référence, cette méthode fournira le cheminement optimal entre (1, 1) et (I, J) (où I et J sont respectivement les durées du test et de la référence), et la mesure correspondante de la distance entre les deux formes. Dans le cas de la reconnaissance vocale, cette méthode est également appelée *« Dynamic Time Warping »* (Alignement Temporel Dynamique), ou DTW, dans la mesure où le résultat principal est d'aligner les axes temporels. La Programmation Dynamique a été introduite par R. Bellman [14], et appliquée pour la première fois à la parole par les chercheurs soviétiques T. Vintsjuk et G. Slutsker à la fin des années 60 [170, 165].

3.2. LA PAROLE ET L'INTELLIGENCE ARTIFICIELLE : LE PROJET ARPA-SUR

Une approche différente, principalement basée sur des techniques d'intelligence artificielle a été initialisée en 1971, dans le cadre du projet ARPA-SUR [119]. L'idée sous-jacente était que l'utilisation des connaissances de haut niveau (lexique, syntaxe, sémantique, pragmatique) peut produire un taux de reconnaissance acceptable, même si le taux de reconnaissance phonémique initial est faible [74]. Le but était la reconnaissance monolocuteur, en parole continue, d'un vocabulaire de 1 000 mots (improprement appelé « compréhension de parole continue » pour la seule raison que les niveaux supérieurs sont utilisés). Plusieurs systèmes ont été réalisés à la fin du projet, en 1976. A Carnegie-Mellon University (CMU) furent conçus le système DRAGON [4], à base d'approche Markovienne, ainsi que les systèmes HEARSAY I et HEARSAY II, basés sur l'utilisation d'un modèle de « Tableau Noir » *(Blackboard Model)*, où chaque source de connaissance peut aller lire et écrire des informations pendant le décodage, avec une stratégie heuristique, et le système HARPY, regroupant des éléments des systèmes DRAGON et HEARSAY. A BBN, les systèmes SPEECHLIS et HWIM furent développés. SDC produisit également un système [80]. Bien que les performances du cahier des charges initial (qui, de fait, était plutôt vague) furent atteintes par au moins un système (HARPY), les algorithmes nécessitaient tant de puissance de calcul, à un moment où cela était cher, et étaient si peu flexibles et si peu robustes qu'il n'y eut pas de retombées industrielles. De fait, une des conclusions principales fut qu'il était nécessaire d'obtenir un meilleur décodage acoustico-phonétique [81] !

4. Améliorations suivant chacun des trois axes

A partir de la méthode de base de reconnaissance globale par mots isolés, des progrès ont été réalisés qui traitent les trois problèmes différents : la taille de la population qui peut utiliser le système, le débit d'élocution, la taille du vocabulaire.

4.1. DU MONOLOCUTEUR AU MULTILOCUTEUR

De manière à permettre à n'importe quel locuteur d'utiliser un système de reconnaissance, une approche par multiréférences a été tentée. Chaque mot du vocabulaire est prononcé par une population importante, composée de sujets masculins et féminins, ayant différents timbres et différents dialectes. La distance entre les différentes prononciations d'un même mot est calculée en utilisant l'algorithme de programmation dynamique. Un algorithme de classification automatique (tel que les K-moyennes) est utilisé pour déterminer des nuages de points correspondant à un certain type de prononciation du mot. Le centroïde de chaque nuage est choisi pour être la forme de référence pour ce type de prononciation (*fig.* 2). Chaque mot est alors représenté par plusieurs références. La reconnaissance est réalisée comme dans le cas monolocuteur, avec, éventuellement, des processus de décision plus sophistiqués (comme le KNN (K-nearest neighbors, K plus proches voisins)) [35].

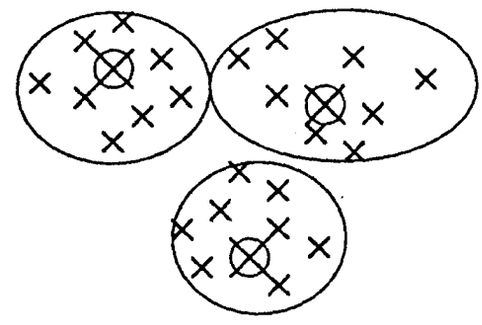

Figure 2. — Illustration de la classification automatique.
Chaque croix est un mot. La distance entre les croix représente la distance, au sens de la programmation dynamique, entre les mots. Chaque nuage de points est représenté par son centroïde (cercles).

4.2. DE LA RECONNAISSANCE DE MOTS ISOLÉS À LA RECONNAISSANCE DE MOTS ENCHAÎNÉS

Afin de permettre à un utilisateur de parler continûment (sans marquer de pause entre les mots), il faut résoudre plusieurs problèmes : déterminer combien de mots il y a dans la phrase et où se trouvent les frontières entre chacun d'eux. Si l'apprentissage a été fait en mode isolé, les formes acoustiques correspondant au début et à la fin des mots seront modifiées à cause du problème du contexte phonémique dû à la fin du mot précédent, et au début du mot suivant. Les deux premiers problèmes sont résolus en utilisant des méthodes qui généralisent la programmation dynamique utilisée dans le cas des mots isolés, telles que le *« Two-Level DP matching »* proposé par H. Sakoe [151], le *« Level building »* proposé par C. Myers et L. Rabiner [113], le *« One-Pass DP »* proposé par J. Bridle [25], également appelé *« One-stage DP »* par H. Ney [120]. Il apparaît en fait que l'approche initiale de la programmation dynamique comme elle a été décrite par T. Vintsjuk en 1968 [170] possédait déjà son extension à la reconnaissance de mots enchaînés [91]. Pour traiter le second problème, la méthode d'apprentissage en contexte (*« embedded training »*) a été proposée [137]. Chaque mot est tout d'abord prononcé de façon isolée. Puis on le prononce au sein d'une phrase connue du système. Les références apprises en isolé vont alors être utilisées pour segmenter la phrase en ces différents constituants de manière optimale, et extraire les images acoustiques des mots « en contexte », qui seront ajoutées comme formes de référence pour le mot.

La technique de détection de mots dans la parole continue (*« Word Spotting »*) est très similaire, et utilise les mêmes techniques de Programmation Dynamique. Mais il convient ici de rejeter également les mots qui ne sont pas dans le vocabulaire. Des résultats récents sur la détection de mots multilocuteurs ont donné 61 % de détection correcte pour de la parole prononcée dans un environnement silencieux, et 44 % lorsque du bruit Gaussien était ajouté, correspondant à un rapport Signal/Bruit de 10 dB, avec un vocabulaire de 20 mots (longs de une à trois syllabes), le taux de fausse alarme étant ajusté à 10 fausses alarmes par heure d'enregistrement [20].

Une syntaxe peut être utilisée pendant le processus de reconnaissance. La syntaxe représente les suites de mots qui sont permises suivant le langage qui correspond à la tâche à réaliser en utilisant le système de reconnaissance. Le rôle de la syntaxe est donc de déterminer quels sont les mots qui peuvent suivre un mot donné (sous-vocabulaire) au sein d'une phrase, accélérant ainsi la reconnaissance en réduisant la taille du vocabulaire à reconnaître à chaque étape, et améliorant les performances en faisant la différence entre des mots qui, acoustiquement, sont proches, mais n'appartiennent pas au même sous-vocabulaire, et ne sont donc pas « en compétition » à un instant donné. L'introduction de telles grammaires dans la procédure de décodage peut être plus ou moins difficile, suivant l'algorithme de programmation dynamique pour la parole continue qui est utilisé. La plupart des syntaxes qui sont utilisées dans les systèmes de reconnaissance globale correspondent à des langages de commande simples (grammaire régulière, ou hors-contexte, introduites manuellement par l'utilisateur du système).

Il apparaît que plus l'apprentissage est important, meilleure est la qualité de la reconnaissance. Afin d'améliorer l'apprentissage, plusieurs techniques ont été utilisées, comme l'apprentissage en contexte, déjà mentionné, l'apprentissage multiréférences, où plusieurs références sont retenues pour chacun des mots, avec les mêmes techniques de classification automatique pour représenter les variétés intralocuteurs que celles qui sont utilisées pour représenter les variétés interlocuteurs dans la reconnaissance multilocuteur multiréférences, et l'apprentissage « robuste » [136].

4.3. DES PETITS VOCABULAIRES AUX GRANDS VOCABULAIRES

Augmenter la taille du vocabulaire soulève un certain nombre de problèmes : dans la mesure où chaque mot est représenté par son image acoustique, la taille de mémoire nécessaire devient très grande. De plus, puisque la comparaison entre la forme à reconnaître et chacune des formes de référence est faite séquentiellement, le temps de calcul, lui-aussi, augmente considérablement. Si le locuteur doit faire l'apprentissage en prononçant tous les mots du vocabulaire, la tâche devient rapidement très fastidieuse. Traiter un grand vocabulaire a aussi pour conséquence que beaucoup de mots seront acoustiquement très voisins, augmentant ainsi le taux d'erreurs. Enfin, cela entraîne le fait que le locuteur sera encouragé à utiliser une façon naturelle de s'exprimer, sans respecter des règles de syntaxe trop contraignantes, et sans pouvoir mémoriser les mots qui sont dans le vocabulaire, et ceux qui n'y sont pas. Pour traiter ces différents problèmes, plusieurs méthodes ont été proposées.

4.3.1. Quantification vectorielle [55, 51, 101]

Dans le domaine du traitement automatique de la parole, cette méthode a tout d'abord été utilisée pour le codage de parole à très faible débit [95]. Si l'on considère une quantité de parole suffisante prononcée par un locuteur, la méthode consiste à calculer les distances (telles que la distance euclidienne) entre chaque vecteur des spectrogrammes correspondants, et à utiliser un algorithme de classification automatique pour déterminer des nuages correspondant à un type de spectre, qui seront représentés par leur centroïde (appelé « prototype » ou *« codeword »*). L'ensemble de ces prototypes est appelé « répertoire » ou *« codebook »*. Dans la phase d'apprentissage chaque spectre est alors reconnu par rapport au répertoire de prototypes. De cette façon, au lieu de représenter le mot par une séquence de vecteurs, il est représenté par une séquence de nombres (ou étiquettes) correspondant aux prototypes. Une *mesure de distorsion* peut être obtenues en calculant la distance moyenne entre les spectres et les spectres prototypes les plus voisins. Pratiquement, si la taille du répertoire est de 256, ou moins (c'est dire adressable par un octet), et si chaque composante du vecteur est codée sur un octet, la réduction d'information est égale à la

dimension du vecteur. Le temps de calcul nécessaire à la reconnaissance de grands vocabulaires est également réduit, dans la mesure où, pour chaque spectre du mot à reconnaître, il ne faut calculer que 256 distances, et non pas toutes les distances avec tous les spectres de toutes les références. De plus, les distances entre spectres prototypes peuvent être calculées pendant l'apprentissage, et conservées dans une matrice de distances. Ces répertoires peuvent tenir compte non seulement de l'information spectrale, mais aussi de l'énergie, des variations spectrales, ou des variations de l'énergie dans le temps. Tout ceci peut être représenté soit par un seul répertoire contenant des « supervecteurs », construits en ajoutant les différents types d'information [160], soit par différents répertoires pour chaque type d'information, approche appliquée avec succès à la reconnaissance du locuteur [166], et à la reconnaissance vocale [57]. Les répertoires peuvent enfin être construits à partir de signaux de parole de différents locuteurs (répertoire multilocuteur) [87].

On pourra remarquer que des méthodes similaires ont été utilisées dans le passé (sous la dénomination de « modèles centisecondes » [89]). Le défaut de ces approches était que l'on étiquetait les vecteurs avec une étiquette linguistique (un phonème), prenant ainsi une décision trop hâtive. Le principe de la quantification vectorielle a inspiré beaucoup d'idées. L'une d'entre elles a été que chaque mot avait peut-être un ensemble spécifique de prototypes sans qu'il soit nécessaire de prendre en compte la suite chronologique de ces prototypes. Même si des mots contiennent les mêmes phonèmes, mais dans un ordre différent, les transitions entre ces phonèmes seront différentes, et les prototypes correspondant à ces transitions seront donc différents, faisant ainsi la différence entre les mots. Pendant l'apprentissage, un répertoire est construit pour chaque mot de référence. Le processus de reconnaissance consiste alors à reconnaître tout simplement les vecteurs au fur et à mesure de leur acquisition, et à choisir la référence qui donne la mesure de distorsion moyenne la plus faible avec le test [161]. Une approche plus raffinée a consisté à segmenter les mots en plusieurs sections, de façon à refléter partiellement les séquences chronologiques des vecteurs pour les mots qui ont plusieurs phonèmes en commun [27]. Ce raffinement augmente le temps de reconnaissance, sans pour autant donner de meilleurs résultats que ceux obtenus avec une approche à base de programmation dynamique.

4.3.2. Unités de décision plus courtes que le mot

Une autre façon de réduire la quantité de mémoire nécessaire est d'utiliser des unités de décision plus courtes que le mot *(subword units)*. Les mots seront ainsi reconnus comme la concaténation de ces unités, en utilisant un algorithme de reconnaissance de « mots » enchaînés par programmation dynamique. Il faut choisir ces unités de telle sorte qu'elles ne soient pas trop affectées par le problème de coarticulation à leurs frontières. Mais, en même temps, il ne faut pas qu'elles soient trop nombreuses. Des exemples de telles unités sont les phonèmes [168], les diphonèmes ou diphones [102, 153, 32, 2, 154], les syllabes [62, 173, 50], les demi-syllabes [149, 144], les dissyllabes [164].

Mot graphémique: émigrante

Mot phonémique: $emigRãt$

phonèmes: $ e m i g R ã t $
diphonèmes: $e em mi ig gR Rã ãt t$
syllabes: $e mi gRãt$
demi-syllabes: $e em mi ig gRã ãt t$
disyllabes: $e emi igRã ãt$

Figure 3. — Représentation d'un mot par des unités plus courtes.
(Le mot est « émigrante », $ désigne le silence).

D'autres approches tendent à utiliser des unités sans signification linguistique, par exemple des segments, obtenus à l'aide d'un algorithme de segmentation. Cette approche, désignée sous le nom de quantification segmentale, ou matricielle, est très semblable à la quantification vectorielle, mais utilise une distance qui doit tenir compte de l'alignement temporel, si les segments ne sont pas de taille identique [26, 147, 88]. De la même manière, plusieurs tentatives de représentations des références par des réseaux de segments peuvent être répertoriées pour la reconnaissance de chiffres isolés multilocuteur [78], de chiffres enchaînés [28], ou de l'alphabet [84].

4.3.3. Compression temporelle

La compression temporelle peut aussi réduire la quantité d'information [79, 48]. L'idée est de compresser (linéairement ou pas) les instants de stabilité qui peuvent avoir des durées très variables suivant le débit d'élocution, mais en conservant tous les vecteurs lors des instants de transition, effectuant ainsi une transformation de l'espace du temps vers l'espace de la variation. Un algorithme comme le VLTS *(Variable Length Trace Segmentation)* [49] divise environ par deux la quantité d'information à traiter. Elle permet également d'obtenir de meilleurs résultats lorsque le débit d'élocution à la reconnaissance est très différent de celui observé à l'apprentissage (des équations classiques de programmation dynamique n'acceptent pas, par exemple, des variations de débit d'élocution dans un rapport supérieur à deux, ce qui est pourtant souvent constaté entre une prononciation en mots isolés et une prononciation en parole continue). Cependant, si la durée des phonèmes est elle-même porteuse de sens, cette information pourra être perdue.

4.3.4. Reconnaissance en deux passes

Afin d'accélérer la reconnaissance, on peut la réaliser en deux passes : on effectue tout d'abord une comparaison grossière, mais rapide, dans le but d'éliminer les mots du vocabulaire qui sont très différents du mot à reconnaître,

avant d'appliquer une méthode de comparaison optimale (programmation dynamique ou algorithme de Viterbi) sur le sous-vocabulaire restant. Dans ce cas, le but n'est pas seulement d'obtenir le mot correct, mais aussi d'éliminer autant de mots-candidats que possible (sans éliminer le bon...). Des approches simples ont été utilisées, comme la sommation des distances sur la diagonale de la matrice des distances [50]. D'autres approches sont basées sur la quantification vectorielle sans alignement temporel, que la méthode de reconnaissance soit basée sur la reconnaissance par comparaison de formes [103] ou sur la modélisation stochastique (*« Poisson Polling »*) [8]. L'utilisation d'un classifieur phonémique, basé sur des catégories phonémiques grossières [46] ou classiques [16], suivi d'une comparaison entre le treillis phonémique reconnu avec les mots de référence du lexique sous leur forme phonémique par programmation dynamique a également été proposée.

4.3.5. Adaptation au locuteur

L'adaptation des références d'un locuteur à un nouveau locuteur peut être obtenue par l'intermédiaire de leurs répertoires respectifs, dans le cas où la quantification vectorielle est utilisée. Le locuteur de référence prononce plusieurs phrases qui sont quantifiées vectoriellement avec son répertoire de spectres-prototypes. Le nouveau locuteur prononce les mêmes phrases, qui sont également quantifiées vectoriellement avec son propre répertoire. L'alignement temporel des deux ensembles de phrases permet d'obtenir une mise en correspondance (*« mapping »*) des deux répertoires. Cette méthode de base a plusieurs variantes [160, 21, 44].

La plupart des progrès obtenus dans le cadre des méthodes de reconnaissance par comparaison de formes ont porté sur un des aspects du problème. Quelques systèmes qui traitent de manière conjointe deux aspects différents peuvent également être recensés, tel le système Conversant d'AT&T [157], qui permet la reconnaissance multilocuteur de chiffres enchaînés par téléphone, avec une approche de programmation dynamique multiréférences. Des progrès plus sensibles ont été obtenus en utilisant des techniques plus élaborées : les Modèles Markoviens Cachés (HMM, Hidden Markov Models), et les Modèles Connexionnistes.

5. Les modèles markoviens cachés

5.1. PRINCIPE

Alors que dans les approches par comparaison de formes *(pattern matching)* décrites précédemment, la référence était représentée par la forme elle-même qui était conservée en mémoire, l'approche par Modèle Markovien contient un niveau plus élevé d'abstraction, dans la mesure où la référence est représentée par un modèle [130, 140]. La reconnaissance de la forme à reconnaître s'effectue ainsi par comparaison avec les modèles de référence. Les premières utilisations de cette approche pour la reconnaissance de la parole peuvent être signalées à CMU [4], IBM [66] et, semble-t-il, au très secret *« Institute for Defence Analysis »* (IDA) [129].

Dans une approche stochastique, si l'on considère un signal acoustique A, le processus de reconnaissance peut être décrit comme le calcul de la probabilité $P(W \mid A)$ qu'une suite de mots (ou phrase) W corresponde au signal acoustique A, et la détermination de la suite de mots qui maximise cette probabilité. En utilisant la formule de Bayes, $P(W \mid A)$ peut se réécrire :

$$P(W \mid A) = P(W) \cdot P(A \mid W)/P(A)$$

et

$$P(\hat{W} \mid A) = \max_i \frac{P(A \mid W_i) \cdot P(W_i)}{P(A)}$$

où $P(W)$ est la probabilité a priori de la suite de mots W, $P(A \mid W)$ est la probabilité du signal acoustique A, étant donné la suite de mots W, et $P(A)$ est la probabilité du signal acoustique (qu'on peut supposer ne pas dépendre de W dans les modèles les plus simples). Il est donc nécessaire de considérer $P(A \mid W)$ (appelé *Modèle Acoustique*), et $P(W)$ (appelé *Modèle Linguistique*). Chacun de ces modèles peut être représenté comme un Modèle Markovien [6]. Nous considérerons tout d'abord le modèle acoustique.

5.1.2. Approche discrète de base

Dans cette approche, chaque entité acoustique à reconnaître, chaque mot de référence par exemple, est représentée par une machine d'états finis, appelée également machine Markovienne, composée d'états et d'arcs entre ces états. Une *probabilité de transition* a_{ij} est attachée à l'arc allant de l'état i à l'état j, qui représente la probabilité que cet arc soit emprunté. La somme des probabilités attachées aux arcs issus d'un état i est égale à 1. On utilise d'autre part une *probabilité d'émission* (*output probability*) $b_{ij}(k)$ qu'un symbole k appartenant à un alphabet fini puisse être émis (et observé) lorsqu'on emprunte l'arc allant de l'état i à l'état j. Dans certaines variantes, cette probabilité d'émission est attachée à un état, et non pas à un arc. Lorsqu'on utilise le codage vectoriel, la distribution de ces probabilités d'émission discrètes est la distribution des probabilités des prototypes. La somme des probabilités dans la distribution est égale à 1 (*fig.* 4). Dans un Modèle Markovien du premier ordre, on suppose que la probabilité pour qu'une chaîne de Markov soit dans un état particulier au temps t dépend uniquement de l'état dans lequel elle était au temps $t-1$ et du symbole observé depuis, et que la probabilité d'émission entre les temps $t-1$ et t dépend uniquement de l'arc qui est emprunté au temps t.

5.1.3. Modèles continus

Nous venons de présenter ce qu'on appelle un « Modèle Markovien Caché Discret ». Un autre type de modèle Markovien est le modèle Markovien continu. Dans ce cas, les distributions de probabilités sur les arcs sont remplacés

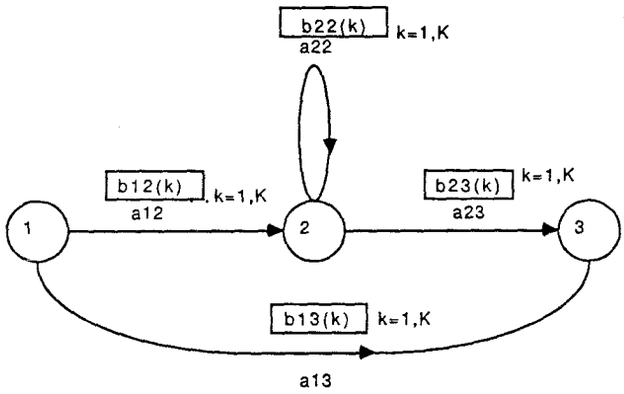

Figure 4. — Exemple de Modèle Markovien Caché.
Les distributions des probabilités d'émission $b_{ij}(k)$ sont placées dans des rectangles. a_{ij} est la probabilité de transition. Ce modèle gauche-droite a trois états et quatre arcs.

par un modèle du spectre continu sur cet arc (ou fonction de densité de probabilité, *(probability density function (pdf)*)). *Un modèle* communément utilisé est la densité Gaussienne multidimensionnelle *(« multivariate Gaussian density »)* [125], qui est représentée par un vecteur moyen, et la matrice de covariance (éventuellement diagonale). L'utilisation d'un mélange de Gaussiennes *(Gaussian Mixture Density)* semble être plus appropriée [140, 70, 142, 127]. L'utilisation d'une méthode voisine, incluant quelques simplifications *(Laplacian Mixture Density)* a permis l'obtention de bons résultats, avec un temps de calcul réduit [121]. Quelques essais visant à comparer les HMMs discrets et continus ont été effectués. Il semble que seuls des modèles continus complexes permettent d'obtenir de meilleurs résultats que les modèles discrets, reflétant le fait que, du moins avec un apprentissage de type « Maximum de Vraisemblance » *(Maximum Likelihood)*, il est nécessaire que le modèle complet soit correct si l'on veut obtenir de bons résultats de reconnaissance [7]. Mais les modèles continus complexes nécessitent une grande quantité de calculs qui peut s'avérer prohibitive.

Le nombre d'états, le nombre d'arcs, les états initiaux et finaux pour chaque arc sont choisis par le concepteur du système. Les paramètres du modèle (probabilités de transition et probabilités d'émission) doivent être obtenus par apprentissage. Trois problèmes doivent être traités :

— Le problème de l'**évaluation** (quelle est la probabilité qu'une suite d'étiquettes a été produite par un certain modèle ?). Cela peut être obtenu en utilisant l'algorithme *Forward,* qui donne l'Estimation par Maximum de Vraisemblance (*Maximum Likelihood Estimation,* MLE) que la séquence a été produite par le modèle.

— Le problème du **Décodage** (quelle séquence d'états a produit la séquence d'étiquettes ?). Cela peut être obtenu par l'algorithme de Viterbi, qui est très voisin de la programmation dynamique [171].

— Le problème de l'**apprentissage** (ou **entraînement**) (comment obtenir les paramètres du modèle, à partir d'une séquence d'étiquettes ?). Cela peut être obtenu par l'algorithme *Forward-Backward* (également appelé Baum-Welch) [12], lorsque l'apprentissage est basé sur le Maximum de Vraisemblance.

5.2. APPRENTISSAGE

5.2.1 Initialisation

L'initialisation des paramètres du modèle doit être réalisée avant le début du processus d'apprentissage. Un corpus d'apprentissage étiqueté à la main peut être utilisé. S'il existe suffisamment de données pour réaliser l'apprentissage, une distribution uniforme sera suffisante pour des modèles d'entités homogènes, tels les phonèmes, avec des modèles markoviens discrets [87]. Dans le cas de modèles de mots, ou pour des modèles Markoviens continus, il est nécessaire d'utiliser des techniques plus sophistiquées [126].

5.2.2. Estimation par Maximum de Vraisemblance *(Maximum Likelihood Estimate (MLE))*

L'Estimation par Maximum de Vraisemblance a été le principe utilisé initialement pour le décodage et l'apprentissage [6]. L'Estimation par Maximum de Vraisemblance est considérée comme garantissant l'optimalité de l'apprentissage si les modèles sont corrects, ce qui n'est sans doute pas le cas en reconnaissance vocale [9]. Cette mesure garantira effectivement l'optimalité en ce qui concerne l'apprentissage, mais pas nécessairement la reconnaissance. Pour améliorer le pouvoir discriminant des modèles, quelques alternatives ont été essayées.

5.2.3. Apprentissage Correctif *(Corrective training)*

Le modèle est tout d'abord appris sur une partie du corpus d'apprentissage en utilisant le MLE. Il est alors utilisé pour reconnaître le corpus d'apprentissage. Lorsqu'il y a une erreur de reconnaissance, ou même si un candidat-mot obtient une note de reconnaissance trop proche de celle du mot correct, le modèle initial est modifié de manière à diminuer la probabilité des étiquettes responsables de l'erreur, ou du *« near-miss »*. Le processus est renouvelé avec les paramètres modifiés. Il est arrêté lorsqu'il n'y a plus de modifications observées. Une liste des mots faciles à confondre acoustiquement peut être utilisée de manière à réduire la durée de ce processus. Cette approche tend à minimiser le taux d'erreurs commises sur le corpus d'apprentissage. Si le corpus de test dans des conditions opérationnelles est semblable au corpus d'apprentissage, le taux d'erreurs sur le corpus de test sera également minimisé [9].

5.2.4. Information Mutuelle Maximale *(Maximum Mutual Information (MMI))*

L'approche par Information Mutuelle Maximale est semblable, mais plus formalisée [7, 109]. Elle se fonde sur le fait que, dans la formule de Bayes, on peut développer le dénominateur en supposant que la probabilité du signal

acoustique dépend de la probabilité de la suite de mots, soit :

$$P(\hat{W} \mid A) = \max_i \frac{P(A \mid W_i) \cdot P(W_i)}{P(A \mid W_i) P(W_i) + \sum_{k \neq i} P(A \mid W_k) P(W_k)}$$

Le but est de déterminer les paramètres du modèle en maximisant la probabilité de générer le signal acoustique, étant donné la séquence de mots correcte W_i, comme pour le MLE, mais, simultanément, en minimisant la probabilité de générer toute séquence de mots erronée W_k, et particulièrement les plus fréquentes. Des résultats comparatifs entre ces deux dernières méthodes ont montré que l'Apprentissage Correctif était meilleur. Cela peut être dû au fait que les séquences de mots qui ont une faible probabilité auront très peu d'effet sur l'apprentissage par MMI, alors qu'elles peuvent avoir un effet en apprentissage correctif [9]. Comparé à l'apprentissage MLE, l'apprentissage MMI est surtout plus robuste lorsque le modèle est incorrect [7], et donne en général de meilleurs résultats [109]. Une méthode différente, l'Information Discriminante Minimale (*Minimum Discriminant Information* (MDI) a été proposée comme une généralisation à la fois du MLE et du MMI [43].

5.2.5. Lissage

Pour obtenir de bons résultats, un modèle de Markov a besoin de beaucoup de données pour l'apprentissage. Si une étiquette n'est jamais trouvée pour un certain arc durant l'apprentissage, on lui donnera une probabilité nulle dans la distribution correspondant à cet arc, et si l'étiquette apparaît durant la reconnaissance, cette probabilité nulle risque d'être attribuée au mot tout entier. Une méthode de lissage simple est de donner une probabilité faible, mais non nulle, à toutes les probabilités qui sont nulles (*floor smoothing* [90]). Une méthode plus sophistiquée consiste à attribuer plusieurs étiquettes au lieu d'une seule à chaque spectre durant l'apprentissage, avec des probabilités calculées à partir des mesures de distance, et définissant ainsi des *prototypes semblables*. Si la probabilité d'émission d'un prototype est nulle sur un arc, elle peut être lissée avec la probabilité non nulle d'un prototype similaire à ce prototype [155]. Une troisième méthode est le lissage par *co-occurrence* [86], qui lisse sur tous les arcs les probabilités des étiquettes qui apparaissent parfois sur les mêmes arcs.

5.2.6. Interpolation par suppression *(Deleted Interpolation)*

Dans le but de lisser les estimations des paramètres obtenues par deux méthodes différentes, il est nécessaire d'appliquer des poids à ces différentes estimations. Ces poids reflèterons la qualité respective des deux estimations, ou même la quantité d'information utilisée pour obtenir chacune d'entre elles. Une méthode pour déterminer automatiquement ces poids est l'*interpolation par suppression,* qui divise l'arc initial en deux arcs différents sur lesquels sont placées les deux estimations, et qui détermine les poids comme les probabilités de transition calculées pour les deux arcs, en utilisant l'algorithme *Forward-Backward* [67].

5.3. MODÉLISATION TEMPORELLE

La modélisation du temps dans un modèle de Markov est contenue dans les probabilités affectées aux arcs. Il apparaît que la probabilité de rester dans un état donné va décroître comme la puissance de la probabilité de suivre l'arc qui boucle sur cet état, ce qui peut être considéré comme un modèle plutôt pauvre dans le cas du signal de parole. On peut trouver différentes tentatives pour améliorer cette situation.

Dans les modèles semi-markoviens [45, 150], un ensemble de fonctions de densité de probabilité $P_i(d)$ à chaque état i indique la probabilité de rester dans cet état pendant une certaine durée, d. Cet ensemble de probabilité est appris en même temps que les probabilités de transition et d'émission en utilisant un algorithme de *Forward-Backward* modifié. Une approche plus simple est d'apprendre indépendamment la probabilité de durée et les paramètres de HMM [139].

Pour faciliter l'apprentissage du modèle, des fonctions de densité de probabilité continues peuvent être utilisées pour la modélisation de la durée, comme la distribution de Poisson [150] ou la distribution Gamma, utilisée par S. Levinson dans son *Continuously Variable Duration Hidden Markov Model* (CVDHMM) [92].

Une autre façon de prendre indirectement le temps en compte est d'inclure la dynamique du spectre comme un nouveau paramètre. Elle peut être représentée comme la différence des coefficients spectraux de deux spectres successifs, et peut également inclure la différence d'énergie. Après quantification vectorielle, les répertoires relatifs à ces différents paramètres nouveaux sont construits. Ils sont introduits dans le HMM avec des distributions de probabilités d'émission sur les arcs indépendantes [87].

5.4. UNITÉS DE DÉCISION

5.4.1. Le mot

L'idée première est de modéliser un mot avec un HMM. Un exemple de machine markovienne correspondant à un mot, dû à R. Bakis [66], est donné en figure 5. Le nombre d'états dans le modèle du mot est égal à la durée moyenne du mot (50 états pour un mot de 500 ms, si l'on a un spectre toutes les 10 ms). Il faut noter que le modèle inclut les phénomènes d'insertion et d'élision de spectres qui étaient auparavant traités lors de la mise en correspondance par programmation dynamique. Plus récemment, des modèles avec moins d'états ont été essayés avec succès [138]. Le problème est que pour obtenir un bon modèle du mot, il est nécessaire d'avoir un grand nombre de prononciations de ce mot. Le processus de décision considère de plus le mot comme une entité globale à reconnaître, et ne focalise pas son attention sur l'information qui permet de différencier deux mots acoustiquement voisins.
Comme pour l'approche par programmation dynamique, l'utilisation d'unités plus petites que le mot a un certain

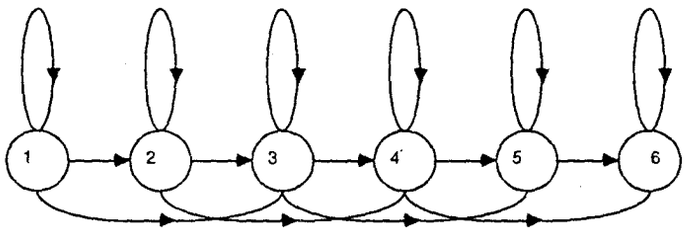

Figure 5. — Exemple de modèle de Bakis pour un mot.
La longueur moyenne du mot est de 50 ms.

nombre d'avantages. Le processus de segmentation-par-reconnaissance réalisé par l'algorithme de Viterbi permet d'éliminer le problème de segmentation a priori, et autorise donc des unités aussi petites que les phonèmes (appelés également phones afin de s'affranchir de la définition linguistique stricte du phonème), les diphonèmes ou diphones, les syllabes, les demi-syllabes, etc.

5.4.2. Diphones

L'utilisation de modèles de diphones a été comparé avec des modèles de phones, ou des modèles composites, composés de phones et de transitions entre phones. Les modèles de transition étaient construits uniquement pour les transitions correspondant à des phénomènes spécifiques (occlusive-voyelle, fricative-voyelle, etc.). Le modèle composite donne les meilleurs résultats, avec un nombre de modèles inférieur à celui des diphones [35].

5.4.3. Phonèmes indépendants du contexte

Les modèles de phones indépendants du contexte sont intéressants car ils sont en nombre limité. Un exemple d'un tel modèle de phone est donné en figure 6. Ils furent utilisés dans les premiers travaux du groupe « parole » d'IBM à T. J. Watson Research Center pour la reconnaissance de mots isolés et de parole continue [6].

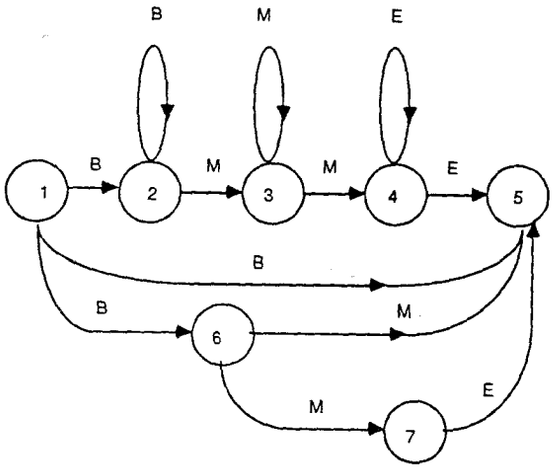

Figure 6. — Exemple d'un modèle de phone dans le système SPHINX.
Les distributions de probabilités d'émission B (début), M (Milieu) et E (Fin) sont « liées » (forcées à être identiques) sur différents arcs. La longueur minimale du mot est d'un spectre. Il n'y a pas de longueur maximale.

Si les unités de décision sont des unités plus courtes que le mot, comme les phones, chaque mot est représenté comme une séquence de ces unités (ou un réseau si l'on tient compte des variations phonologiques dans la prononciation du mot (*fig.* 7)). Si l'on n'utilise pas d'information

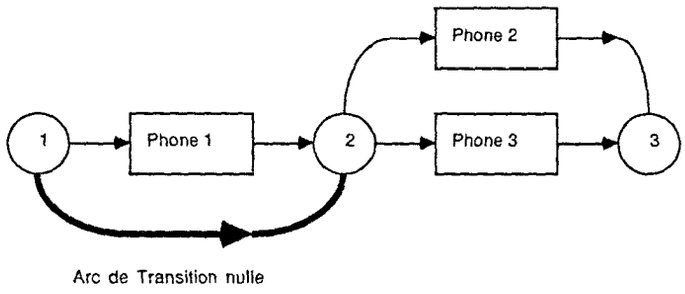

Figure 7. — Exemple d'un modèle de mot construit à partir de modèles de phones.
Le mot peut avoir une longueur d'un ou deux phonèmes. Le premier phonème peut être omis. Il y a deux possibilités pour le second phonème. La probabilité des différentes variantes phonologiques peut être placée sur les arcs. L'arc de « transition nulle » est suivi sans émission de symbole.

lexicale, les unités de décodage sont placées dans un modèle phonémique « en boucle » (*Looped Phonetic Model* (LPM)), où l'on peut éventuellement affecter différentes probabilités aux successions de phonèmes (*fig.* 8) [108].

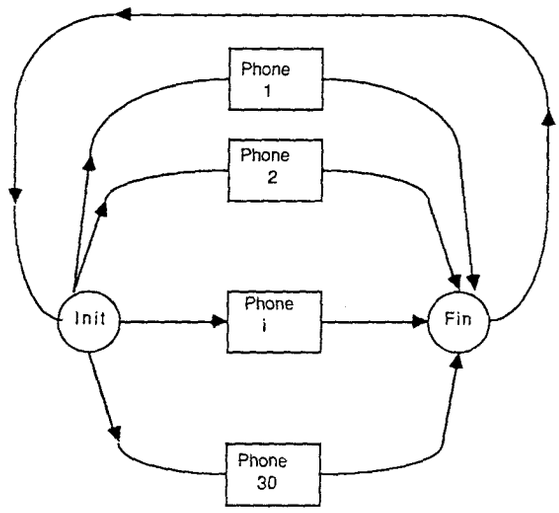

Figure 8. — Un modèle phonémique en boucle.
Chaque rectangle est une machine phonémique. Les arcs reliant l'état initial à chaque phone, chaque phone à l'état final, et l'état final à l'état initial sont des arcs de transition nulle. La probabilité de succession des phonèmes peut être utilisée comme un « modèle de langage » phonologique.

Malheureusement, les simples modèles de phones sont très affectés par le contexte, et les paramètres de ces modèles de phones reflèteront des signaux acoustiques très différents pour le même phonème.

5.4.4. Phones dépendants du contexte

Pour traiter ce problème, les phones dépendants du contexte ont été proposés [5, 155]. Des modèles de phone différents sont construits pour chaque contexte du phone. Si l'on utilise 30 phones, il y aura donc environ 1 000 modèles pour chaque phone si l'on considère le contexte gauche, ou le contexte droit, et environ 30 000 modèles si l'on considère à la fois les contextes gauche et droit (appelés modèles de *triphones*). Ici aussi, il peut être difficile d'obtenir assez de données d'apprentissage pour faire l'apprentissage de tous ces modèles. Si l'on considère les triphones que l'on trouve à l'intérieur des mots du vocabulaire, la prononciation de chacun de ceux-ci permet d'avoir tous les triphones. Par contre, si l'on considère également les triphones à la jonction des mots, leur nombre devient beaucoup plus grand, et on n'est pas sûr d'avoir la totalité des successions possibles dans le texte d'apprentissage (triphones de frontière de mots *(between-word triphones)* [64]). On peut utiliser des connaissances en phonétique pour réduire le nombre de triphones à apprendre, dans la mesure où certains contextes auront le même effet sur le phone central [38]. Une alternative, dans l'approche par *triphone généralisé* [87], est d'effectuer une comparaison de mesures d'entropie du HMM, (suivant que l'on conserve deux modèles de phones différents pour deux contextes différents, ou qu'on les intègre, dans un seul modèle), et de déterminer ainsi les modèles de triphones contextuels à conserver, en choisissant la solution qui donne l'entropie minimale.

5.4.5. Phones dépendants des mots

De la même façon, un modèle de phone peut être appris dans le contexte d'un mot donné. Si le vocabulaire est de faible taille, et si le nombre de prononciations de chaque mot est grand, l'apprentissage de tels modèles est alors possible. Cette approche a été utilisée par les chercheurs du CNET dans leur système de reconnaissance multilocuteur en mots isolés par téléphone [40], et par BBN pour un vocabulaire de 1 000 mots [30]. A CMU, K. F. Lee a utilisé des modèles de phones dans le contexte de mots-outils [87]. Les mots-outils sont les mots grammaticaux, habituellement courts et mal prononcés, et donc difficiles à reconnaître. Ils sont très fréquents dans la parole courante, et vont donc affecter de façon importante la qualité des performances de reconnaissance. Mais, puisqu'ils sont fréquents, leur apprentissage est possible. Tout cela justifie le besoin, et la possibilité d'avoir des modèles spéciaux pour les phones dans le contexte de ces mots-outils.

Il est possible de mélanger les différents modèles (indépendant du contexte, dépendant du contexte, dépendant du mot) d'un même phone en utilisant la méthode d'*interpolation par suppression* [87, 38].

5.4.6. Fénones

D'autres modèles sont de nature acoustique. L. Bahl et al. utilisent le concept de *fénones* [10]. L'idée est de représenter la prononciation d'un mot comme une suite d'étiquettes correspondant à des spectres prototypes obtenus par quantification vectorielle, et de créer une machine markovienne très simple, appelée une machine de fénone (*fig.* 9), pour chacune des étiquettes. Les paramètres de ces modèles peuvent être appris sur plusieurs prononciations d'un même mot. Cette approche est proche de la programmation dynamique sur des formes correspondant à des mots. Les phénomènes d'omission et d'insertion de spectres, qui sont traités par la PD sont ici inclus dans le modèle. Par exemple, les étiquettes correspondant à un instant de stabilité auront une probabilité importante d'emprunter l'arc en boucle. Mais à la différence de la comparaison de formes, les auteurs soulignent que les

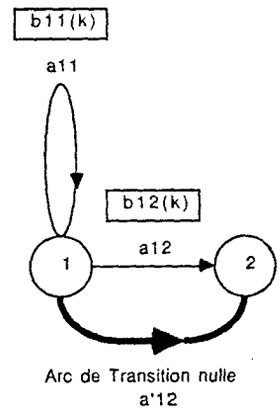

Figure 9. — Une machine fénonique.
L'arc en gras est une transition nulle. La longueur de la machine peut être de 0, 1 ou plusieurs spectres.

modèles de fénones peuvent être appris pour un nouveau locuteur. L'utilisation de modèles de fénones multilocuteurs est donc une façon de représenter le modèle de durée de chaque mot.

5.4.7. Segments

Ces segments sont similaires à ceux qui sont utilisés dans la *quantification segmentale* avec des méthodes de comparaison de formes par programmation dynamique [147]. Ils sont obtenus en appliquant un algorithme de segmentation par l'approche du Maximum de Vraisemblance. Un processus de quantification segmentale est alors appliqué. Chacun des segments prototypes du répertoire résultant est représenté par un HMM, dont on apprend les paramètres sur les données initiales. Chaque mot du lexique est représenté par un réseau de ces unités acoustiques. Les résultats en mots isolés sont semblables à ceux qui sont obtenus avec des modèles de mots [88].

5.4.8. Choix des unités

Plusieurs aspects doivent être pris en compte dans le choix d'une unité :

a) Comme pour les modèles de mots, il est nécessaire d'avoir un grand nombre d'unités présentes dans le corpus d'apprentissage. Plus l'unité est petite, plus elle sera

présente dans le corpus d'apprentissage, et meilleurs seront les paramètres du modèle.

b) Mais également plus ils seront sensibles au contexte. Pour résoudre ce problème, nous avons vu qu'il était possible d'associer les unités à des contextes spécifiques.

c) Un autre aspect important est la possibilité d'améliorer un modèle précis construit avec des données en nombre insuffisant, en utilisant les paramètres obtenus avec un modèle d'unités plus générales, comme un modèle de phones contextuels, qui peut être amélioré en lissant ses paramètres avec ceux d'un modèle indépendant du contexte pour le même phone.

Ces trois aspects sont appelés **apprenabilité, sensitivité** et **partageabilité** *(trainability, sensitivity and sharability)* [87].

5.5. ADAPTATION AU LOCUTEUR

L'adaptation à un nouveau locuteur peut être obtenue en utilisant des techniques d'adaptation basées sur la mise en correspondance de répertoires. L'approche poursuivie à BBN consiste initialement à quantifier vectoriellement une phrase inconnue avec le répertoire du locuteur de référence [156], et à appliquer un algorithme Forward-Backward modifié pour calculer la matrice de transformation représentant la probabilité conditionnelle d'un spectre quantifié du nouveau locuteur, étant donné un spectre quantifié du locuteur de référence. Cela a été amélioré en construisant le répertoire du nouveau locuteur, en effectuant un alignement par PD d'une phrase inconnue prononcé à la fois par le locuteur de référence et le nouveau locuteur, et en comptant les co-occurrences des spectres-prototypes du locuteur de référence et du nouveau locuteur [44]. Dans le contexte de la reconnaissance monolocuteur, l'adaptation d'un locuteur de référence à un nouveau locuteur, même sur une faible durée de parole (15 secondes de parole), donne des résultats proches de ceux obtenus avec un apprentissage monolocuteur sur 15 minutes de parole. Dans le contexte de la reconnaissance multilocuteur, les expériences conduites à CMU qui combinent des paramètres dépendants du locuteur et des paramètres indépendants du locuteur obtenus sur 30 phrases avec une méthode d'*interpolation par suppression*, ont donné 4 % d'amélioration en utilisant la méthode d'adaptation (dans le cas où on n'utilise pas de syntaxe) (tableau 1).

5.6. MODÈLE LINGUISTIQUE

Le *modèle de langage* peut être aussi représenté comme un processus Markovien. Dans un modèle de bigramme, la probabilité qu'un mot suive un mot donné est calculée comme la probabilité d'une séquence de deux mots [6]. Dans un modèle de Trigramme, la probabilité qu'un mot suive une suite de deux mots est calculée de la même façon. Un modèle d'unigramme est simplement la probabilité d'un mot. Un modèle plus simple est le modèle de paire-de-mots *(word pair)* où l'on donne la même probabilité à tous les mots qui peuvent suivre un mot donné.

Ces différents modèles doivent être appris sur un corpus important. Si le corpus n'est pas assez grand, et si le nombre de mots dans le vocabulaire est important, beaucoup de suites de mots qui sont en fait valides seront absentes, et le modèle, surtout dans le cas des trigrammes, contiendra beaucoup de probabilités nulles (si un vocabulaire de 5 000 mots est utilisé, la taille de la matrice des trigrammes est de $5{,}000^3$!). Cela peut être amélioré en utilisant des techniques de lissage, de type « lissage-plancher » *(floor smoothing)*, comme l'estimation de Turing-Good [115], qui énonce qu'une estimation de la probabilité des mots jamais rencontrés dans un corpus d'apprentissage est le rapport entre le nombre de mots qui ont été rencontrés une seule fois, divisé par le nombre total de mots dans le corpus. Une autre possibilité est d'utiliser l'interpolation par suppression pour combiner les probabilités d'unigrammes, de bigrammes et de trigrammes dans le modèle de langage complet [37].

TABLEAU 1

Systèmes de reconnaissance de parole continue pour de grands vocabulaires

(D'après K. F. Lee [87, 127], PSI : Philips-Siemens-IPO, En ce qui concerne les résultats de TI, seules les voix masculines ont été testées))

Lab	Système	Loc.	Voc.	Pxté	% Corr.	Acuité Mots	Date
CMU	Hearsay	dépt	1 011	4,5	86 %		1977
CMU	Harpy	dépt	1 011	4,5	97 %		1977
IBM	Laser	dépt	1 000	24	91,1 %	88,6 %	1980
BBN	BYBLOS	dépt	997	60	94,8 %	92,5 %	1987
BBN	BYBLOS	dépt	997	997	70,1 %	67,6 %	1987
PSI	SPICOS	dépt	917	74	92,8 %	92,0 %	1988
CSELT	CSELT	dépt	1 011	35	79,0 %		1988
CMU	ANGEL	indépt	997	997	45,5 %	41,0 %	1986
TI	TI	indépt	997	997	55,5 %	44,3 %	1987
SRI	SRI	indépt	997	997	43,6 %	40,4 %	1988
CMU	SPHINX	dépt	997	20	96,9 %	96,1 %	1988
CMU	SPHINX	dépt	997	60	94,9 %	94,0 %	1988
CMU	SPHINX	dépt	997	997	76,5 %	74,1 %	1988
CMU	SPHINX	indépt	997	20	96,2 %	95,8 %	1988
CMU	SPHINX	indépt	997	60	94,7 %	93,7 %	1988
CMU	SPHINX	indépt	997	997	74,2 %	70,6 %	1988
Lincoln	Lincoln	dépt	991	60		94,5 %	1988
Lincoln	Lincoln	dépt	991	991		80,7 %	1988
Lincoln	Lincoln	indépt	991	60		89,9 %	1988
Lincoln	Lincoln	indépt	991	991		66,1 %	1988

Le pourcentage des faits linguistiques réels contenu dans le modèle de langage est appelé le recouvrement *(coverage)*. De manière intéressante, les expériences conduites sur la reconnaissance de vocabulaires de grande taille ont donné, sur un corpus de test de 722 mots un taux d'erreur de 17,3 % avec un dictionnaire de 10 000 mots, incluant 43 erreurs correspondant à des mots qui ne sont pas dans le lexique (soit un recouvrement de 94 %). Avec un dictionnaire de 200 000 mots, le taux d'erreurs descend à 12,7 %, dans la mesure où, même si la tâche est plus difficile, le recouvrement est alors de 100 % [108] !

Dans un modèle de Biclasse ou de Triclasse, la probabilité des successions de mots est remplacée par la probabilité de succession de classes grammaticales [3]. La probabilité d'un mot donné à l'intérieur d'une classe peut également être utilisée pour raffiner le modèle (modèle TriPOS (ou *Part Of Speech*) [37]). Une approche intermédiaire est d'utiliser le modèle de trigrammes lissés, où les probabilités des mots longs (de trois syllabes ou plus) sont liées, dans la mesure où ils sont faciles à reconnaître, et n'ont habituellement pas d'homophones (du moins en anglais...) [39].

L'avantage des modèles de langage basés directement sur les mots est qu'ils contiennent à la fois de l'information syntaxique et de l'information sémantique. Ils seront également plus simples à apprendre, dans la mesure où il n'est pas nécessaire d'étiqueter le corpus de textes d'apprentissage. Cependant la quantité de données nécessaire à l'apprentissage du modèle, surtout dans le cas d'un modèle de trigrammes, sera très importante. Lorsqu'on utilise des catégories grammaticales, il faudra étiqueter le corpus de textes, mais on pourra utiliser un corpus plus réduit. De plus, si un nouveau mot est introduit dans le lexique, il peut hériter des probabilités calculées pour les mots qui ont la même catégorie syntaxique.

Pour la tâche de dictée vocale, la référence est le texte écrit dicté à voix haute. Dans cette mesure, on peut utiliser un corpus de textes écrits pour apprendre le modèle. IBM a utilisé un texte de 250 millions de mots pour apprendre leur modèle dans le système Tangora (comme cela est indiqué dans [87]). Dans le programme Européen ESPRIT, des modèles de langage multilingues ont été construits [24]. Pour la compréhension du langage parlé, il est nécessaire d'utiliser un corpus de parole pour apprendre le langage, mélangeant ainsi la modélisation acoustique et la modélisation linguistique. Lorsqu'il existe des transcriptions par écrit des dialogues, le modèle peut être appris sur ces transcriptions. Dans la mesure où seul un corpus réduit était disponible (1 200 phrases de la base de données « Resource Management Database » du DARPA), BBN a proposé récemment d'utiliser un modèle qui utilise à la fois la probabilité des successions de syntagmes, et la probabilité des mots à l'intérieur des syntagmes [143].

L'utilisation d'un modèle de langage est une nécessité absolue. Des expériences menées en Français sur la conversion phonème-graphème pour des suites de phonèmems exemptes d'erreur ont montré qu'une simple phrase de 9 phonèmes engendrait plus de 32 000 segmentations en mots et orthographications de ces mots ainsi segmentés [3].

5.7. SYSTÈMES DE RECONNAISSANCE VOCALE BASÉS SUR LES MODÈLES MARKOVIENS

Les HMMs ont été utilisés dans beaucoup de systèmes, pour tenter de résoudre différents types de problèmes :

5.7.1. Reconnaissance Monolocuteur par mots isolés pour des petits vocabulaires

Dans cette tâche relativement simple, les HMMs ont été utilisés pour rendre le système plus robuste à des variations de la prononciation d'un locuteur donné. A Lincoln Lab, les modèles HMM continus de mots sont appris sur différents modes d'élocution (débit normal, rapide, voix forte, voix douce, voix criée, et effet Lombard). Cela est appelé « Apprentissage multi-style ». Sur la base de données de Texas Instruments utilisant un vocabulaire de 105 mots, les résultats ont été de 0,7 % d'erreurs [125]. Sur la difficile tâche du clavier, qui inclut l'alphabet, les chiffres et les signes de ponctuation, IBM a obtenu un taux d'erreur de 0,8 %, en utilisant des modèles de fénones [10].

5.7.2. Reconnaissance Multilocuteur par mots isolés pour des petits vocabulaires

Le CNET en France a utilisé cette approche pour la reconnaissance par téléphone d'un petit nombre de mots, prononcés en mode isolé. Le système est robuste. Le modèle est appris à partir de nombreuses prononciations de chaque mot du vocabulaire, par une population importante à travers le réseau téléphonique. Le système utilise des modèles HMM continus des mots. Les résultats sont de 85 % pour les chiffres isolés, et 89 % pour les signes du zodiaque, à travers le réseau téléphonique public analogique [40].

5.7.3. Reconnaissance Monolocuteur de parole continue pour des petits vocabulaires

Les travaux de Lincoln Lab sur la reconnaissance robuste de mots isolés ont été étendus à la parole continue. Des tests préliminaires ont été effectués avec un vocabulaire de 207 mots, le langage ayant une perplexité de 14 mots. Dix types de prononciation étaient présents. Le taux d'erreur sur les mots a été de 16,7 % [126].

TABLEAU 2

Taux de reconnaissance sur les phrases comparés à l'acuité sur les mots [127]

Lab	Système	Loc.	Voc.	Pxté	Acuité Mots	Phr. corr.	Date
Lincoln	Lincoln	dépt	991	60	94,5 %	68,0 %	1988
Lincoln	Lincoln	indépt	991	60	89,9 %	57,3 %	1988

5.7.4. Reconnaissance Multilocuteur de parole continue pour des petits vocabulaires

A AT&T Bell Labs, de très bons résultats ont été obtenus sur la reconnaissance monolocuteur, plurilocuteur et omnilocuteur de chiffres prononcés en continu, avec des modèles HMM continus de mots (0,78 %, 2,85 %, and 2,94 % d'erreur sur les suites, pour des suites de chiffres de longueur inconnue) [141]. Au CNET, un système a été conçu pour la composition vocale de numéros de téléphone, avec des nombres de deux chiffres. Le résultat est une cabine téléphonique sans cadran qui est actuellement testée dans divers endroits publics [69].

5.7.5. Reconnaissance Monolocuteur de mots isolés pour des grands vocabulaires

Le groupe « Parole » d'IBM-TJ Watson Research Center a annoncé un système de reconnaissance de 5 000 mots, monolocuteur et par mots isolés, autonome sur PC, en 1986 [159]. En 1987, ils ont présenté une nouvelle version avec un vocabulaire de 20 000 mots. Ils utilisent à la fois des modèles de phones et de fénones [68].

Au Centre Scientifique IBM de Paris, des expériences ont été menées sur un très grand vocabulaire (200 000 mots). Le mode de prononciation est syllabe par syllabe. Bien que ce mode de prononciation soit difficilement acceptable, l'intérêt est que le modèle de langage se doit de correspondre à la parole continue, qui inclut les problèmes de liaisons et d'élisions en Français. Ils utilisent des modèles de phones [108].

A INRS/Bell Northern (Canada), des tests avec un vocabulaire de 75 000 mots et 3 modèles de langage différents ont été menés. Les meilleurs résultats (autour de 90 %) ont été obtenus avec un modèle de trigramme qui donne la perplexité la plus faible [39].

TABLEAU 3

Résultats de quelques systèmes de reconnaissance de mots isolés pour des grands vocabulaires (avec modèle de langage)
(* pour des phrases qui obéissent à la grammaire) [39, 159, 68, 146]

Lab	Système	Loc.	Voc.	Pxté	% mots Corrects	Date
IBM	Tangora	dépt		5 000 160	97,1 %	1986
IBM	Tangora	dépt		20 000 250	94,6 %	1987
DRAGON	Dragon	dépt		997 21*	98,6 %	1988
	writer	indépt		997 21*	91,9 %	1988
INRS	INRS	dépt		75 000 67	89,5 %	1988

5.7.6. Reconnaissance monolocuteur de parole continue pour des grands vocabulaires

Le groupe « Parole » d'IBM TJ Watson Research Center a présenté un système de reconnaissance de 5 000 mots, monolocuteur, en parole continue, en 1989 [11]. BBN a présenté le système BYBLOS en 1987 [31]. Ce système utilise à présent des modèles de phones indépendants du contexte, dépendants du contexte et dépendants du mot, et reconnaît un vocabulaire de 1 000 mots en temps réel.

5.7.7. Reconnaissance multilocuteur de parole continue pour des grands vocabulaires

Le système SPHINX a été développé à CMU. Il a été testé sur la base de données DARPA « Resource Management », avec un vocabulaire de 967 mots. Il utilise des modèles HMM discrets de triphones généralisés, et des modèles de phones dans les mots-outils. La syntaxe est donnée par les paires de mots, ou par un modèle de bigramme [87]. La même tâche a été réalisée à Lincoln Labs avec des résultats légèrement moins bons, en utilisant des modèles HMM continus, avec un mélange de 4 gaussiennes [127]. A SRI, un système similaire a été réalisé avec des modèles HMM discrets plus simples de phones [111].

6. L'approche connexionniste

Dans l'approche connexionniste, les données de référence sont représentées comme des patterns d'activité distribués sur un réseau d'unités de traitement simples [96, 98].

6.1. LES PERCEPTRONS

6.1.1. Principe

L'ancêtre dans cette approche est le Perceptron, un modèle de perception visuelle proposé par F. Rosenblatt [145], qui fut finalement abandonné après qu'on eût prouvé qu'il était inopérant dans certains cas [110]. Plus récemment, il y a eu un regain d'intérêt pour ce modèle. Cela est dû au fait qu'il a été montré que les Perceptrons Multi-couches (MLP) ont des capacités de classification supérieures à celles du Perceptron initial [96], et qu'un algorithme d'apprentissage, appelé *Rétropropagation (Back-Propagation)* a été proposé récemment pour les MLP [174, 148, 83, 124]. Un Perceptron Multi-couches est composé d'une couche d'entrée, d'une couche de sortie, et d'une ou plusieurs couches cachées intermédiaires. Chaque couche est composée de plusieurs cellules. Chaque cellule i dans une couche donnée est reliée à chaque cellule j de la couche suivante par des liens, ou arcs, auxquels est attaché un poids W_{ij} qui peut être positif ou négatif, suivant que la cellule de départ i excite, ou au contraire inhibe la cellule d'arrivée j. L'analogie avec le fonctionnement du cerveau humain a conduit à appeler les cellules « neurones », les liens « synapses », et le modèle « modèle neuronal », ou « neuromimétique ». Le stimulus est introduit dans les cellules d'entrée (qui ne peuvent prendre que les valeurs 0 ou 1 si le modèle est binaire), et est propagé dans le réseau. Dans chaque cellule d'arrivée j, la somme des énergies pondérées y parvenant est calculée. Si cette somme est supérieure à un seuil T_j, la cellule réagit, et, à son tour, transmet de l'*énergie* vers les cellules des couches suivantes (la réponse de la cellule à l'énergie entrante est donnée par une fonction *sigmoïde* S[ ]) (*fig.* 10).

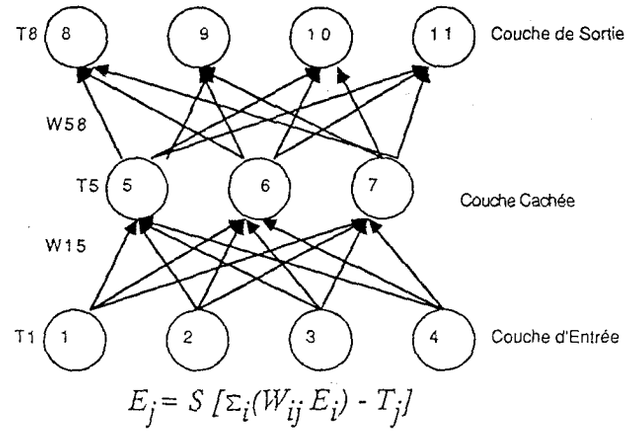

Figure 10. — Un Perceptron à deux couches.
Chaque lien a un poids W_{ij}. Chaque cellule a un seuil d'activité T_j. E_i est l'énergie émise par la cellule i.

Dans la phrase d'apprentissage, le stimulus propagé, lorsqu'il atteint les cellules de sortie, est comparé avec la réponse désirée en sortie, en calculant une valeur d'erreur qui est *rétropropagée* vers les couches inférieures, de façon à ajuster les poids affectés aux liens, et le seuil d'activité dans chaque cellule. Ce processus est itéré jusqu'à ce que les paramètres du modèle atteignent une stabilité suffisante. Cela est effectué pour toutes les paires stimulus-réponse.

Dans la phase de reconnaissance, le stimulus est propagé jusqu'à la couche de sortie. Dans certains systèmes, la cellule de sortie qui a la plus grande valeur désigne la forme reconnue. Dans d'autres, le vecteur correspondant à l'ensemble des cellules de sortie est comparé avec chaque forme de référence, en utilisant une distance (comme la distance de Hamming, si les cellules ont des valeurs binaires). Le rôle des cellules cachées est d'organiser l'information de telle sorte que l'information discriminante soit activée dans le réseau pour distinguer deux éléments très proches. L'espoir est que la cellule cachée correspondant au trait discriminant agira sur la bonne cellule de sortie avec un poids important et positif, mais agira sur la cellule de sortie erronée avec un poids fortement négatif.

Une étude attentive du comportement des cellules de la couche cachée durant la reconnaissance a montré que certaines d'entre elles réagissaient de fait à certains traits discriminants, comme la distinction de plosives alvéolaires vs vélaires [42], ou un deuxième formant chutant vers 1 600 Hz vs un formant stable vers 1 800 Hz [172], dans la reconnaissance de « B », « D », « G » pour des contextes variés. De tels aspects d'autoorganisation, assurément intéressants, ont été également rencontrés dans les HMMs, en utilisant un modèle ergodique à 5 états, où chaque état était relié à tous les autres, sans étiquetage préalable. Après apprentissage, il est apparu que les états correspondaient à des traits acoustico-phonétiques bien connus (Voisement marqué, silence, Nasale ou Liquide, Explosion d'une occlusive, Frication) [129]. Cette qualité n'est donc pas l'apanage unique des modèles connexionnistes.

6.1.2. Traitement du temps

Si le pouvoir discriminant de tels réseaux est intéressant pour la reconnaissance vocale, le paramètre « temps » est par contre difficile à modéliser. Plusieurs façons de le prendre en compte peuvent être mentionnées :

6.1.2.1. *Compression temporelle à longueur fixe*

Une approche est d'utiliser simplement comme longueur de référence la longueur maximale possible des mots, et de rajouter des spectres de silence aux mots qui ont une longueur moindre [128]. Une autre possibilité est de normaliser le spectrogramme correspondant à un mot à une longueur fixe (cela peut être obtenu par compression linéaire, ou non linéaire, à longueur fixe). Si le spectrogramme est de longueur I, chaque vecteur étant de dimension D, le réseau aura donc D.I cellules d'entrée. Si la taille du vocabulaire est de M mots, le réseau aura M cellules de sortie. Un mot sera reconnu comme étant celui pour lequel la cellule de sortie correspondante a la plus grande valeur d'activation.

6.1.2.2. *MLP contextuel*

Dans le but de prendre en compte l'information contextuelle, l'entrée peut inclure le contexte dans lequel le stimulus se produit. T. Sejnowski a utilisé cette méthode pour la conversion graphème-phonème en anglais [158]. Considérons qu'il y ait 26 graphèmes, 3 signes de ponctuation et 30 phonèmes en Anglais. L'entrée est composée de 7 groupes de 29 cellules. Chaque groupe représente un graphème, ou un signe de ponctuation, la cellule correspondante ayant la valeur 1, les 28 autres ayant la valeur 0. Le groupe central représente le graphème à convertir, les 3 groupes à gauche, et les 3 groupes à droite représentant respectivement les contextes gauche et droite (les 3 graphèmes qui suivent celui qui est à convertir, et les 3 qui le précèdent). Le phonème correspondant est donné dans les cellules de sortie (*fig.* 11). C'est-à-dire que parmi les 30 cellules de sortie, celle qui correspond au bon phonème est mise à 1, alors que les autres sont à 0 (en réalité, un codage différent a été utilisé pour les cellules de sortie, basé sur 17 traits phonologiques, 4 traits de ponctuation, et 5 traits d'accent et de frontières de syllabes (comme voyelle, voisé...), résultant en un total de 26 cellules de sortie). Le réseau peut être appris sur des paires graphème-phonème à l'intérieur d'un mot, obtenues à partir d'un lexique, ou de textes, et il peut alors apprendre des règles de conversion qu'il peut appliquer à des mots nouveaux. La qualité de conversion de mots nouveaux s'améliorera avec la taille du corpus d'apprentissage. Avec

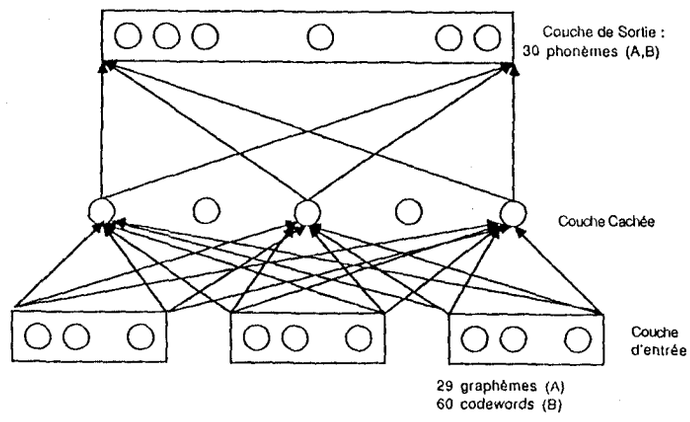

Figure 11. — Exemple de Perceptron Multi-couches contextuel.
Chaque rectangle correspond à plusieurs cellules, représentant différentes unités (dans le cas de la conversion graphème-phonème (A), ou de la reconnaissance phonétique (B)). Ici, le contexte est d'une entité à droite et à gauche du stimulus.

cette approche, les auteurs ont obtenu un taux de conversion correcte de 78 % sur les mots d'un texte de test, ce qui est nettement inférieur aux résultats obtenus en utilisant classiquement des règles de conversion introduites manuellement.

Cette approche a été étendue à la reconnaissance de la parole [22]. Dans ce cas, l'entrée est constituée de

11 groupes de 60 cellules correspondant aux étiquettes obtenues par quantification vectorielle sur un répertoire de 60 prototypes. La sortie est constituée de 26 cellules correspondant aux 26 phonèmes utilisés pour composer les chiffres en allemand. L'apprentissage est effectué sur un corpus de parole étiqueté phonétiquement, où l'on donne simultanément l'étiquette du spectre prototype, et les étiquettes en contexte comme entrée, et le phonème correspondant en sortie. Le corpus d'apprentissage comprend 2 prononciations de chaque chiffre en isolé. La reconnaissance des chiffres eux-mêmes, prononcés en mode isolé ou continûment, est effectuée par Programmation Dynamique après reconnaissance phonétique à travers le Perceptron. La comparaison de cette technique avec un modèle HMM simple utilisant des modèles de phones à un seul état, a été en faveur de l'approche MLP (pas d'erreurs en reconnaissance de mots isolés, et 92,5 % de reconnaissance correcte pour des suites de 7 chiffres, contre 80 % et 70 % (HMM discret (avec le même répertoire que pour les MLP), et 100 % et 90 % (avec un HMM continu)). Un des résultats remarquables est le pouvoir discriminant de l'approche MLP, l'émergence du bon phonème étant bien plus apparente que dans le cas des HMMs, où le bon phonème a une émergence faible comparé au second candidat, même si la décision finale de reconnaissance du bon mot est correcte dans les deux cas.

6.1.2.3. *Les réseaux neuronaux à délai temporel* (Time Delay Neural Networks (TDNN))

Une autre approche similaire a été proposée par A. Waibel [172]. La tâche est la reconnaissance de 3 phonèmes « B », « D », « G » dans différents contextes. Ici, le Perceptron Multi-couches est composé de 2 couches cachées. La longueur du stimulus est fixe, et égale à 15 spectres (150 ms). La couche d'entrée est constituée de 16 cellules représentant 16 coefficients cepstraux, chaque cellule étant reliée aux cellules de la première couche cachée par 3 arcs représentant la valeur du coefficient cepstral aux temps t, $t - 10$ ms, et $t - 20$ ms. La première couche cachée est composée de 8 cellules. Chaque cellule est reliée aux cellules de la deuxième couche cachée par 5 arcs représentant la valeur de la cellule aux temps t à $t - 40$ ms. La deuxième couche cachée a 3 cellules. Chaque cellule de la couche de sortie reçoit l'énergie intégrée (sommée) sur la durée totale du stimulus pour l'une des cellules de la deuxième couche cachée. La couche de sortie est composée de 3 cellules représentant chacun des 3 phonèmes. La phase d'apprentissage tient compte du fait que les arcs correspondant à un coefficient cepstral à un temps donné t seront observés 3 fois (aux temps t, $t + 10$ ms avec un délai de 10 ms, et $t + 20$ ms, avec un délai de 20 ms) et que les poids sur ces arcs doivent donc être identiques. Cette approche a été comparée à une approche par HMM discret, utilisant 4 états, 6 transitions et un répertoire de 256 prototypes, chacun des 3 phonèmes ayant un modèle différent. Les résultats ont été en faveur de l'approche MLP (1,5 % d'erreur, contre 6,3 %). Ici aussi, l'émergence du phonème correct comparé au second candidat a montré les plus grandes capacités de discrimination de l'approche MLP.

6.1.2.4. *Les réseaux récurrents*

Ces derniers réseaux font intervenir un retour de la couche de sortie, ou de la couche cachée, vers la couche d'entrée ([24], [47]) (*fig.* 12). Cette approche permet donc de

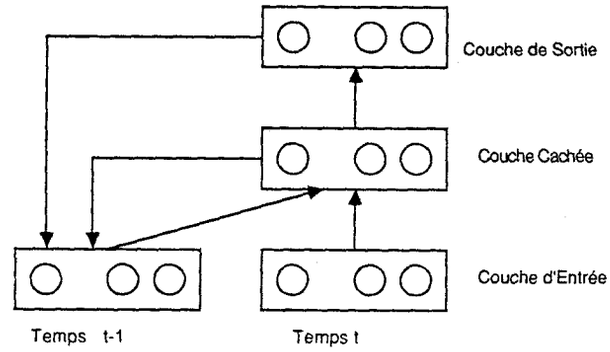

Figure 12. — Perceptron multicouche récurrent.
(à partir de la couche de sortie, ou de la couche cachée)

rendre compte du fait que tel stimulus, dans le contexte de telle reconnaissance précédente, a telle signification (cas du retour de la couche de sortie). Ou que tel stimulus, dans le contexte de tel trait discriminant précédent (sans qu'on l'ait pour autant reconnu), a telle signification (cas du retour de la couche cachée). Ces modèles généralisent donc les perceptrons contextuels.

6.1.2.5. *Les réseaux neuronaux et les algorithmes de Programmation Dynamique ou de Viterbi*

Dans le but de mieux tenir compte des propriétés de bonne discrimination de l'approche par réseaux neuronaux, tout en profitant des propriétés de bon alignement temporel des algorithmes de Programmation Dynamique [152] ou de Viterbi [97, 61], quelques premiers essais de les utiliser conjointement peuvent être recensés. Les travaux utilisant des réseaux neuronaux complexes (contextuels et récurrents), et ayant pour cellules de sortie les états d'une machine markovienne ont été présentés [24], [49]. Les tests effectués ne permettent cependant pas actuellement d'obtenir de meilleurs résultats qu'avec un simple modèle markovien.

6.2. LA MACHINE DE BOLTZMAN/LE RECUIT SIMULÉ

La Machine de Boltzman est également composée de nœuds (ou cellules) et de liens pondérés entre ces nœuds. Contrairement au MLP, les nœuds ne sont pas organisés suivant des couches différentes, chaque nœud pouvant être relié à tout autre nœud (liaison complète *(fully connected)*). Par contre, on trouve une analogie dans le fait que les nœuds sont habituellement divisés en nœuds visibles, et nœuds cachés, et que les nœuds visibles peuvent également être divisés en nœuds d'entrée et nœuds de sortie. Une autre différence est que l'on donne habituellement aux nœuds uniquement des valeurs binaires. On affecte à chaque nœud une probabilité d'être à 0

ou à 1. Cette fonction probabiliste dépend de la « différence d'énergie » (égale à la somme pondérée de l'énergie en provenance des autres nœuds) arrivant dans ce nœud, suivant qu'il est placé à 0 ou à 1. Il contient également un terme semblable à la température en thermodynamique. Plus la température est élevée, plus le nœud sera capable de prendre la valeur 0 ou 1 de façon aléatoire. Plus la température est basse, plus le nœud sera influencé par l'état des nœuds qui lui sont reliés, et par le poids affectés aux arcs correspondants. Au début du processus d'optimisation, la température est élevée, puis elle est abaissée lentement. Ce procédé, connu sous le nom de « recuit simulé » *(« simulated annealing »)* [73], a pour but d'empêcher que le système ne se stabilise dans un minimum local d'énergie (correspondant à une solution non optimale), ayant pour conséquence de rater le vrai minimum, en aidant le système à quitter ce minimum local.

Pendant l'apprentissage, on donne tout d'abord à chaque nœud une valeur 0 ou 1 de façon aléatoire. Puis on présente le stimulus aux nœuds d'entrée, et la réponse désirée aux nœuds de sortie. La méthode de *recuit simulé* permet d'obtenir le meilleur équilibre, correspondant à l'énergie minimale pour le réseau complet. Pour l'ensemble du corpus d'apprentissage, on conserve pour chaque arc les statistiques sur le nombre de fois où les nœuds aux extrémités de l'arc étaient à 1 en même temps. Le même procédé est ensuite utilisé sans donner aucune information aux nœuds de sortie. La comparaison des deux résultats permet l'apprentissage du réseau, c'est-à-dire la mise à jour des poids affectés aux arcs, en diminuant, ou en augmentant leur valeur par une valeur fixe. Le processus de reconnaissance consiste, lui, à appliquer le stimulus aux nœuds d'entrée du réseau, à utiliser le recuit simulé pour obtenir la solution optimale, et à considérer les nœuds de sortie pour obtenir la forme reconnue.

Cette approche a été utilisée dans le cadre d'expériences de reconnaissance de voyelles multilocuteur, en utilisant comme entrée un spectre pour représenter la voyelle prononcée en mode isolé [131]. Elle a été également comparée à un MLP avec le même nombre de nœuds (50 nœuds cachés) [132]. Les résultats ont montré que la Machine de Boltzman était légèrement meilleure que le MLP (environ 3 % de différence : 2 % d'erreur contre 5 % sur les données utilisées pour l'apprentissage après 25 cycles d'apprentissage, 39 % contre 42 % sur des données inconnues des mêmes locuteurs, après 15 cycles d'apprentissage). Il a été également remarqué que le MLP était environ 10 fois plus rapide que la Machine de Boltzman.

6.3. LES CARTES PHONOTOPIQUES (Feature Maps)

Les Cartes Phonotopiques (Feature Maps, ou Phonotopic Maps) [75], sont basées sur l'hypothèse que, pour la reconnaissance de la parole, les informations qui sont de nature proche, doivent être également proche topologiquement, comme cela semblerait être le cas dans le cerveau [76]. C'est une approche non supervisée dans la mesure où aucune information n'est donnée au système sur la sortie désirée lorsqu'on construit la carte.

Le procédé est semblable à la classification automatique. Le réseau peut être représenté comme une grille en deux dimensions. Chaque point de la grille correspond à un spectre prototype. Lorsqu'un nouveau spectre des données de parole est présenté, il est comparé à tous les prototypes existants, en utilisant une mesure de similarité comme la distance Euclidienne. Lorsque le spectre prototype le plus proche est trouvé, le prototype correspondant est moyenné avec le nouveau spectre, en prenant comme coefficient de pondération le nombre de spectres qui étaient intervenus dans la construction du spectre prototype. Les huit voisins adjacents sont également modifiés en tenant compte de la nouvelle entrée, avec une influence moindre. La même modification s'applique également aux 16 voisins suivants (*fig.* 13). A la fin du processus, la quantification est réalisée, comme elle le serait avec une approche par classification automatique, mais, de plus, chaque prototype est proche des prototypes qui lui sont similaires. La qualité de ce quantificateur *(mesure de distorsion)* a été comparée à des quantificateurs traditionnels, à leur avantage [176]. Le réseau est ensuite étiqueté, en reconnaissant des phrases elles-mêmes étiquetées, et en plaçant les étiquettes correspondantes aux nœuds de la grille, avec une stratégie de décision appropriée. Le processus de reconnaissance correspond à une trajectoire dans la grille étiquetée (appelée aussi *carte phonotopique*).

Cette approche a été appliquée avec succès à la reconnaissance du Finnois et du Japonais (monolocuteur, en mots isolés, pour un vocabulaire de 1 000 mots). Le taux de reconnaissance phonétique varie de 75 % à 90 %, le taux de reconnaissance des mots varie de 96 % à 98 %, la transcription orthographique d'un mot varie de 90 % à 97 %, suivant le vocabulaire et le locuteur [77].

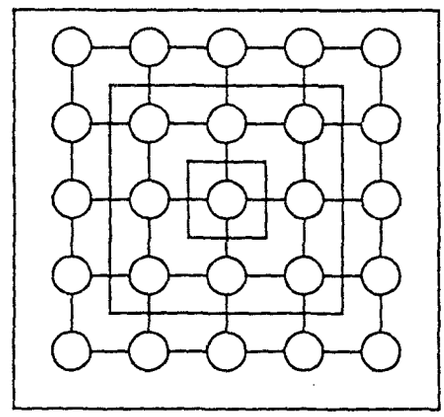

Figure 13. — Exemple d'une architecture de Carte Phonotopique.
Chaque cellule correspond à un prototype en quantification vectorielle, relié à ses voisins, qui sont des prototypes similaires. Si la cellule du milieu est modifiée, ces 8 proches voisins, et les 16 suivants seront également modifiés.

6.4. PROPAGATION GUIDÉE

Un autre type de système utilise le principe de la *Propagation Guidée*, basé sur une mémoire topographique. La parole est transformée en un spectre d'événements de

simulation discrets et localisés qui sont traités au fur et à mesure. Ces événements nourissent un flot de signaux internes qui se propagent le long de chemins parallèles en mémoire correspondant à des unités de parole (des mots par exemple). Comparée aux méthodes par couches décrites plus haut, cette architecture suppose un ensemble d'unités de traitement organisées en chemin entre les couches. Très schématiquement, chacune de ces unités contextuelles détecte les coïncidences entre l'activation interne qu'elle reçoit du chemin auquel elle participe (contexte), et les événements de simulation externes.

Cette approche a été utilisée pour la reconnaissance monolocuteur des chiffres isolés (0-9) dans le bruit, sur une base de données de test limitée. Le bruit est lui-même constitué de parole (une prononciation du nombre 10 par le même locuteur), avec un rapport Signal/Bruit de 0 dB. Elle a été comparée à l'algorithme classique de reconnaissance par Programmation Dynamique. Les résultats sans bruit ont été de 0 % d'erreurs avec la programmation dynamique, et 2 % d'erreurs pour le modèle connexionniste. Lorsqu'on ajoute le bruit, cela donne 47 % d'erreurs pour la programmation dynamique, et 10 % d'erreur pour le modèle connexionniste. Cependant, il faut noter que le traitement du signal était différent dans les deux cas (coefficients cepstraux pour la programmation dynamique, modèle d'oreille simplifié, incluant inhibition latérale et adaptation à court terme, pour le système connexionniste) [15, 82].

6.5. AUTRES APPROCHES

D'autres systèmes connexionnistes, qui peuvent être appliqués à la reconnaissance de forme en général, et à la reconnaissance de parole en particulier, existent. Le Réseau d'Hopfield *(« Hopfield Net »)*, a une seule couche, chaque cellule étant reliée à toutes les autres cellules. On l'utilise comme une mémoire associative, et il peut rétablir des signaux bruités. Le réseau de Hamming est semblable au réseau d'Hopfield mais calcule tout d'abord une distance de Hamming pour comparer le vecteur d'entrée aux formes de référence [98].

D'autres approches sont actuellement poursuivies, sans que des résultats définitifs aient été publiés. J. L. Elman et J. L. McClelland ont proposé le modèle TRACE comme un modèle intéressant de perception auditive, ou une architecture pour le traitement en parallèle de la parole. La première version, TRACE I, acceptait le signal vocal en entrée [41]. Une version améliorée, TRACE II, accepte seulement les traits acoustiques comme entrée [100]. Fondée sur des données neurobiologiques, l'approche par Colonnes Corticales a été appliquée à la reconnaissance vocale [58].

6.6. UTILISATION DES RÉSEAUX NEURONAUX POUR LA MODÉLISATION DU LANGAGE

L'utilisation d'une approche auto-organisatrice a montré son efficacité dans la modélisation du langage, comme cela apparaît avec les modèles de langage Markoviens. Quelques premiers essais utilisant les réseaux neuronaux peuvent également être répertoriés. Un de ces travaux a consisté à essayer d'étendre les modèles de Bigrammes, ou de Trigrammes à des modèles de Ngrammes [118]. Pour un modèle de Bigramme de base, le MLP qui est utilisé a 89 cellules d'entrée (correspondant à 89 catégories grammaticales) pour le mot N, et 89 cellules de sortie pour donner la catégorie grammaticale du mot N + 1. Il y a deux couches cachées avec 16 cellules dans chacune. Ce MLP a été généralisé à des 4-grammes (3 groupes de 89 cellules en entrée). Il a été appris sur 512 phrases, et testé sur 512 autres phrases. Pour un modèle de Trigrammes, les résultats sont comparables à ceux obtenus avec une approche Markovienne, l'information étant réduite plus de 130 fois. L'examen des cellules cachées montre qu'elles semblent organiser les catégories de mot dans des groupes linguistiquement signifiants.

Bien que l'approche par réseau neuronal semble très intéressante et pleine de promesses, plusieurs problèmes sont encore non résolus. Quelle architecture choisir ? Combien de couches, combien de cellules utiliser ? Comment traiter la modélisation du temps ? Que doit être la représentation des paires stimulus-réponse ? Comment diminuer le temps de calcul, encore prohibitif ? Actuellement, aucune expérience déterminante, dans des conditions rigoureusement identiques, sur un corpus suffisamment large, et sur une tâche suffisamment générale, prenant réellement les avantages et les intérêts respectifs des deux types d'approches, n'a prouvé la supériorité de la méthode par réseau neuronal sur les méthodes statistiques, ou par comparaison de forme.

7. Méthodes « À Base de Connaissances »

7.1. PRINCIPE

L'approche « A Base de Connaissances » devint très populaire lorsque la technique des « Systèmes Experts » fut proposée en Intelligence Artificielle. L'idée est de séparer la connaissance à utiliser dans un processus de raisonnement (la *Base de Connaissance*), de la stratégie, ou mécanisme de raisonnement sur cette connaissance (basée sur un *moteur d'inférence*, qui active des règles). La stratégie de raisonnement est également reflétée dans la façon dont les informations à l'entrée (les *Faits*) sont traitées (de la gauche vers la droite, ou à partir de points d'ancrage). La plupart des manipulations de l'information, incluant l'introduction des données à traiter, est faite au niveau de la *Base de Faits*. La connaissance est représentée par des règles du type « **Si** *Faits* **alors** *Conclusion 1* **sinon** *Conclusion2* ». Ces règles peuvent être accompagnées d'un poids représentant, comme une heuristique, la confiance que l'on peut accorder à la règle. Le Moteur d'Inférence peut essayer de mettre en correspondance les buts à atteindre au signal d'entrée, en appliquant les règles à partir de la base de connaissances, en commençant par les buts présents dans les conclusions des règles, et en examinant si le résultat de telles activations est en fait le signal d'entrée (méthodes appelées en « Chaînage

Arrière » *(Backward Chaining)*, « dirigées par le but » *(Goal Directed)*, ou « par la connaissance » *(Knowledge Driven)*). Ou, au contraire, il peut commencer à partir du signal d'entrée, trouver des règles applicables, et les activer jusqu'à ce qu'un but soit obtenu (« Chaînage avant » *(Forward Chaining)*, ou « Guidé par les données » *(Data Driven)*). La stratégie peut changer pendant le processus de décodage, sur la base de résultats intermédiaires.

Cette approche implique que la connaissance soit introduite manuellement, à moins que des processus d'apprentissage automatique soient utilisables. Les efforts nécessaires à l'obtention d'une quantité de connaissances suffisante, pour la reconnaissance multilocuteur de parole continue, et pour des grands vocabulaires, ont été mesurés au début des premiers travaux utilisant cette approche (au début des années 80) comme devant durer environ 15 ans.

7.2. LES SYSTÈMES EXPERTS DE LECTURE DE SPECTROGRAMMES

Comme il a été montré que certains experts en lecture de spectrogrammes étaient capables de « lire » des spectrogrammes de parole avec un score de réussite élevé (80 à 90 %), plusieurs essais ont été tentés pour imiter ces experts dans un système expert à base de connaissances [32].

L'expert a des discussions avec un « ingénieur cogniticien » (habituellement un informaticien), qui a le rôle d'extraire les faits, la connaissance et les stratégies avec lesquelles l'expert applique ses connaissances sur les faits. La plupart du temps, de telles approches ont pour but d'étudier un ensemble précis de phonèmes pour un locuteur donné [167], ou un ensemble de phonèmes dans un contexte particulier, comme les occlusives en début de mot, pour n'importe quel locuteur (à MIT [179]), ou même des indices acoustiques spécifiques.

Un problème réside dans le fait que l'expert, avant d'appliquer ses règles, utilise des indices visuels, qui sont difficiles à représenter par des règles s'appliquant à des symboles. Une façon d'éviter ce problème de perception visuelle, qui est du ressort de la Vision par Ordinateur, est de vérifier manuellement tous les traits mesurés par le système [179], ou de prendre comme entrée une liste de traits donnés par l'utilisateur lisant le spectrogramme. Le système expert peut prendre l'initiative de poser des questions [167].

7.3. AUTRES APPROCHES

A part les projets de « systèmes experts de lecture de spectrogrammes », des travaux ont été menés à MIT sur la segmentation et l'étiquetage de parole avec une approche basée sur les connaissances [180]. Le processus de segmentation produit une représentation à plusieurs niveaux, appelée un *« dendrogram »*, très similaire à l'idée de filtrage *« scale-space »* utilisé dans d'autres domaines comme la Vision par Ordinateur [175]. Le spectrogramme de parole est segmenté en unités de différents niveaux de description, des plus fins aux plus larges, le dernier segment étant la phrase elle-même. Ce procédé est basé sur le calcul d'une mesure de similarité entre les segments adjacents, utilisant une distance Euclidienne sur les spectres moyens de chaque région préalablement délimitée, et sur la fusion des segments semblables. Les résultats de segmentation sont de 3,5 % d'omission, et de 5 % d'insertion, pour 100 locuteurs. Un treillis phonétique est alors produit en utilisant un classifieur statistique. La représentation lexicale tient compte de plusieurs prononciations différentes pour chaque mot. Le résultat est un treillis de mots. Sur un corpus de test de 225 phrases, avec un vocabulaire moyen de 256 mots, en considérant l'ordre des mots qui commencent en même temps que le mot correct, mais ayant une note de ressemblance meilleure que celle du mot correct, il a été montré que le mot correct arrive en première position dans 32 % des cas, parmi les 5 premiers dans 67 % des cas, et parmi les 10 premiers dans 80 % des cas. Le taux de reconnaissance des allophones correspondants est de 70 % (allophone correct en première position) et 97 % (allophone correct dans les 5 premiers).

Le système « **Angel** » [32] a été développé au sein du programme DARPA à CMU, pour réaliser la reconnaissance multilocuteur, en parole continue, de grands vocabulaires. La reconnaissance se fait en utilisant des modules de localisation, et de classification, qui ont pour tâche de segmenter et d'identifier les segments correspondants. La sortie est un treillis de phonèmes, avec des probabilités affectées aux étiquettes pour chaque segment. Ces modules sont par exemple affectés à la reconnaissance des occlusions, des frications, des explosions, ou du caractère « voisé ». Le système a été testé sur la Base de données DARPA « Resource Management », avec des résultats relativement médiocres.

Certains travaux ont eu pour but d'intégrer l'approche à base de connaissances avec une approche stochastique de type HMM [59]. D'autres ont tenté d'utiliser des architectures à base de connaissances plus complexes, comme la structure de Société de Spécialistes *(Specialist Society)* [56], ou de Société de Systèmes Experts, avec de l'apprentissage symbolique inductif [36].

8. Traitement de la Parole et du Langage Naturel

Maintenant que certains systèmes de reconnaissance vocale atteignent des résultats acceptables dans des conditions acceptables (taille de vocabulaire suffisamment importante pour permettre des applications intéressantes, prononciation continue, reconnaissance de n'importe quel locuteur avec peu, ou pas d'apprentissage préalable), l'une des difficultés principales qui demeurent est le lien avec le traitement du langage qui sera utilisé dans l'application. Ce langage peut contenir des structures de phrase qui ne sont pas dans la syntaxe, des mots qui ne sont pas dans le lexique, des hésitations, du bégaiement, etc. Actuellement, les démonstrations des systèmes les plus avancés nécessitent de la part des locuteurs la prononciation d'une phrase lue parmi une liste contenant les phrases accepta-

bles, de manière à ce que les phrases lues suivent les règles de la syntaxe, et que seuls les mots acceptables soient utilisés.
Lorsqu'un utilisateur réel prononce une phrase qui n'a pas été prévue dans la syntaxe, cela va poser des problèmes au système. Bien sûr, moins la syntaxe est contrainte, et plus le système sera capable d'accepter des phrases qui dévient d'une description manuelle préalable des phrases possibles. Mais, en même temps, la syntaxe aidera moins à éviter les erreurs de reconnaissance acoustique. Cela est déjà le cas lorsqu'une grammaire apprise, par paire-de-mots ou par bigrammes, est utilisée à la place d'une grammaire déterministe hors contexte par exemple.

Dans le cas du texte écrit, un apprentissage est réalisable en utilisant une quantité de données textuelles suffisante. Pour le dialogue parlé, ces données sont difficiles à obtenir. De plus, si une « compréhension » en profondeur n'est pas nécessaire pour une tâche de dictée vocale, elle est une obligation dans une tâche de dialogue, où le système doit produire la réponse appropriée (génération d'une réponse ou action correspondante). Le lien avec le Traitement du Langage Naturel est alors une nécessité. Le problème alors est que la plupart des méthodes de Traitement du Langage Naturel sont utilisables dans le cas d'entrée déterministe (une suite de mots dépourvue d'erreurs). Dans le cas de la parole, la séquence de mots est ambiguë, à la fois au niveau du décodage acoustique et au niveau de la segmentation en mots, et le processus de « compréhension » lui-même intervient pour résoudre ces ambiguïtés ! Des premières tentatives pour traiter ce problème dans des conditions réalistes peuvent être mentionnées. Cependant, l'utilisation d'informations relatives à la sémantique, ou à la pragmatique de la tâche, vont réduire la généralité d'un système à la tâche pour laquelle on veut l'utiliser.

Très peu de systèmes avancés intégrant reconnaissance de parole et traitement du langage naturel peuvent être mentionnés. Dans le système MINDS à CMU [177], le but est de réduire la complexité du langage en étant capable d'effectuer les prédictions des concepts qui pourraient être présents dans la prochaine phrase durant un dialogue, faisant ainsi des prédictions sur le vocabulaire et la syntaxe correspondantes. Les informations utilisées sont la connaissance de plans de résolution de problèmes, la connaissance sémantique du domaine d'application, la connaissance indépendante de la tâche sur les mécanismes de locution, l'historique du dialogue, l'expertise de l'utilisateur, et aussi les habitudes linguistiques du locuteur. Les corpus d'apprentissage et de test provenaient de la base de données TONE (NOSC, US Navy). L'utilisation de telles informations a réduit la perplexité du corpus de test de 242,4 mots (si l'on n'utilise que la grammaire) à 18,3 mots. Avec le système SPHINX, pour 10 locuteurs, chacun prononçant 20 phrases, en mode multilocuteur, l'acuité sur les mots est passée de 82,1 % à 96,5 %, et l'acuité sémantique de 80 % à 100 %.

Dans le projet VODIS poursuivi dans le cadre de l'initiative Alvey au Royaume Uni, la tâche consiste en un système d'interrogation vocale de base de données, accessible aux utilisateurs non spécialistes via le réseau téléphonique [178]. Le système de reconnaissance de parole continue, de type programmation dynamique, est relié à un module de contrôle du dialogue. VODIS I, utilise des contraintes syntaxiques très fortes. Dans VODIS II, le but est d'alléger ces contraintes, à la suite d'expériences réalisées en mode opérationnels avec des utilisateurs « naïfs ». Le processus de reconnaissance, qui inclut des contraintes syntaxiques données par une grammaire hors contexte, produit des listes de mots reliées entre elles, à partir desquelles un treillis de mots possibles peut être généré. Ce treillis est analysé par un analyseur syntaxique de type analyseur synoptique (*« chart parser »*) ascendant, et les solutions résultant de cette analyse qui offrent les meilleures notes de reconnaissance sont converties suivant un format de schémas, incluant les solutions arrivant dans les meilleures positions. La sémantique et la pragmatique de la tâche sont appliquées aux schémas pour obtenir la solution acceptable la meilleure. A BBN, un analyseur synoptique est également utilisé sur un treillis de mots [13].

Un autre projet, mené au LIMSI, concerne l'utilisation du dialogue oral pour l'entraînement au contrôle aérien [107]. Le système de reconnaissance de type DTW est relié à une représentation des connaissances sous forme de schémas. A une étape du dialogue, une phrase est reconnue par un algorithme de DTW optimal, en utilisant une syntaxe faiblement contrainte. L'analyse des phrases détermine dans quelle catégorie elle se place et instancie le schéma correspondant, en mettant les mots dans les cases du schéma. Un procédé de contrôle de validité, qui utilise des contraintes syntactico-sémantiques associées au schéma, détecte les erreurs du système, ou du locuteur. La correction d'erreur peut être faite en examinant dans une *matrice de confusion entre mots* les mots qui pourraient être confondus avec ceux qui ont été reconnus, et sont également syntaxiquement et sémantiquement acceptables, ou en lançant une nouvelle reconnaissance sur le même signal avec des paramètres différents, ou en générant une question vers l'utilisateur. L'interprétation du message donne une séquence de commandes au simulateur de contrôle aérien, met à jour le contexte de la tâche et l'historique du dialogue.

9. Progrès réalisés conjointement dans d'autres domaines du traitement automatique de la parole

Dans la mesure où la quantification vectorielle a été initialement conçue pour le codage de parole à très bas débit (aux alentours de 800 bits/s), on trouvera beaucoup d'applications de la QV dans ce domaine. On trouvera également des essais de codage segmental. Le but initial était de réaliser un vocoder phonétique, à partir de l'idée que, si le débit initial du codage PCM (Pulse Code Modulation, simple échantillonnage) est de l'ordre de 64 Kbits/s, le débit de transmission des phonèmes après reconnaissance serait de 50 bits/s ! De plus, il peut n'être pas nécessaire de reconnaître la suite de phonèmes sans faire d'erreurs, dans la mesure où les « niveaux supérieurs » de l'auditeur humain peuvent amener des informa-

tions supplémentaires permettant de restituer le sens de la phrase, malgré les erreurs de reconnaissance phonétique. Des expériences conduites en modifiant les phonèmes dans un synthétiseur à partir du texte en Français [93] ont montré qu'un taux d'erreur phonétique supérieur à 15 %, ou même une seule erreur « grave », pouvait empêcher la reconnaissance de la phrase complète. Cela donne une idée du taux de reconnaissance phonémique minimal à atteindre, dans une situation où le système de compréhension de la parole aurait des niveaux supérieurs aussi performants que ceux d'un être humain (pour un univers sémantique indéfini au départ).
De façon intéressante, les premières méthodes basées sur la reconnaissance par diphonèmes [154] n'ont pas abouti à un taux de reconnaissance suffisant, et n'ont donc pas permis une transmission suffisamment intelligible. Alors que de nouveaux essais tendant à utiliser le codage segmental, sans étiqueter phonétiquement ces segments, ont donné des résultats acceptables, avec un débit légèrement supérieur (autour de 200 bits/s) [147]. Comme pour la comparaison du modèle *Centiseconde* avec la quantification vectorielle, il est également visible ici que l'étiquetage ne doit être fait que lorsque l'on peut, et que l'on doit, prendre des décisions. Des segments du même type ont été utilisés pour la synthèse vocale [117].

Les techniques de quantification vectorielle et segmentale ont été utilisées en vérification du locuteur [166, 60], et en transformation de voix [1]. Pour la transformation de voix, le répertoire d'un locuteur est mis en correspondance avec le répertoire d'un autre locuteur, donnant ainsi la correspondance pour chaque spectre prototype. Quand le locuteur de référence prononce une phrase, le processus de quantification vectorielle est appliqué, et chaque étiquette est remplacée par l'étiquette correspondante pour l'autre locuteur. Cela a pour résultat la même phrase prononcée avec le timbre de l'autre locuteur. Cette approche a été utilisée par ATR pour synthétiser la traduction en Japonais d'une phrase initialement prononcée en anglais, avec la voix qu'aurait le locuteur anglais s'il parlait japonais...

Les HMMs ont été utilisés pour le suivi de formants, l'estimation du fondamental et le marquage des accents d'intensité (en synthèse vocale), en reconnaissance du locuteur, en analyse linguistique des textes écrits (pour choisir les règles de conversion graphème-phonème convenables), en reconnaissance de caractères, en traduction automatique, etc.
Comme cela a déjà été mentionné, les approches connexionnistes ont été utilisées, non seulement pour la conversion graphème-phonème, mais aussi pour l'amélioration du signal vocal bruité, pour le traitement du signal, pour l'affectation des marqueurs prosodiques en synthèse vocale. Une application possible dans le futur est l'utilisation de tels modèles pour gérer la communication multimodale.

10. Matériel d'accompagnement

L'arrivée de circuits intégrés spécialisés pour le traitement du signal vocal a été d'une importance certaine dans l'histoire du Traitement Automatique de la Parole. Texas Instruments (TI) a initialisé ce processus, avec son circuit de synthèse LPC dans le jeu électronique Speak'n Spell dès 1978.

10.1. CIRCUITS DE TRAITEMENT NUMÉRIQUE DU SIGNAL *(DSPs)*

Les circuits DSP ont permis un traitement numérique du signal en temps réel, avec des transformées diverses, amenant par là-même une analyse invariante, de la flexibilité, et un encombrement moindre. Le premier exemple de tels circuits a été le 2920 d'Intel, suivi du NEC 7720. Les circuits DSP plus récents sont ceux de la famille TI TMS 320, les AT&T DSP16 et DSP32, le MOTOROLA DSP56000, l'ADSP-2100 d'Analog Devices, etc. Alors que les premiers circuits ne permettaient que le calcul en entier, les plus récents, comme le TMS 320C30 [169], le DSP56000 [163] ou le DSP32 [18], permettent le calcul en virgule flottante.

10.2. CIRCUITS DE PROGRAMMATION DYNAMIQUE

Les circuits de Programmation Dynamique sont spécialisés dans le calcul de cet algorithme. Dans la mesure où cet algorithme nécessite beaucoup de puissance de calcul, il est souhaitable d'avoir des outils qui le fassent rapidement. Cela permet de traiter des vocabulaires plus grands, ou d'améliorer la qualité des résultats de reconnaissance en permettant l'utilisation d'algorithmes optimaux, et non pas sous-optimaux à cause de l'introduction d'heuristiques. NEC a proposé son ensemble de circuits *(« chip set »)* pour la reconnaissance de mots isolés (NEC 7761-7762) en 1983. Ils ont aussi présenté un circuit de reconnaissance de mots enchaînés par programmation dynamique (NEC 7764) au même moment [65]. A Berkeley, un circuit a été développé pour la reconnaissance de 1 000 mots isolés en 1984 [71]. Un nouveau circuit est à l'étude actuellement, en collaboration avec SRI, pour la reconnaissance de 1 000 mots en continu, qui devrait être capable d'effectuer l'algorithme de Viterbi pour des HMMs discrets, avec une vitesse de 75 000 à 100 000 arcs par spectre en temps réel [112]. VECSYS [133] et AT&T [53] proposent des circuits de DTW avec une puissance comparable. Le MUPCD de VECSYS est annoncé pour 70 MOPS (Millions d'Opérations par Seconde), et reconnaît 5 000 mots isolés, ou 300 mots en parole continue, en temps réel. Le GSM *(Graph Search Machine)* d'AT&T est annoncé pour 50 MIPS (Millions d'Instructions par Seconde). Il a également été essayé pour faire de la reconnaissance en utilisant un réseau de Hopfield [114].

10.3. ARCHITECTURES SPÉCIALES

La nécessité de puissance de calcul peut conduire à des architectures spéciales. En développant son circuit propre « Hermes », qui a été intégré dans sa carte PI pouvant être placée dans un bus de PC-AT, IBM a été capable en 1986 de remplacer un système qui tournait en 1984 sur 3 calculateurs vectoriels, un IBM 4341, une station de travail

Apollo (et un PC !), avec une plus grande crédibilité, apportant la preuve que les HMMs n'étaient pas uniquement un outil mathématique réservé aux « accros » des Super Ordinateurs !

A CMU, le système SPHINX a bénéficié de l'architecture de la machine BEAM (3 000 arcs par spectre en temps réel) [17]. L'algorithme *« Level Building »* a été installé à AT&T sur le système ASPEN à structure arborescente [54]. Les aspects intéressants de parallélisme offerts par les transputers ont aussi conduit à des résultats nouveaux [29]. Cependant, la taille de l'effort nécessaire à installer le logiciel sur une architecture spécialisée sans le support des langages de haut niveau standards, doit être comparée à l'accroissement de performances espérées pour justifier qu'on le fasse. Des compilateurs C, ou C vectorisés, sont proposés sur les architectures mentionnées plus haut.

11. Évaluation

Un des plus gros échecs du projet ARPA-SUR, mené de 1971 à 1976, a été qu'à la fin, il s'est avéré difficile de comparer les systèmes, dans la mesure où ils étaient testés sur des langages complètement différents, ayant des difficultés différentes, et correspondant à des tâches complètement différentes. Seuls des systèmes différents venant d'un même laboratoire ont été comparés sur des données communes (comme les systèmes HEARSAY et HARPY à CMU). Le problème était que dans le cahier des charges initial, seule la taille du vocabulaire était donnée, pas la difficulté du langage. Dans le projet DARPA actuel, un soin tout particulier a été placé sur la définition d'une méthodologie d'évaluation, sur l'élaboration des bases de données de parole correspondantes, résultant ainsi en la conduite de tests réguliers, et sur la comparaison de plusieurs systèmes sur les mêmes données. Cela est valable aussi bien pour évaluer les améliorations pendant le développement d'un système, que pour la comparaison des résultats venant de différents laboratoires ayant des approches semblables, ou très différentes.

La mesure a priori de la difficulté d'un langage à reconnaître est difficile. Cela inclut à la fois les contraintes amenées par la syntaxe, et la ressemblance acoustique des mots, s'ils peuvent être prononcés au même moment à un même nœud de la syntaxe. La perplexité du langage donne sa difficulté indépendamment des similarités acoustiques entre les mots. Elle peut être calculée à partir de l'entropie du langage. Cela est facilement faisable pour une syntaxe décrite par un automate d'états finis. Si la syntaxe est de type local, comme c'est le cas pour les bigrammes ou les paires de mots, il faut la calculer à partir du corpus de test (perplexité du corpus de test) [72].

Pour tester un système de reconnaissance, l'idée de base est de construire un grand corpus de test, et de tester tous les systèmes sur ce même corpus. Texas-Instruments a été parmi les premiers à réaliser des bases de données importantes. En France le GRECO du CNRS a suivi. Le groupe RSG10 de l'OTAN a construit une base de données multilingue en 1980. Suivant la taille de la base de données et les performances d'un système, la précision des résultats sera plus ou moins significative.

11.1. RÉSULTATS DE RECONNAISSANCE SUR LES MOTS ET LES PHRASES

La méthodologie de test est également importante ; La technique de détermination des scores de reconnaissance elle-même doit être définie avec beaucoup de soin. Pour la parole continue, différents types d'erreurs peuvent intervenir : erreur de substitution (un mot est reconnu à la place d'un autre), erreur d'insertion (un mot est reconnu alors que rien n'a été prononcé) et erreur d'omission (rien n'est reconnu alors qu'un mot a été prononcé). Deux mesures de performance ont été proposées. Le « Pourcentage de mots corrects » *(« Percent Correct »)*, qui se réfère à la suite de mots prononcées, et examine combien de ces mots ont été correctement reconnus. Il ne tient donc pas compte des erreurs d'insertion. L'autre mesure, appelée « Acuité lexicale » *(« Word Accuracy »)* considère les trois types d'erreurs. L'addition de ces trois types d'erreurs peut conduire à un taux de reconnaissance négatif (par exemple, la reconnaissance de *« You need »* au lieu de *« unit »* compte pour deux erreurs d'insertion (ou une de substitution et une d'insertion) pour la prononciation d'un seul mot). Un logiciel, basé sur la programmation dynamique, permettant d'extraire automatiquement le score de reconnaissance, a été conçu par NIST (National Institute of Standards and Technology, anciennement NBS (US National Bureau of Standards)). Il est utilisé pour tester les systèmes réalisés dans le projet DARPA actuel [123]. NIST continue à présent son effort dans cette direction aux USA. Au niveau Européen, le projet SAM a pour but de définir et d'utiliser des bases de données de parole multilangues importantes. Un effort similaire est en cours au Japon. Les tableaux suivants donnent des exemples de systèmes qui ont été testés.

11.2. RÉSULTATS DE RECONNAISSANCE PHONÉTIQUE

En utilisant un modèle très proche de celui qui a été utilisé dans le système SPHINX, K. F. Lee à CMU, a essayé de reconnaître des phonèmes dans la parole continue en mode multilocuteur. La base de données de test est la base de données TIMIT. 2 830 phrases prononcées par 357 locuteurs ont été utilisées pour l'apprentissage, et 160 phrases prononcées par 20 locuteurs pour le test. Le modèle global est un « modèle phonétique en boucle » (*Looped Phonetic Model* (LPM)), les meilleurs résultats ayant été obtenus avec des modèles de phones contextuels à droite. Un modèle de bigrammes de phones peut être utilisé pour donner la probabilité des successions de deux phonèmes. Un modèle d'unigrammes donnant la probabilité d'un phonème a également été essayé [85]. Des tests comparables ont également été conduits au Centre Scientifique d'IBM à Paris, en mode monolocuteur, avec un apprentissage par MMI, et un modèle de bigrammes phonémiques [109], et à l'Université d'Helsinki, avec une approche par *Cartes Phonotopiques* [77]. On voit que l'on s'approche des 85 % fatidiques.

TABLEAU 4

Taux de reconnaissance phonémiques et types d'erreurs

Type d'erreur	SPHINX (Indépt)	IBM-F (Dépt)	Helsinki (Dépt)
Correct	73,8 %	84,8 %	75 %-90 %
Substitutions	19,6 %		
Omissions	6,6 %		
Insertions	7,7 %	4,18 %	

TABLEAU 5

Taux de Reconnaissance Phonémique par catégories phonétiques grossières, et suivant le modèle linguistique « phonémique » utilisé dans le système SPHINX [87]

Classe	% correct	Modèle Phon.	% correct
sonores	65,7 %	bigramme	73,8 %
plosives	69,9 %	unigramme	70,4 %
fricatives	78,4 %	aucun	69,5 %
occlusives	93,3 %		

12. Les développements

12.1. LE POINT SUR LES APPLICATIONS

Si les systèmes de traitement automatique de la parole ont atteint une qualité très convenable, leur utilisation reste encore très relative. De fait, les meilleurs systèmes sont testés avec des corpus de données qui sont des textes lus, correspondant à une syntaxe correcte. Par contre, lorsqu'on quitte l'ambiance du laboratoire pour confronter le système à des utilisateurs réels, on rencontre de nombreux problèmes qui vont très nettement diminuer les performances, et rendre le système inutilisable, par manque de robustesse. Ainsi, les problèmes de bruits de l'environnement, d'hésitations, de bégaiement, d'utilisation de mots qui ne sont pas dans le lexique, ou de tournures qui ne sont pas dans la syntaxe sont rarement traités. Pour être capable de traiter ces problèmes, et pour extraire le sens de l'acte de parole, il est nécessaire de « comprendre » le message, et cette compréhension nécessite d'établir une relation très étroite avec les travaux menés dans le cadre du traitement du langage naturel, y compris ceux relatifs à la gestion de l'implicite.

La nécessité d'apprendre des connaissances complexes, et les résultats intéressants des méthodes auto-organisatrices font apparaître la nécessité de traiter divers modes de communications (vision, texte, mouvement) pour pouvoir les intégrer dans l'élaboration d'un modèle permettant de reconnaître et de comprendre la parole. L'intérêt d'une communication multimodale apparaît également pour la communication homme-machine, et le problème de la fusion des informations provenant de plusieurs canaux différents se pose maintenant que ces moyens coexistent avec des performances suffisantes. La gestion de l'implicite la rend également nécessaire.

12.2. LES MARCHÉS

Les études de marché portant sur les technologies vocales ont toujours péché par optimisme depuis 1980. Il faut néanmoins souligner que le marché a atteint en 1989 le Milliard de $ aux États-Unis, et que les prévisions effectuées en 1984 s'avèrent aujourd'hui correctes. On peut penser que son éclosion viendra avec l'évolution de l'informatisation de la société, qui devra nécessairement accorder une priorité importante à l'interface convivial entre l'homme et la machine.

12.3. LES ACTEURS

Si l'on considère la conférence IEEE ICASSP qui en est à sa quinzième édition annuelle, et qui est une référence admise de qualité, on peut recenser 2 000 articles concernant le Traitement Automatique de la Parole (analyse, synthèse, reconnaissance, codage, amélioration du signal vocal, reconnaissance du locuteur ou de la langue) [105].

Si l'on considère les pays individuellement, 138 pays ont participé, mais 7 pays (États-Unis, Japon, France, Grande-Bretagne, RFA, Italie et Canada) ont publié 90 % des articles.

Le pays le plus productif est les USA, qui ont publié la moitié des articles. L'Europe est responsable du quart, et le Japon d'un huitième.

La France vient au troisième rang, derrière le Japon, avec 160 articles, devant la Grande-Bretagne, la RFA et l'Italie.

Le nombre de laboratoires varie suivant les pays. 150 laboratoires américains environ ont publié, 120 en Europe (dont 25 en France), 40 au Japon, et 70 dans les autres pays, pour un total de 380 laboratoires. On peut estimer à 3 000 personnes la population travaillant sur la recherche et le développement en matière de Traitement Automatique de la Parole en Europe [104].

Les 11 laboratoires qui ont publié le plus (AT&T Bell Labs (10 % du tout), BBN, MIT, Lincoln Lab, CMU, IBM-Yorktown, NTT (Japon), Texas-Instruments, CNET (France), CSELT (Italie) et Georgia Tech) sont responsables de 30 % des articles.

13. Conclusions

Nous avons vu que des améliorations intéressantes peuvent être notées dans les avancées récentes en reconnaissance de la parole. Pour résumer, l'utilisation de grandes bases de données, accompagnées de procédures d'apprentissage élaborées, peuvent permettre une reconnaissance multilocuteur de parole continue sur un vocabulaire de taille moyenne (1 000 mots) avec des résultats acceptables, si l'on ne considère du moins que le taux de reconnaissance ramené aux mots. En même temps, le taux de reconnaissance phonémique a également atteint un niveau de qualité qui permet d'envisager l'utilisation de systèmes de reconnaissance dans des tâches complexes. Les Modèles de Markov Cachés ont prouvé qu'ils étaient des outils

puissants. Les modèles connexionnistes peuvent apporter de nouvelles possibilités, et de nouvelles améliorations. Le lien avec le traitement du langage naturel est maintenant une nécessité, afin de déterminer la robustesse des systèmes lorsqu'ils sont confrontés à de la parole courante dans des applications réelles, et de les rendre utilisables.

Quelques résultats importants ont été découverts :

— Il n'est pas nécessaire de réaliser une segmentation a priori, dans la mesure où une segmentation implicite est faite lors du processus de reconnaissance lui-même.

— Plus il y a de données pour l'apprentissage, meilleurs seront les résultats de reconnaissance. Dans la mesure où il est facile d'avoir des données provenant de plusieurs locuteurs, la reconnaissance multilocuteur peut être réalisée avec des résultats presqu'aussi bons qu'en monolocuteur, même si la difficulté de la tâche peut sembler plus grande.

La remarque introductive sur l'utilisation d'expertise humaine par rapport à l'auto-organisation peut signifier qu'il est possible de créer des systèmes qui offrent de bonnes performances sans pour autant être capable de comprendre en détail la façon dont ils fonctionnent, tout comme les humains sont capables d'utiliser leurs moyens de perception, d'action et de raisonnement sans comprendre la manière dont ils fonctionnent (si c'était le cas, l'approche à base de connaissances deviendrait triviale). Cependant, dans la mesure où le système est basé sur un modèle, l'étude fine des paramètres du modèle après qu'il ait été appris sur la base de données, peut aider la compréhension de ce que sont les structures d'organisation sous-jacentes.

Manuscrit reçu le 25 avril 1990.

Bibliographie

Il est certain que cet article ne peut présenter tous les travaux intéressants qui ont été réalisés ces dernières années. J'ai essayé d'utiliser les références les plus synthétiques sur un sujet, et j'ai plutôt tenu compte des articles où des expériences de test ont été conduites sur des données de taille raisonnable. Les publications en langue anglaise, et en particulier celles des IEEE, ont été préférées de par leur audience internationale. Beaucoup de références proviennent du LIMSI, car il était plus simple pour moi d'obtenir l'information venant de collègues proches, dans les cas où un travail similaire a également été accompli dans un autre laboratoire. Je voudrais présenter à l'avance mes excuses pour toutes les lacunes que l'on pourra y trouver, et pour toutes les erreurs qui sont inévitables dans ce genre de panorama.

[1] M. Abe, S. Nakamura, K. Shikano, H. Kuwabara, *Voice Conversion through Vector Quantization*, IEEE ICASSP'88, pp. 665-658, New York, 1988.

[2] M. Adda-Decker, J. Mariani, *Collaboration between Analytical and Global Methods for Continuous Speech Recognition*, 7th FASE Symposium on Acoustics and Speech, SPEECH'88, Vol. 1, pp. 345-352, Edinburgh, 1988.

[3] A. Andreewski, J. P. Binquet, F. Debili, C. Fluhr, Y. Hlal, J. S. Liénard, J. Mariani, B. Pouderoux, *Les dictionnaires en forme complète et leur utilisation dans la transformation lexicale et syntaxique de chaînes phonétiques correctes*, 10e JEP du GALF, Grenoble, May 1979.

[4] J. K. Baker, *The DRAGON system, An overview*, IEEE Transactions on ASSP, Vol. 23, No. 1, pp. 24-29, February 1975.

[5] L. R. Bahl, R. Bakis, P. S. Cohen, A. G. Cole, F. Jelinek, B. L. Lewis, R. L. Mercer, *Further Results on a Continuously Read Natural Corpus*, IEEE ICASSP'80, pp. 872-875, Denver, April 1980.

[6] L. R. Bahl, F. Jelinek, R. L. Mercer, *A Maximum Likelihood approach to continuous speech recognition*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 5, No. 2, pp. 179-190, March 1983.

[7] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, *Speech Recognition with Continuous-Parameter Hidden Markov Models*, Computer Speech and Language, Vol. 2, No. 3-4, September/December 1987.

[8] L. R. Bahl, R. Bakis, P. de Souza, R. L. Mercer, *Obtaining Candidate Words by Polling in a Large Vocabulary Speech Recognition System*, IEEE ICASSP'88, pp. 489-492, New York, 1988.

[9] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, *A New Algorithm for the Estimation of Hidden Markov Model Parameters*, IEEE ICASSP'88, pp. 493-496, New York, April 1988.

[10] L. R. Bahl, P. F. Brown, P. V. de Souza, R. L. Mercer, M. A. Picheny, *Acoustic Markov Models used in the Tangora Speech Recognition System*, IEEE ICASSP'88, pp. 497-500, New York, April 1988.

[11] L. R. Bahl, P. F. Brown, P. V. de Souza, P. S. Gopalakrishnan, F. Jelinek, R. L. Mercer, *Large Vocabulary Natural Language Continuous Speech Recognition*, IEEE ICASSP'88, Glasgow, May 1989.

[12] L. E. Baum, *An Inequality and Associated Maximization Technique in Statistical Estimation of Probabilistic Functions of Markov Processes*, Inequalities, Vol. 3, pp. 1-8, 1972.

[13] *Speech and Natural Language Work at BBN*, DARPA Review Meeting, Pittsburgh, June 1988.

[14] R. E. Bellman, *Dynamic Programming*, Princeton Univ. Press, 1957.

[15] D. Béroule, *Guided Propagation inside a Topographic Memory*, 1st International Conference on Neural Networks, IEEE San Diego, June 1987.

[16] R. Billi, G. Massia, F. Nesti, *Word Preselection for Large Vocabulary Speech Recognition*, IEEE ICASSP'86, pp. 65-68, Tokyo, 1986.

[17] R. Bisiani, T. Anantharaman, L. Butcher, *BEAM : An Accelerator for Speech Recognition*, IEEE ICASSP'89, Glasgow, 1989.

[18] J. R. Boddie, W. P. Hays, J. Tow, *The Architecture, Instruction Set and Development Support for the WE DSP32 Digital Signal Processor*, IEEE ICASSP'86, Tokyo, April 1986.

[19] B. P. Bogert, M. J. R. Healy, J. W. Tukey, *The Quefrency Alanysis of Time Series for Echoes : Cepstrum, Pseudo-autocovariance, Cross-Cepstrum and Saphe Cracking*, Proc. Symposium Time Series Analysis, M. Rosenblatt Ed., John Wiley and Sons, New York, pp. 209-243, 1963.

[20] S. Boll, J. Porter, L. Bahler, *Robust Syntax Free Speech Recognition*, IEEE ICASSP'88, pp. 179-182, New York, 1988.

[21] H. Bonneau, J. L. Gauvain, *Vector Quantization for Speaker Adaptation*, IEEE ICASSP'87, pp. 1434-1437, Dallas, 1987.

[22] H. Bourlard, C. J. Wellekens, *Speech Pattern Discrimination and Multilayer Perceptrons,* Research Report M 211, Philips Research Laboratory, Brussels, September 1987.

[23] H. Bourlard, C. J. Wellekens, *Links between Markov Models and Multi-Layer Perceptrons,* Research Report M 263, Philips Research Laboratory, Brussels, September 1988.

[24] L. Boves, M. Refice *et al., The Linguistic Processor in a Multi-Lingual Text-to-Speech and Speech-to-Text Conversion System,* European Conference on Speech Technology, pp. 385-388, Edinburgh, 1987.

[25] J. S. Bridle, M. D. Brown, R. M. Chamberlain, *An Algorithm for Speech Recognition,* ICASSP'82, pp. 899-902, Paris, May 1982.

[26] D. Burton, *Applying Matrix Quantization to Isolated Word Recognition,* IEEE ICASSP'85, pp. 29-32, Tampa, 1985.

[27] D. K. Burton, J. E. Shore, J. T. Buck, *Isolated-Word Speech Recognition Using Multisection Vector Quantization Codebooks,* IEEE Trans. on ASSP, Vol. 33, No. 4, August 1985.

[28] M. A. Bush, G. E. Kopec, *Network-Based Connected Digit Recognition,* Trans. on ASSP, Vol. 35, No. 10, October 1987.

[29] D. Chalmers, *A Transputer-Based Speech Recognition System,* IEEE ICASSP'89, Glasgow, 1989.

[30] Y. L. Chow, R. M. Schwartz, S. Roucosm, O. A. Kimball, P. Price, G. F. Kubala, M. O. Dunham, M. A. Krasner, J. Makhoul, *The Role of Word-Dependent Coarticulatory Effects in a Phoneme-Based Speech Recognition System,* IEEE ICASSP'86, Tokyo, April 1986.

[31] Y. L. Chow, M. O. Dunham, O. A. Kimball, M. A. Krasner, G. F. Kubala, J. Makhoul, P. J. Price, S. Roucos, R. M. Schwartz, *BYBLOS : the BBN Continuous Speech Recognition System,* IEEE ICASSP'87, pp. 89-92, Dallas, 1987.

[32] R. A. Cole, A. I. Rudnicky, V. Zue, D. R. Reddy, *Speech as Patterns on Paper,* in R. A. Cole Ed., « Perception and Production of Fluent Speech », Lawrence Elbaum Associates, Hillsdale, NJ, 1980.

[33] A. M. Colla, *Automatic Diphone Bootsrapping for Speaker-Adaptive Continuous Speech Recognition,* IEEE ICASSP'84, pp. 35, 2.1-35, 2.4, San Diego, 1984.

[34] J. W. Cooley, J. W. Tukey, *An algorithm for the Machine Calculation of the Compex Fourier Series,* Math. Computation, Vol. 19, pp. 297-301, 1965.

[35] M. Cravero, R. Pieraccini, F. Raineri, *Definition and Evaluation of Phonetic Units for Speech Recognition by Hidden Markov Models,* IEEE ICASSP'86, pp. 2235-2236, Tokyo, 1986.

[36] R. de Mori, L. Lam, *Plan Refinement in a Knowledge-Based System for Automatic Speech Recognition,* IEEE ICASSP'86, pp. 1217-1220, Tokyo, April 1986.

[37] A. M. Derouault, B. Mérialdo, *Natural Language Modeling for phoneme-to-text transcriptions,* IEEE Trans. on PAMI, Vol. 5, No. 2, pp. 742-749, 1986.

[38] A. M. Derouault, *Contexte-Dependent Phonetic Markov Models for Large Vocabulary Speech Recognition,* IEEE ICASSP'87, pp. 360-363, Dallas, April 1987.

[39] P. Dumouchel, V. Gupta, M. Lennig, P. Mermelstein, *Three Probabilistic Language Models for a Large-Vocabulary Speech Recognizer,* IEEE ICASSP'88, pp. 513-516, 1988.

[40] D. Dutoit, *Évaluation of Speaker-Independent Isolated-Word Recognition Systems over Telephone Network,* European Conference on Speech Technology, pp. 241-244, Edinburgh, September, 1987.

[41] J. L. Elman, J. L. McClelland, *An Architecture for Parallel Processing in Speech Recognition : the TRACE Model,* Biblthca phonet., No. 12, pp. 6-35 (Karger, Basel 1985).

[42] J. L. Elman, D. Zipser, *Learning the Hidden Structure of Speech,* UCSD Techn. Report, CS-8701, February 1987.

[43] Y. Ephraim, L. R. Rabiner, *On the Relations between Modeling Approaches for Information Sources,* IEEE ICASSP'88, pp. 24-27, New York, April 1988.

[44] M. W. Feng, F. Kubala, R. Schwartz, J. Makhoul, *Improved Speaker Adaptation using Text Dependent Spectral Mappings,* IEEE ICASSP'88, pp. 131-134, New York, 1988.

[45] J. D. Ferguson, *Variable Duration Models for Speech,* Symposium on Applications of Hidden Markov Models to Text and Speech, pp. 143-179, Princeton, 1980.

[46] L. Fissore, G. Micca, R. Pieraccini, *Strategies for Lexical Access to Very Large Vocabularies,* Speech Communication, Vol. 7, No. 4, December 1988.

[47] M. A. Franzini, M .J. Witbrock, K. F. Lee, *A Connectionist Approach to Continuous Speech Recognition,* IEEE ICASSP'89, pp.425-428, Glasgow, May 1989.

[48] J. L. Gauvain, J. Mariani, J. S. Liénard, *On the Use of Time Compression for Word-Based Recognition,* IEEE ICASSP'83, pp. 1029-1032, Boston, 1983.

[49] J. L. Gauvain, J. Mariani, *Evaluation of Time Compression for Connected-Word Recognition,* IEEE ICASSP'84, pp. 35-12, San Diego, 1984.

[50] J. L. Gauvain, *A Syllable Based Isolated Word Recognition Experiment,* IEEE ICASSP'86, pp. 57-60, Tokyo, 1986.

[51] A. Gersho, V. Cuperman, *Vector Quantization : A Pattern Matching Technique for Speech Coding,* IEEE Communications Magazine, December 1983.

[52] O. Ghitza, *Auditory Neural Feedback as a Basis for Speech Processing,* IEEE ICASSP'88, pp. 91-94, New York, 1988.

[53] S. Glinski, T. Mariano, D. Cassiday, T. Koh, C. Gerveshi, G. Wilson, J. Kumar, *The Graph Search Machine (GSM) : A Programmable Processor for Connected Word Speech Recognition and Other Applications,* IEEE ICASSP'87, pp. 519-522, Dallas, 1987.

[54] A. L. Gorin, B. R. Roe, *Parallel Level-Building on a Tree Machine,* IEEE ICASSP'88, pp. 295-298, New York, 1988.

[55] R. M. Gray, *Vector Quantization,* IEEE ASSP Magazine, Vol. 1, No. 2, April 1984.

[56] Y. F. Gong, J. P. Haton, *A Specialist Society for Continuous Speech Understanding,* IEEE ICASSP'88, pp. 627-630, New York, 1988.

[57] V. N. Gupta, M. Lennig, P. Mermelstein, *Integration of Acoustic Information in a Large Vocabulary Word Recognizer,* IEEE ICASSP'87, pp. 697-700, Dallas, 1987.

[58] F. Guyot, F. Alexandre, J. P. Haton, *Towards a Continuous Model of the Cortical Column : Application to Speech Recognition,* IEEE ICASSP'89, pp. 37-40, Glasgow, May 1989.

[59] J. P. Haton, N. Carbonnel, D. Fohr, J. F. Mari, A. Kriouille, *Interaction between stochastic modeling and knowledge-based techniques in acoustic-phonetic decoding of speech,* IEEE ICASSP'87, pp. 868-871, Dallas, 1987.

[60] A. L. Higgins, *Speaker Recognition by Template Matching,* Speech Tech'86 Conference, pp. 273-276, New York, April 1986.

[61] W. Huang, R. Lippmann, *A Neural Net Approach for Speech Recognition,* IEEE ICASSP'88, pp. 99-102, New York, April 1988.

[62] M. J. Hunt, M. Lennig, P. Mermelstein, *Experiments in Syllabe-Based Recognition of Continuous Speech,* IEEE ICASSP'80, pp. 880-883, Denver 1980.

[63] M. J. Hunt, C. Lefèbvre, *Speaker Dependent and Independent Speech Recognition Experiments with an Auditory Model,* IEEE ICASSP'88, pp. 91-94, New York, 1988.

[64] M. Y. Hwang, H. W. Hon, K. F. Lee, *Modeling between-word Coarticulation in Continuous Speech Recognition,* Proceedings Eurospeech'89, Vol. 1, pp. 5-8, Paris, September 1989.

[65] H. Ishizuka *et al., A Microprocessor for Speech Recognition,* IEEE ICASSP'83, pp. 503-506, Boston, 1983.

[66] F. Jelinek, *Continuous Speech Recognition by statistical methods,* IEEE Proceedings, Vol. 64, No. 4, pp. 532-556, April 1976.

[67] F. Jelinek, R. Mercer, *Interpolated Estimation of Markov Source Parameters from Sparse Data,* in *Pattern Recognition in Practice,* North-Holland, 1980.

[68] F. Jelinek *et al., Experiments with the Tangora 20 000 word speech recognizer,* IEEE ICASSP'87, pp. 701-704, Dallas, 1987.

[69] D. Jouvet, J. Monné, D. Dubois, *A New Network-Based Speaker-Independent Connected-Word Recognition System,* IEEE ICASSP'86, pp. 1109-1112, Tokyo, 1986.

[70] B. H. Juang, L. R. Rabiner, *Mixture Autoregressive HMM for Speech Signals,* IEEE Trans. on ASSP, Vol. 33, pp. 1404-1413, 1985.

[71] R. Kavaler *et al., A Dynamic Time Warp IC for a One Thousand Word Recognition System,* IEEE ICASSP'84, San Diego, March 1984.

[72] O. Kimball, P. Price, S. Roucos *et al., Recognition Performance and Grammar Constraints,* Proceedings of the DARPA Recognition Workshop, pp. 53-59, February 1986.

[73] S. Kirkpatrick, C. D. Gelatt, M. P. Vacchi, *Optimization by Simulated Annealing,* Science, 220, pp. 671-680, 1983.

[74] D. H. Klatt, *Overview of the ARPA-Speech Understanding Project,* in W. A. Lea Editor, *New Trends in Speech Recognition,* pp. 249-271, Prentice-Hall, 1980.

[75] T. Kohonnen, K. Mäkisara, T. Saramäki, *Phonotopic Maps - Insightful Representation of Phonological Features for Speech Recognition,* Proc. 7th ICPR, Montreal, pp. 182-185, July 30-August 2, 1984.

[76] T. Kohonen, *Self-Organization and Associative Memory,* Springer, 1984.

[77] T. Kohonen, K. Torkkola, M. Shozakai, J. Kangas, O. Ventä, *Microprocessor implementation of a large vocabulary speech recognizer and phonetic typewriter for Finnish and Japanese,* European Conference on Speech Technology, pp. 377-380, Edinburgh, September 1987.

[78] G. E. Kopec, M. A. Bush, *Network-Based Isolated Digit Recognition using Vector Quantization,* Trans. on ASSP, Vol. 33, No. 4, August 1985.

[79] M. Kuhn, H. Ney, H. Tomaschewski, *Fast Non Linear Time Alignment for Isolated Word Recognition,* IEEE ICASSP'81, pp. 736-740, Atlanta, 1981.

[80] W. A. Lea, Ed., *Trends in Speech Recognition,* Prentice-Hall, 1980.

[81] W. A. Lea, J. E. Shoup, *Specific Contributions of the ARPA-SUR project,* in W. A. Lea Editor, *New Trends in Speech Recognition,* pp. 382-421, Prentice-Hall, 1980.

[82] J. Lebœuf, D. Béroule, *Processing of Noisy Patterns with a Connectionnist System using a Topographic Representation of Speech,* European Conference on Speech Technology, pp. 191-194, Edinburgh, September 1987.

[83] Y. Le Cun, *Une procédure d'apprentissage pour réseaux à seuil asymétrique,* Proceedings Cognitiva'85, pp. 599-604, Paris, 1985.

[84] K.F. Lee, *Incremental Network Generation in Word Recognition,* IEEE ICASSP'86, pp. 77-80, Tokyo, 1986.

[85] K. F. Lee, H. W. Hon, *Speaker-Independent Phone Recognition Using Hidden Markov Models,* CMU Research Report CS-88-121, March 1988.

[86] K. F. Lee, H. W. Hon, *Large Vocabulary Speaker-Independent Continuous Speech Recognition,* IEEE ICASSP'88, pp. 24-27, New York, April 1988.

[87] K. F. Lee, *Large-Vocabulary Speaker-Independent Continuous Speech Recognition : the SPHINX system,* CMU report, CMU-CS-88-148, April 1988.

[88] C. H. Lee, F. K. Soong, B. H. Juang, *A Segment Model Based Approach to Speech Recognition,* IEEE ICASSP'88, pp. 501-504, New York, 1988.

[89] J. LeRoux, J. P. Arrou-Vignod, M. Invernizzi, *Correlation Data Classification for Low Bit Rate Speech Transmission and Real Time Word Recognition,* Eusipco Conference, 1980.

[90] S. E. Levinson, L. R. Rabiner, M. M. Sondhi, *An Introduction of the Application of the Theory of Probabilistic Functions on a Markov Process to Automatic Speech Recognition,* The Bell System Technical Journal, Vol. 62, No. 4, April 1983.

[91] S. E. Levinson, *Structural Methods in Automatic Speech Recognition,* Proceedings of the IEEE, Vol. 73, No. 11, pp. 1625-1650, November 1985.

[92] S. E. Levinson, *Continuously Variable Duration Hidden Markov Models for Automatic Speech Recognition,* Computer Speech and Language, Vol. 1, pp. 29-45, 1986.

[93] J. S. Liénard, J. Mariani, G. Renard, *Intelligibilité de phrases synthétiques altérées. Application à la transmission phonétique de la parole,* 9th ICA, Madrid, 1977.

[94] B. Lindblom, *On the Teleological Nature of Speech Processes,* Speech Communication, Vol. 2, No. 2-3, July 1983.

[95] Y. Linde, A. Buzo, R. M. Gray, *An Algorithm for Vector Quantizer Design,* IEEE Trans. on Communications, COM-28, pp. 84-95, January 1980.

[96] R. P. Lippman, *An Introduction to Computing with Neural Nets,* IEEE ASSP, Vol. 4, No. 2, pp. 4-22, April 1987.

[97] R. P. Lippmann, B. Gold, *Neural Classifiers Useful for Speech Recognition,* 1st International Conference on Neural Networks, IEEE, June 1987.

[98] R. P. Lippman, *Neural Nets for Computing,* IEEE ICASSP'88, oo. 1-6, New York, April 1988.

[99] E. P. Loeb, R. F. Lyon, *Experiments in Isolated Digit Recognition with a Cochlear Model,* IEEE ICASSP'87, pp. 1131-1134, Dallas, 1987.

[100] J. L. McClelland, J. L. Elman, *Interative Processes in Speech Perception : the TRACE model* in *Parallel Distributed Processing : Exploration in the Microstructure of Cognition, Vol. 1 : Foundation,* D. E. Rumelhart and J. L. McClelland Eds, MIT Press, 1986.

[101] J. Makhoul, *Vector Quantization in Speech Coding,* Proceedings of the IEEE, Vol. 73, No. 11, November 1985.

[102] J. Mariani, *Reconnaissance de la parole continue par diphonèmes,* in *Processus d'Encodage et de Décodage Phonétiques,* Symposium Galf-Greco, Toulouse, 1981.

[103] J. Mariani, *Hamlet : A Prototype of a Voice-Activated Typewriter,* European Conference on Speech Technology, Edinburgh, pp. 222-225, September 1987.

[104] J. Mariani, *Speech Technology in Europe,* European Conference on Speech Technology, Edinburgh, September 1987.

[105] J. Mariani, *Study of the IEEE ICASSP Conference,* Notes LIMSI 90-8, septembre 1990.

[106] J. D. Markel, A. H. Gray, *Linear Prediction of Speech,* Springer Verlag, 1976.

[107] K. Matrouf, F. Néel, J. L. Gauvain, J. Mariani, *Adaptive-Syntax Representation in an oral task-oriented dialogue for air-traffic controller training,* Eurospeech'89, Paris, September 1989.

[108] B. Mérialdo, *Speech Recognition with Very Large Size Dictionary,* IEEE ICASSP'87, pp. 364-367, Dallas, April 1987.

[109] B. Mérialdo, *Phonetic Recognition using Hidden Markov Models and Maximum Mutual Information Training,* IEEE ICASSP'88, pp. 11-114, New York, 1988.

[110] M. Minsky, S. Papert, *Perceptrons : An Introduction to Computational Geometry*, MIT Press, 1969.

[111] H. Murveit, M. Weintraub, *1 000-Word Speaker Independent Continuous-Speech Recognition using Hidden Markov Models*, IEEE ICASSP'88, pp. 115-118, New York, 1988.

[112] H. Murveit, *Design for a Real-Time Large-Vocabulary Continuous-Speech Recognition System*, DARPA Review Meeting, Pittsburgh, June 1988.

[113] C. S. Myers, L. R. Rabiner, *A Level Building Dynamic Time Warping Algorithm for Connected Word Recognition*, Trans. ASSP, Vol. 29, No. 2, pp. 284-297, April 1981.

[114] H. Na, S. Glinski, *Neural Net Based Pattern Recognition on the Graph Search Machine*, IEEE ICASSP'88, pp. 2168-2171, New York, 1988.

[115] A. Nadas, *On Turing's Formula for Word Probabilities*, IEEE Trans. on ASSP, Vol. 33, No. 6, December 1985.

[116] A. Nadas, D. Nahamoo, M. A. Picheny, *On a Model-Robust Training Method for Speech Recognition*, Trans. ASSP, Vol. 36, No. 9, pp.1432-1436, September 1988.

[117] S. Y. Nakajima, H. Hamada, *Automatic Generation of Synthesis Units Based on Contexte Oriented Clustering*, IEEE ICASSP'88, pp. 659-662, New York, 1988.

[118] M. Nakamura, K. Shikano, *A Study of English Word Category Prediction Based on Neural Networks*, ATR Technical Report, TR-I-0052, November 1988.

[119] A. Newell *et al., Speech Understanding Systems : Final Report of a Study Group*, North-Holland, 1973.

[120] H. Ney, *The Use of a One-Stage Dynamic Programming Algorithm for Connected Word Recognition*, Trans. on ASSP, Vol. 32, No. 2, pp. 263-271, April 1984.

[121] H. Ney, A. Noll, *Phoneme Modeling Using Continuous Mixture Densities*, IEEE ICASSP'88, pp. 437-440, New York, April 1988.

[122] A. V. Oppenheim, R. W. Schafer, *Digital Signal Processing*, Prentice-Hall, 1975.

[123] D. S. Pallett, *Selected Test Material, Test and Scoring Procedures for the May-June 1988 DARPA Benchmark Tests*, DARPA Review Meeting, Pittsburgh, June 1988.

[124] D. B. Parker, *Learning Logic*, Techn. Report, TR-47, ccrems, MIT, April 1985.

[125] D. B. Paul, R. P. Lippmann, R. P. Chen, C. Weinstein, *Robust HMM-Based Techniques for Recognition of Speech Produced under Stress and in Noise*, Speech'Tech 1986, New York, April 1986.

[126] D. B. Paul, E. A. Martin, *Speaker Stress-Resistant Continuous Speech Recognition*, IEEE ICASSP'88, pp. 283-286, New York, 1988.

[127] D. Paul, *MIT Lincoln Lab Presentation at the DARPA Strategic Computing Speech Recognition Program Meeting*, Pittsburgh, June 1988.

[128] S. M. Peeling, R. K. Moore, *Isolated Digit Recognition Experiments Using the Multi-Layer Perceptron*, Speech Communication, Vol. 7, No. 4, pp. 403-410, December 1988.

[129] A. B. Poritz, *Linear Prediction HMM and the Speech Signal*, IEEE ICASSP'82, pp. 1291-1296, Paris, 1982.

[130] A. B. Poritz, *Hidden Markov Models : A Guided Tour*, IEEE ICASSP'88, pp. 7-13, New York, 1988.

[131] R. W. Prager, T. D. Harrison, F. Fallside, *Boltzmann Machines for Speech Recognition*, Computer Speech and Language, Vol. 1, No. 1, March 1986.

[132] R. W. Prager, F. Fallside, *A Comparison of the Boltzmann Machine and the Back Propagation Network as Recognizers of Static Speech Patterns*, Computer Speech and Language, Vol. 2, Nos. 3-4, September/December 1987.

[133] G. Quénot, J. L. Gauvain, J. J. Gangolf, J. Mariani, *A Dynamic Time Warp Processor for Continuous Speech Recognition*, IEEE ICASSP'86, pp. 1549-1552, Tokyo, 1986.

[134] L. R. Rabiner, R. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, 1979.

[135] L. R. Rabiner, S. C. Levinson, A. E. Rosenberg, J. C. Wilpon, *Speaker Independent Recognition of Isolated Word Using Clustering Techniques*, IEEE Trans. on ASSp, Vol. 27, August 1979.

[136] L. R. Rabiner, J. G. Wilpon, *A Simplified Robust Training Procedure for Speaker Trained Isolated Word Recognition*, JASA, Vol. 68, No. 5, pp. 1271-1276, November 1980.

[137] L. R. Rabiner, A. Bergh, J. G. Wilpon, *An Embedded Word Training Procedure of Connected Digit Recognition*, IEEE ICASSP'82, pp. 1161-1624, Paris, May 1982.

[138] L. R. Rabiner, S. E. Levinson, *A Speaker-Independent Syntax-Directed Connected Word Recognition System based on Hidden Markov Model and Level-Building*, IEEE Trans. on ASSP., Vol. 33, No. 3, pp. 561-573, June 1985.

[139] L. R. Rabiner, B. H. Juang, S. E. Levinson, M. M. Sondhi, *Recognition of Isolated Digits using Hidden Markov Models with Continuous Mixture Densities*, AT&T Techn. Journal, Vol. 4, No. 6, pp. 1211-1233, July 1985.

[140] L. R. Rabiner, B. H. Juang, *An Introduction to Hidden Markov Models*, IEEE ASSP Magazine, Vol. 3, No. 1, pp. 4-16, January 1986.

[141] L. R. Rabiner, J. G. Wilpon, F. K. Soong, *High Performance Connected digit Recognition using Hidden Markov Models*, IEEE ICASSP'88, New York, April 1988.

[142] A. G. Richter, *Modeling of Continuous Speech Observations*, Proceedings « Advances in Speech Processing » Conference, IBM Europe Institue, 1986.

[143] J. R. Rohlicek, Y. L. Chow, S. Roucos, *Statistical Language Modeling using a Small Corpus from an Application Domain*, IEEE ICASSP'88, pp. 267-270, New York, 1988.

[144] A. E. Rosenberg, L. R. Rabiner, S. E. Levinson, J. G. Wilpon, *A Preliminary Study on the Use of Demi-Syllables in Automatic Speech Recognition*, IEEE ICASSP'81, pp. 967-970, Florida, 1981.

[145] R. Rosenblatt, *Principles of Neurodynamics*, Spartan Books, New York, 1959.

[146] B. Roth, *Real Time Implementations on a PC*, DARPA project review, Pittsburgh, June 1988.

[147] S. Roucos, R. Schwartz, J. Makhoul, *Segment Quantization for Very Low Rate Speech Coding*, IEEE ICASSP'82, pp. 1565-1568, Paris, 1982.

[148] D. E. Rumelhart, G. E. Hinton, R. J. Williams, *Learning Internal Representations by Error Propagation*, in *Parallel Distributed Processing : Exploration in the Microstructure of Cognition, Vol. 1 : Foundation*, D. E. Rumelhart and J. L. McClelland Eds, MIT Press, 1986.

[149] G. Ruske, T. Schotola, *The Efficiency of Demi-Syllabe segmentation in the Recognition of Spoken Words*, IEEE ICASSP'81, pp. 971-974, Florida, 1981.

[150] M. J. Russell, R. K. Moore, *Explicit Modeling of State Occupancy in Hidden Markov Models for Automatic Speech Recognition*, IEEE ICASSP'85, pp. 5-8, Tampa, March 1985.

[151] H. Sakoe, *Two-Level DP Matching — A Dynamic Programming Based Pattern Matching Algorithm for Connected Word Recognition*, IEEE Trans. on ASSP, Vol. 27, No. 6, pp. 588-595, Dec. 1979.

[152] H. Sakoe, R. Isotani, K. Yoshida, K. Iso, T. Watanabe, *Speaker-Independent Word Recognition using Dynamic Programming Neural Networks*, IEEE ICASSP'89, Glasgow, May 1989.

[153] C. SCAGLIOLA, *The Use of Diphones as Basic Units in Speech Recognition,* 4th FASE Symposium on Acoustics and Speech, Venise, April 1981.

[154] R. M. SCHWARTZ, J. KLOVSTADT, J. MAKHOUL, J. SORENSEN, *A Preliminary Study of a Phonetic Vocoder Based on a Diphone Model,* IEEE ICASSP'80, pp. 32-35, Denver, 1980.

[155] R. M. SCHWARTZ, Y. L. CHOW, S. ROUCOS, M. KRASNER, J. MAKHOUL, *Improved Hidden Markov Modeling for Acoustic-Phonetic Recognition of Continuous Speech,* IEEE ICASSP'84, San Diego, April 1984.

[156] R. SCHWARTZ, Y. CHOW, F. KUBALA, *Rapid Speaker Adaptation using a Probabilistic Spectral Mapping*, IEEE ICASSP'87, pp. 633-636, Dallas, 1987.

[157] D. SCHINKE, *Speaker Independent Recognition Applied to Telephone Access Information Systems,* Speech Tech'86, pp. 52-53, New York, April 1986.

[158] T. J. SEINOWSKI, C. R. ROSENBERG, *NETtalk : A Parallel Network that Learns to Read Aloud,* Technical Report, Johns Hopkins University, EECS-86/01, June 1986.

[159] G. SHICHMAN et al., *An IBM PC Based Large-Vocabulary Isolated Utterance Speech Recognizer,* IEEE ICASSP'86, pp. 53-56, Tokyo, 1986.

[160] K. SHIKANO, K. F. LEE, R. REDDY, *Speaker Adaptation through Vector Quantization,* IEEE ICASSP'86, pp. 2643-2646, Tokyo, 1986.

[161] J. E. SHORE, D. K. BURTON, *Discrete Utterance Speech Recognition without Time Alignment*, IEEE Trans. on Information Theory, Vol. 29, pp. 473-491, July 1983.

[162] M. R. SHROEDER, *Predictive Coding of Speech : Historical Review and Directions for Future Research*, IEEE ICASSP'86, pp. 3157-3164, Tokyo, April 1986.

[163] R. SIMAR Jr, *TMS320 : Texas Instrument's Family of Digital Signal Processor,* Proceedings Speech Tech'87, New York, April 1987.

[164] H. SINGER, J. L. GAUVAIN, *Connected Speech Recognition using Dissylable Segmentation,* Proceedings of the Acoustical Society of Japan Conference, October 1988.

[165] G. S. SLUTSKER, *Nelinejnyp Method Analiza Recevych Signalov,* Trudy Niir, No. 2, 1968.

[166] F. K. SOONG, A. E. ROSENBERG, *On the Use of Instantaneous and Transitional Spectral Information in Speaker Recognition,* IEEE ICASSP'86, pp. 877-880, Tokyo, 1986.

[167] P. E. STERN, M. ESKENAZI, D. MEMMI, *An expert system for speech spectrogram reading,* IEEE ICASSP'86, pp. 1193-1196, Tokyo, April 1986.

[168] N. SUGAMURA, K. SHIKANO, S. FURUI, *Isolated Word Recognition Using Phoneme-Like Templates,* IEEE ICASSP'83, pp. 723-726, Boston, 1983.

[169] J. THOMAS, *DSP56000, A High Performance DSP Architecture,* Proceedings Speech Tech'87, New York, April 1987.

[170] T. K. VINTSJUK, *Recognition of Words of Oral Speech by Dynamic Programming,* Kibernetika, Vol. 81, No. 8, 1968.

[171] A. J. VITERBI, *Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm,* IEEE Trans. on Information Theory, Vol. 13, No. 2, pp. 260-269, April 1967.

[172] A. WAIBEL, T. HANAZAWA, G. HINTON, K. SHIKANO, K. LANG, *Phoneme Recognition using Time-Delay Neural Networks,* ATR Research Report, TR-I-0006, October 1987.

[173] T. WATANABE, *Segmentation-free Syllable Recognition in Continuously Spoken Japanese,* IEEE ICASSP'83, pp. 320-323, Boston, 1983.

[174] P. WERBOS, *Beyond Regression : New Tools for Prediction and Analysis in the Behavioral Sciences,* PhD Thesis, Harvard, August 1974.

[175] A. P. WITKIN, *Scale-Space Filtering : A New Approach to Multi-Scale Description,* IEEE ICASSP'84, San Diego, April 1984.

[176] F. H. WU, K. GANESAN, *Comparative Study of Algorithms for VQ Design Using Conventional (LBG) and Neural-Net Based Approaches,* IEEE ICASSP'89, Glasgow, May 1989.

[177] S. R. YOUNG, W. H. WARD, *Towards Habitable Systems : Use of World Knowledge to Dynamically Constrain Speech Recognition,* Proceedings 2nd Symposium on Advanced Man-Machine Interface through Spoken Language, pp. 30-130-12, Hawaii, November 1988.

[178] S. J. YOUNG, N. H. RUSSELL, J. H. S. THORNTON, *Speech Recognition in VODIS II,* IEEE ICASSP'88, pp. 441-444, New York, April 1988.

[179] V. W. ZUE, L. F. LAMEL, *An Expert Spectrogram reader : a Knowledge-Based Approach to Speech Recognition,* IEEE ICASSP'86, pp. 1197-1200, Tokyo, April 1986.

[180] V. ZUE, *Recent Speech Recognition Results at MIT,* DARPA Review Meeting, Pittsburgh, June 1988.