RECONNAISSANCE DE LA PAROLE : les principes.

J. Caelen

1. INTRODUCTION

1.1. Le langage et la parole

L'acte de parole, lié à l'acte de langage, participe de l'intelligence et du raisonnement. De nombreuses études psychologiques fondées sur le développement simultané du langage et de l'intelligence chez l'enfant et sur la comparaison entre l'homme et le champanzé ont montré l'importance du langage dans le développement de l'intelligence [Viaud, 1971]. La notion d'intelligence, conçue comme la fonction spécifique de l'homme face aux animaux, a souvent été définie comme étant tout ce qui distingue l'homme de l'animal ou de la machine. Cependant cette notion est si délicate à manier que Minsky [Minsky, 1985] l'évacue des débats non-philosophiques dans lesquels doit évoluer l'Intelligence Artificielle. C'est dans ce contexte ambigü que prend place la reconnaissance et la compréhension de la parole continue : elle ressort des activités humaines les plus spécifiques et se prétend automatisable.

La linguistique et la parole (langage parlé) entretiennent des rapports étroits, c'est une évidence de le rappeler ici. Or la linguistique peut-être vue comme une branche de la psychologie qui étudie les comportements humains qui eux-mêmes dépendent de la structure de la langue. La philosophie du langage pose la pragmatique comme primat de la langue : parler est une forme d'action. Depuis qu'on étudie le comportement de l'être humain, deux démarches fondamentalement opposées ont alterné au cours du temps :

1. pour les uns le progrès des connaissances dépend de l'observation rigoureuse du comportement réel de l'homme : c'est l'approche empiriste fondée sur la notion d'acquis,
2. pour les autres, de telles observations ne sont intéressantes que dans la mesure où elles nous révèlent des lois sous-jacentes, lesquelles ne se manifestent dans le comportement que sous une forme partielle et altérée, c'est l'approche rationnaliste qui s'appuie sur la notion d'inné [Searle, 1972], [Searle, 1973].

Ces deux démarches illustrent bien le dilemne de la linguistique : être cognitive ou être computationnelle. C'est en quelque sorte un débat entre performance et compétence. Selon [Chomsky, 1969], [Chomsky, 1971] la grammaire complète d'une langue comporte trois composantes, pour décrire compétence et performance :

1. Une partie syntaxique qui engendre et décrit la structure interne des phrases d'une langue,
2. La phonologie qui rend compte de la structure phonique des phrases engendrées par la composante syntaxique,
3. La sémantique qui rend compte du sens.

La compétence d'un locuteur peut être représentée par des règles de réécriture permettant de différencier les phrases ambiguës. Depuis, Chomsky a lui-même révisé plusieurs fois sa théorie. Mais parmi toutes les "computations" présentées en linguistique, aucune n'a subi avec succès les épreuves psychologiques.

1.2. Les sources de connaissance dans les systèmes de compréhension automatique de la parole

Un système de compréhension automatique de la parole dispose des sources de connaissance suivantes :

1. Phonétiques.
2. Phonologiques.
3. Prosodiques.
4. Lexicales.
5. Syntaxiques.
6. Sémantiques.
7. Pragmatiques.

1.2.1. La phonétique

La phonétique est une science qui concerne l'étude des caractéristiques physiques des sons de la parole en liaison avec la langue. On peut analyser les sons sur trois plans complémentaires :

- perceptif.
- articulatoire.
- acoustique.

Le DAP (Décodage Acoustico-Phonétique) est un module qui utilise essentiellement les caractéristiques acoustiques pour obtenir, à partir du signal :

- un ensemble de segments, et un ensemble de traits acoustiques pour chaque segment.
- l'ensemble des transcriptions phonétiques de chaque segment.

1.2.2. La phonologie

La phonologie a comme objectif d'étudier les variantes phonétiques contextuelles. En reconnaissance de la parole, la phonologie regroupe l'ensemble des modules de traitement des altérations possibles d'un phonème ou d'un mot dans un contexte donné.

La phonologie regroupe trois types d'informations :

- les altérations phonologiques dans le mot, (variantes de prononciation),
- les altérations phonologiques dues aux flexions en fin de mot (conjugaisons des verbes, pluriels des noms et des adjectifs),
- les altérations qui apparaissent à la jonction de deux mots (liaison).

A ces trois types d'altérations on peut ajouter les altérations qui se produisent artificiellement à cause des erreurs du décodage acoustique phonétique :

- les délétions de phonèmes dans les cas de sous-segmentation du signal de la parole,
- les insertions de phonèmes dans les cas de sur-segmentation du signal de la parole,

- les substitutions de phonèmes dans les cas de mauvaises identifications.

Un module phonologique doit tenter dans la mesure du possible de compenser ces erreurs.

1.2.3. La prosodie

La prosodie peut être considérée comme une sorte de “ponctuation acoustique” de la parole. Elle recouvre les aspects liés à la hauteur de la voix, à l'intensité et à la durée des segments syllabiques. Son rôle dans la langue est multiple :

- un rôle démarcatif des mots, syntagmes, groupes prosodiques,
- un rôle accentuel des mots,
- un rôle illocutoire (force des actions),
- un rôle rhétorique.

Un module de traitement de la prosodie dans un système de compréhension est donc lié à tous les autres modules et son interaction avec eux est complexe.

1.2.4. Le lexique

Les performances d'un système de reconnaissance sont affectées par la taille du vocabulaire et aussi par le degré de confusion entre les mots. Le dictionnaire doit être étudié de telle sorte qu'il autorise économiquement la représentation de toutes les prononciations envisageables des mots, mais aussi pour qu'il permettre d'accéder directement à tous les mots contenant le même syllabe ou le même trait acoustique, de telle sorte qu'il soit possible de générer les hypothèses des mots à partir des caractéristiques du signal dans l'analyse ascendante. L'analyseur lexical doit être capable également de proposer les traits acoustiques pour les mots hypothèses développés afin d'être vérifiés sur le signal à un instant précis dans l'analyse descendante. Le lexique doit contenir des informations syntaxiques et sémantiques pour alimenter les niveaux linguistiques. Il joue donc un rôle pivot dans le système, à l’articulation entre les niveaux phonétiques (DAP, phonologie) et les niveaux linguistiques (syntaxe, sémantique).

1.2.5. La syntaxe

Du point de vue de la langue, la syntaxe est l'ensemble des règles contraignant l'ordre des mots dans la phrase. Chomsky [Chomsky, 1965], par sa théorie du langage et par sa grammaire générative, a beaucoup marqué la linguistique computationnelle des années 65. Beaucoup de formalismes utilisés depuis en sont dérivés.

Dans un système de compréhension, le but de la syntaxe est de réduire le nombre de phrases autorisées à partir du vocabulaire choisi. Par exemple, on peut construire 250^8 ($\approx 10^{19}$) phrases de 8 mots à partir d'un vocabulaire de 250 mots, mais seules 10^7 d'entre elles environ ont un sens. On a donc divisé l'espace de recherche par 10^{12}.

Les outils utilisés pour décrire explicitement la syntaxe et opérer l’analyse des phrases sont nombreux.

On peut citer par exemple les grammaires TNF (utilisant des réseaux and-or), les réseaux à états finis, les ATN (Augmented Transition Network) [Woods, 1970], les grammaires d'unification (GPSP, HGSP), les réseaux d'arbres adjoints (TAG, XTAG), les grammaires lexicales fonctionnelles (LFG), etc., autant de formalismes ou de variantes dérivés des grammaires génératives incluant des traits sémantiques.

Une autre grande voie est d'utiliser des processus markoviens [Baker, 1975] sans décrire explictement la grammaire : on opère des statistiques sur des séquences de mots (bi-grammes ou tri-grammes) sur de vastes corpus. Les matrices de transition entre catégories dee mots sont utilisés ensuite pendant la phase de reconnaissance pour filtrer les séquences improbables. On peut classer ce type de méthode parmi les grammaires stochastiques.

1.2.6. La sémantique

La sémantique est définie d'un point de vue linguistique, comme la relation entre la forme des signes linguistiques, ou "signifiants", et ce qui est signifié, ou "signifiés" [Brekle, 1974], [Le Ny, 1979]. On peut distinguer plusieurs types de sémantique : la sémantique descriptive, la sémantique générative, la sémantique interprétative, la sémantique différentielle, la sémantique logique, etc.

Les informations sémantiques des objets et les relations entre ces objet peuvent être codées dans des réseaux sémantiques. En reconnaissance de la parole la sémantique restreint la combinatoire syntaxique, elle peut avoir un statut autonome ou être immergée dans un grammaire syntaxico-démantique. Une sémantique de cas [Fillmore, 1969] est souvent utilisée par les informaticiens et formalisée dans des réseaux à objets (Sowa par exemple).

1.2.7. La pragmatique

La pragmatique peut être définie comme la relation entre signifiés et interprétations [Morris, 1938]. La pragmatique peut être définie aussi comme l'étude des aspects du langage qui font référence aux relations entre locuteur et interlocuteur, d'une part, et entre interlocuteurs et situation concrète, d'autre part. La pragmatique recouvre l'ensemble des relations entre le langage et le contexte d'énonciation.

En compréhension automatique du langage, la pragmatique permet de traiter les références et d'interpréter la situation de communication :

1. les déictiques ou ensemble des mots dont la référence fait partie de la situation de la communication, tels du locuteur (je, moi, nous,....) ou les lieux (ici, là.....).
2. les références anaphoriques, qui permettent de reprendre un terme ou une phrase antérieure (il l'attend depuis....).
3. les ellipses ou phrases incomplètes.

Dans les applications de communication homme-machine la pragmatique joue un rôle très important dans l'interaction entre l'homme et l'univers de l'application et pour résoudre les problèmes référentiels.

2. La reconnaissance et la compréhension automatiques de la parole

On peut distinguer entre deux grands types de systèmes :

a) les systèmes de reconnaissance de la parole qui ont pour objectif de décoder les phrases prononcées mot par mot voire phonème par phonème, sans comprendre le sens de la phrase (comprendre s'entend ici par "fournir une représentation sémantique de l'énoncé"),

b) les systèmes de compréhension de la parole qui ont pour objectif de comprendre la phrase prononcée même si la phrase d'entrée n'est pas reconnue précisément (il y a des erreurs au niveau du décodage acoustique phonétique) [Pierrel, 1981].

2. 1. Les différents méthodes de reconnaissance de la parole

Il y a deux méthodes en reconnaissance de la parole :

2.1.1. La méthode globale

Cette méthode considère le plus souvent le mot ou le phonème comme unité de reconnaissance minimale, c'est-à-dire indécomposable. Dans ce type de méthode on compare globalement le message d'entrée (mot, phrase) aux différentes références stockées dans un dictionnaire en utilisant des algorithmes de programmation dynamique ou des modèles de Markov (HMM = Hidden Markov Model). Cette méthode a pour avantage d'éviter l'explicitation des connaissances relatives aux transitions qui apparaissent entre les phonèmes.

La généralisation de la méthode à des unités enchaînés présente un certain intérêt car les unités phonétiques sont représentées par des modèles et les connaissances phonétiques, lexicales et syntaxiques sont compilées dans un seul réseau, ce qui rend le système de reconnaissance très homogène, des niveaux acoustiques jusqu'aux niveaux linguistiques. La reconnaissance consiste alors à trouver le meilleur chemin dans le réseau global pour reconnaître une phrase prononcée.

Ce type de méthode est utilisé dans les systèmes suivants :

- reconnaissance de mots isolés.
- reconnaissance d'unités enchaînées.
- reconnaissance de parole dictée avec pauses entre les mots.

2.1.2. La méthode analytique

Cette méthode fait intervenir un modèle phonétique du langage. Il y a plusieurs unités minimales pour la reconnaissance qui peuvent être choisies (syllabe, demi-syllabe, diphone, phonème, phone homogène, etc.). Le choix parmi ces unités dépend des performances des méthodes de segmentation utilisées. La reconnaissance dans cette méthode, passe par la segmentation du signal de la parole en unités de décision puis par l'identification de ces unités en utilisant des méthodes de reconnaissance des formes (classification statistique, réseau de neurones, etc.) ou des méthodes d'intelligence artificielle (systèmes experts par exemple). Cette méthode est beaucoup mieux adaptée pour les systèmes à grand vocabulaire et pour la parole continue. Les problèmes qui peuvent apparaître dans ce type de système sont dus en particulier aux erreurs de segmentation (délétions, insertions, substitutions, recouvrements) et d'étiquetage phonétique. C'est pourquoi le DAP (Décodage Acoustico-Phonétique) est fondamental dans une telle approche.

2.2. La reconnaissance de la parole aux niveaux « bas » ou décodage acoustico-phonétique

La reconnaissance et la production de séquences sont des problèmes quotidiens pour l'être humain : associer à un monde toujours changeant une représentation mentale de ce monde peuplé d'objets connus est une tâche relativement simple et inconsciente. Cette association passe pourtant par diverses étapes de traitement qui ont deux tâches principales : a) percevoir les objets et b) les nommer. Ces deux tâches sont difficilement séparables mais résument bien les enjeux de la reconnaissance des formes et de l'intelligence artificielle : un certain nombre de traitements perceptifs et/ou moteurs sont câblés qui permettent de caractériser les formes sensibles, ces formes alors dégrossies, séparées, peuvent être communiquées au traitement cognitif qui s'attachera à les classer et les nommer vis-à-vis d'un code appris.

Au début des années 70, la démarche implicite des fondateurs de l'intelligence artificielle était de porter le maximum d'efforts sur le raisonnement : en reconnaissance de parole notamment, il était d'usage de contraindre au maximum le décodage par les niveaux symboliques en se contentant d'une description acoustico-phonétique sommaire. Si ce pari sur le tout cognitif a permis de franchir des étapes importantes vers la "conquête" du monde sensible, les conclusions du projet DARPA [Klatt, 1977], constatant le relatif échec des projets de reconnaissance lancés aux Etats-Unis, mettent l'accent sur la nécessité de remonter aux traitements de haut-niveau (linguistique) des informations sûres. D'un autre côté et en guise de boutade, il est d'usage de citer cette phrase célèbre de Jelinek : "Each time we fire a phonetician, recognition rate increases 5%", qui prétendrait au contraire se passer des connaissances de haut-niveau pour le décodage acoustico-phonétique, en se fondant uniquement sur les données acoustiques. Ces deux positions extrêmes conduisent à l'heure actuelle à des attitudes plus mesurées chez les chercheurs qui adoptent des méthodes dans lesquelles les niveaux symbolique (connaissances structurelles) et sous-symbolique (pré-catégorisation des formes) coopèrent dans le processus de décodage après qu'une analyse morphologique ait fait clairement émerger les formes à discriminer : on retrouve donc actuellement un certain équilibre entre traitement du signal, reconnaissance des formes et intelligence artificielle, le tout étant souvent sous-tendu par des approches cognitives.

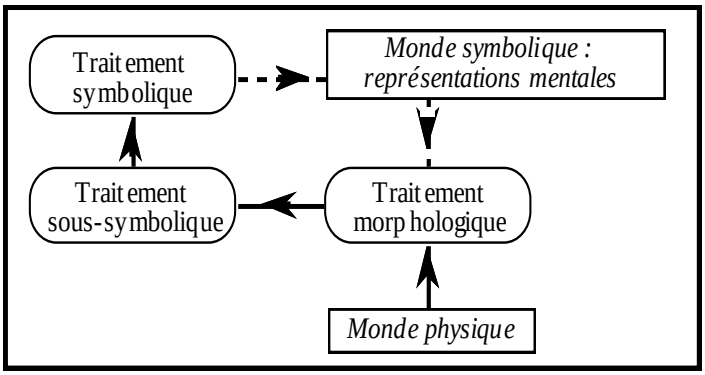

Figure 1: Les divers paradigmes de la reconnaissance.
- Traitement morphologique : description des traitements aboutissant au monde perçu. C'est dans cet espace de représentation que les représentations mentales sont construites (cf. illusions optiques).
- Traitement sous-symbolique : traitement de la classification catégorielle opérée à partir des représentations ; nous avons les possibilités innées de discriminer une importante quantité de formes auditives et visuelles ; nous constituons un lexique de formes par "oubli" des frontières (cf. "Naître humain", [Mehler et al., 1990]).
- Traitement symbolique : les structures cognitives plus complexes (compréhension) sont alors construites par le raisonnement cognitif.

2.2.1. Nature et organisation des sons de parole

Le signal de parole peut être considéré comme la concrétisation d'une suite de symboles abstraits (ou phonèmes) organisés selon un code linguistique et liés entre eux par des relations structurantes. La portée de ces relations sur l'axe syntagmatique définit les contextes interne, immédiat et lointain. Le contexte "interne" résulte de facteurs linguistiques, articulatoires, idiosyncrasiques, etc. et agit directement sur la réalisation individuelle du phonème ; le contexte "immédiat" se manifeste par la déformation des phonèmes adjacents sous l'effet de la coarticulation (qui provoque des migrations de traits sur l'axe syntagmatique) et des amalgames lexicaux ; le contexte "lointain" opère par le biais de la prosodie de groupe en agissant sur l'émergence des phonèmes (via les syllabes accentuées) dans le groupe (accent lexical, accent de groupe, accent de phrase, etc.). Ainsi le signal de parole résulte de la contribution de tous ces facteurs qui opèrent à des niveaux de structuration différents (Fig. 2).

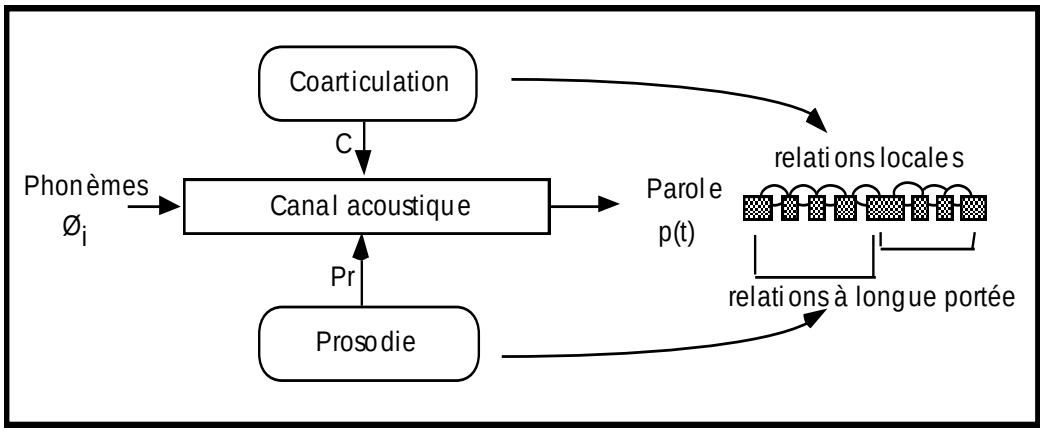

Figure 2: La parole résulte de contributions diverses tant au niveau structurel qu'au niveau physique : elle se présente comme

une séquences de formes organisées et structurées par des relations à courte et longue portée.

Plus formellement, le signal de parole p(t) produit à l'instant t par un locuteur, résulte d'un codage du triplet $P_i = (Ø, c, \pi)_i$ dans lequel,

$Ø_i$ représente le phonotype à réaliser à cet instant,

c_i les contraintes locales (coarticulation) et,

π_i les contraintes lointaines (prosodie).

Du fait de la redondance de la langue et de la capacité de l'auditeur à s'adapter au locuteur (ce qui autorise ce dernier à une marge d'imprécision plus grande dans la réalisation des phonèmes), le phonotype à réaliser n'est pas forcément une cible précise : il résulte d'un choix parmi les allophones pouvant convenir vis-à-vis du lexique et de la phonologie en position i ; cela signifie que $Ø_i \in \{ø_{ij}\}$ où les $ø_{ij}$ sont les J allophones possibles en position i. Vu par un observateur qui ignorerait la phonologie, le locuteur aurait donc une certaine probabilité de réaliser l'allophone $ø_{ij}$ pour coder le phonotype $Ø_i$.

Au niveau acoustique la substance phonique p(t) se présente comme un continuum et non comme une suite d'éléments discrets ; sur le plan acoustique, la parole présente une structure évolutive, non-stationnaire, dans laquelle il est possible de trouver des changements (vs. similarités ou stabilités ou régularités) permettant d'y localiser des éléments e_k — à la limite ces éléments peuvent être totalement arbitraires et être définis par exemple, par une fenêtre d'analyse (trame de signal) de longueur fixe et résulter d'un processus de quantification vectorielle. Ces éléments, véhiculant un code linguistique, sont donc structurés et organisés : on peut supposer qu'ils le sont à travers une "grammaire" G_p liée au triplet P_i. On obtient alors une représentation discrète du signal p(t) en posant : $p(t) = (\{e_k\}, G_p)$.

Le décodage acoustico-phonétique :

Le problème général du décodage acoustico-phonétique (DAP) est de localiser et d'identifier les phonotypes $Ø_i$ sous leurs diverses réalisations allophoniques $ø_{ij}$, dans le signal p(t). Après extraction d'éléments discrets e_k, une des phases du décodage revient à mettre en correspondance la suite $\{e_k\}$ et la suite $\{ø_{ij}\}$ en s'affranchissant des contraintes c_i et π_i considérées comme des bruits dans ce problème.

La grammaire G_p est une "grammaire acoustico-phonétique" dépendante du locuteur et non de l'allocutaire (auditeur). Dans la mesure où le langage et le processus de compréhension résultent d'une négociation entre le locuteur et l'allocutaire, il peut être utile d'introduire la compétence de ce dernier dans le processus de décodage. Le problème du DAP semble mieux formulé si l'on pose qu'il est un décodage des éléments e_k vis-à-vis d'une grammaire "apprise" G_a différente de G_p ; G_a serait mise en œuvre par un auditeur "idéal". Cela permet d'imaginer que tous les sons émis par le locuteur ne sont pas tous "audibles" par l'allocutaire ou peuvent référer à un système phonétique différent : les cibles qu'a voulu produire le locuteur, n'ont été réalisées que partiellement (avec une certaine probabilité et

une certaine netteté) et n'ont eu qu'un impact partiel sur l'allocutaire (fonction d'adéquation entre les cibles attendues comme telles et leurs réalisations). L'introduction de cette grammaire G_a met l'emphase sur les processus descendants pouvant permettre de s'affranchir de la variabilité inter-locuteur, mais ne permet pas de remonter des éléments e_k aux phonotypes $Ø_i$ directement : on ne peut que trouver au mieux des phonotypes $Ø_i$'.

Pour résumer, nous pouvons poser le problème du DAP comme une hiérarchie de trois sous-problèmes :

- extraire et décrire e_k (analyse morphologique),
- catégoriser e_k vis-à-vis de la grammaire "apprise" G_a en classes $Ø_i$' (analyse sous-symbolique),
- mettre en correspondance $Ø_i$' et $ø_{ij}$ (analyse symbolique).

Une autre variante de ce problème consiste à tenter d'inférer la grammaire G_p pour chaque locuteur en évitant ainsi d'introduire un biais dès le point 2) et en supprimant le processus de mise en correspondance du point 3). La décomposition en sous-problèmes devient alors :

- extraire et décrire e_k (analyse morphologique),
- inférer la grammaire G_p (par une méthode d'inversion),
- catégoriser e_k en classes $Ø_i$ vis-à-vis de la grammaire "adaptée" G_p (analyse sous-symbolique).

Nous allons maintenant examiner quelques solutions pour le DAP à la lumière de ce modèle général.

2.2.2. Approches pratiques pour le DAP

Pratiquement, la mise en place des étapes de résolution pour le décodage acoustico-phonétique automatique s'appuie sur les trois types d'analyse décrites dans le paragraphe précédent. La première de ces étapes, l'analyse morphologique, est d'extraire et de décrire ou de caractériser des éléments e_k servant de base aux constructions symboliques des étapes ultérieures. Il existe de nombreuses façons d'aborder ce problème : selon divers points de vue, les éléments phoniques e_k sont (a) des µ-segments ou (b) des segments chevauchants ou (c) des suites événements ou (d) des cibles acoustiques. Dans tous les cas ils se ramènent à des éléments discrets pouvant toujours être caractérisés par un vecteur de paramètres. Examinons plus en détail toutes ces solutions.

2.2.2.1. Le modèle "trame" et les chaînes de Markov

Dans ce modèle, il n'y a pas de segmentation en-dehors de celle qui est imposée de facto par le séquencement des trames obtenu au cours de l'analyse acoustique par une fenêtre glissante. Les éléments e_k sont donc des trames vectorisées. Dans la variante la plus simple, on suppose qu'une trame appartient à une et une seule unité phonétique $Ø_i$ (de ce fait il n'y a pas de recouvrement de phonèmes ni d'organisation complexe de la substance acoustique). A partir de là, la méthode utilisée en DAP est généralement fondée sur une grammaire pour modéliser l'enchainement des éléments e_k et

tient compte d'une relation d'appartenance « floue » $\square_f$ des éléments e_k dans l'ensemble des classes de phonotypes. Un des formalismes les plus achevés consiste à traiter le problème comme une chaîne de Markov cachée, à l'aide d'une matrice de transition entre états (grammaire G_a) et d'une matrice d'émission de symboles à partir de chaque état. Une deuxième grammaire markovienne peut-être ajoutée à un niveau linguistique et le DAP se résout par cheminement dans des réseaux locaux (modèles de phonèmes enchainés) enchâssés dans un réseau global (modèle de mots ou plus généralement modèle de langage). A travers ces grammaires probabilistes toute une série de phénomènes sont pris en considération de manière globale : les contraintes contextuelles locales et la relation $\square_f$. Par contre les contraintes prosodiques ne sont pas prises en compte par un tel modèle.

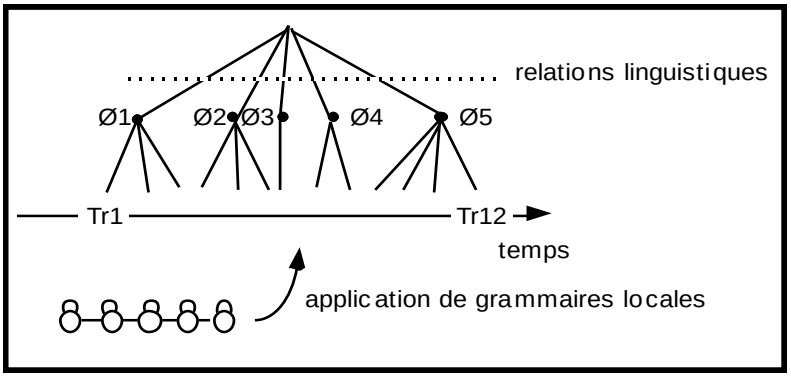

Figure 4 : Les trames Tr sont structurées par des grammaires stochastiques en chaînes de Markov. Les états "émettent" les symboles $Ø_i$.

Le phénomène physique observé est supposé être à tout instant dans l'un d'un ensemble de N états E = {ß1,ß2,....ßN}. A intervalles de temps réguliers (longueur d'une trame), le système entreprend un changement d'état (il peut rester aussi dans le même état) selon un ensemble de probabilités associées à l'état courant q_t. Une description complète de la source Markovienne exigerait, en général, la spécification de l'état courant (à l'instant t) ainsi que tous les états précédents. Cette description est cependant souvent tronquée à l'ordre 1 (processus du 1er ordre), c'est-à-dire que la probabilité d'être à l'instant t dans l'état $ß_i$ sachant l'ensemble des états parcourus précédemment vaut :

$$P[q_t=ß_i \mid q_{t-1}=ß_j, q_{t-2}=ß_k,...] = P[q_t=ß_i \mid q_{t-1}=ß_j]$$

Ces probabilités sont appelées probabilités de transition et sont souvent notées a_{ji} (transition de l'état j à l'état i) avec $A = [a_{ji}, i=1,N, j=1,N]$ matrice de transition dont la somme des coefficients sur une ligne est égale à l'unité.

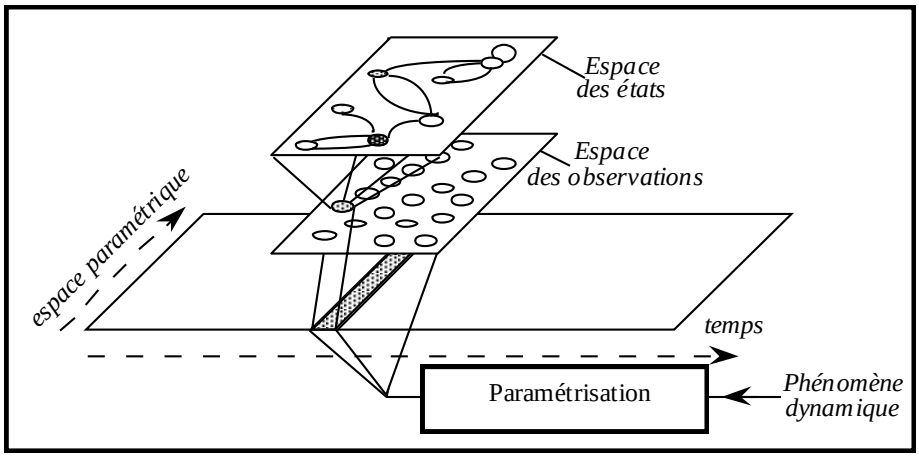

Figure 5 : Modélisation par chaîne de Markov cachée. Les contraintes structurelles sont données par l'espace des états.

Une chaîne de Markov est alors caractérisée par un triplet {E, A, I} où I représente l'état initial (les probabilités de présence de q_o sur chacun des états de E) et permet d'initialiser le processus d'identification qui consiste à calculer la probabilité que la séquence ait été produite par l'une des chaînes de Markov caractérisant le lexique. Dans les chaînes de Markov, chaque état est associé à un événement (physique) observable. En l'état, ce modèle effectue l'analyse morphologique (sur les trames) et l'analyse sous-symbolique (association à un ensemble d'états). C'est donc insuffisant pour l'analyse symbolique : une extension du concept des modèles markoviens permet de décrire l'émission probabiliste d'un symbole dans chaque état. Le modèle résultant appelé chaîne de Markov cachée (HMM = Hidden Markov Model en anglais) est un processus doublement stochastique dont un, sous-jacent, n'est pas directement observable (caché) mais peut être observé à travers un autre ensemble de processus stochastiques qui produisent les séquences d'observations (fig. 5).

Une chaîne de Markov cachée est donc un 5-uplet {E, A, I, S, B} où la chaîne de Markov est augmentée par un ensemble S d'observations $\{Ø_1, Ø_2, ...Ø_M\}$ et où une matrice B dite de probabilités d'émission exprime les probabilités b_{kj} que l'état $ß_j$ émette à l'instant t le symbole $Ø_k$:

$$b_{kj} = P[Ø_k \text{ à } t \mid q_t = ß_j]$$

On cherchera donc à exprimer par divers algorithmes de parcours de la chaîne d'états la probabilité que la chaîne de Markov cachée émette la séquence observée, sachant que chaque transition dans l'espace des états provoque l'émission d'un symbole dans l'espace S. La reconnaissance d'une séquence d'observations (calcul de probabilité de produire la séquence par une chaîne de Markov cachée) peut être effectuée de diverses manières :
(a) en utilisant une technique analogue à la programmation dynamique, appelée algorithme de Viterbi [Bridle, 1984], permettant de trouver la trace des états
(b) en utilisant le même algorithme qu'à l'apprentissage (Forward-Backward) {Baum, 1972].

L'aspect le plus intéressant des chaînes de Markov cachées concerne l'apprentissage qui peut être rendu automatique et évite l'étape difficile du recueil d'expertise des règles de la grammaire G_a et de la stratégie de décodage.

La structure du système est donnée a priori (E,S, topologie du réseau fixée en fixant certains poids a_{ji} à 0) et seuls les paramètres de cette structure (A,I,B) sont estimés à partir d'exemples. La stratégie

d'apprentissage est, elle aussi, identique : les matrices reçoivent des probabilités initiales aléatoires ou constantes, et un algorithme de réestimation se charge de les modifier au cours de séances d'apprentissage.

Deux problèmes se posent alors :
a) Connaissant la séquence d'observations $\{Ø_i\}$ et les paramètres initiaux {A,I,B}, comment choisir la suite d'états qui est optimale.
b) Comment ajuster les paramètres du modèle {A,I,B} afin de maximiser cette probabilité.

Les solutions à ces deux problèmes sont classiques et présentées en détail dans [Levinson, 1985] : le problème (a) est résolu par l'algorithme dit de "Forward-Backward" qui estime les densités de probabilités de présence de l'automate sur chaque état à l'instant t par des récursions temporelles de gauche à droite et de droite à gauche. Le problème (b) est résolu par des algorithmes itératifs (comme celui de Baum-Welch [Baum, 1972]) ou en utilisant comme dans le cas des modèles connexionnistes classiques des techniques de gradient.

En résumé, ces techniques de modélisation stochastique sont très efficaces pour la modélisation de processus dynamiques : les contraintes topologiques sont inhérentes à la topologie des transitions entre états. Ces techniques regroupent les avantages suivants :
(a) un formalisme mathématique très solide,
(b) la structure séquentielle des automates : les "frontières" (transitions entre états) peuvent être apprises et donc ne dépendent pas d'une manière cruciale d'un choix à priori d'une conception segmentale,
(c) un ensemble d'algorithmes d'apprentissage performant permettant l'optimisation automatique incrémentale de ses performances,
(d) et enfin, la possibilité d'ajouter des fonctions de coûts additionnelles sous forme de l'ajout d'autres types de densités de probabilités : probabilités de durées sur les états [Levinson, 1086], multiples espaces d'observations [Liporice, 1982], etc.

Mais cette structuration importante induit un lissage important des modèles stochastiques sous-jacents : ces modèles dont en fait plus aptes à mémoriser les invariances que les indices discriminants entre formes à reconnaître. Opposer deux formes suppose un outil de modélisation qui soit susceptible d'ajuster le seul critère que l'on cherche à optimiser : le taux de reconnaissance. Et si dans cette approche le niveau sous-symbolique est bien développé, l'analyse morphologique se réduit à une simple paramétrisation et le niveau symbolique est sujet à critique puisqu'il se réduit à émettre des symboles sans égard pour le code linguistique.

2.2.2.2. Le modèle « phone » et les grammaires phonétiques

Dans ce modèle les « phones » sont les éléments e_k de la substance phonique. Ces phones sont obtenus par découpage du signal en µ-segments acoustiquement homogènes [Vigouroux et al., 1985], [Zue, 1986] et représentent de ce fait une image des phases successives de réalisation des phonèmes. Certaines de ces phases peuvent être communes à deux phonèmes consécutifs. Le décodage revient alors à analyser localement la structure syntaxique des phones (en une structure des constituants acoustiques) puis à raisonner sur cette structure pour en retrouver les relations logiques (sorte de « sémantique » permettant d'associer une signification à chacun des éléments) en faisant l'hypothèse

que le phone contient une partie de l'information structurale de haut niveau — ceci parait raisonnable dans la mesure où le phone subit les déformations contextuelles et prosodiques. Insistons encore sur le fait qu'il n'y a pas de segmentation, au sens phonémique du terme, dans cette approche : il y a simplement extraction des éléments e_k, briques sur lesquelles les phonèmes $Ø_i$ sont bâtis.

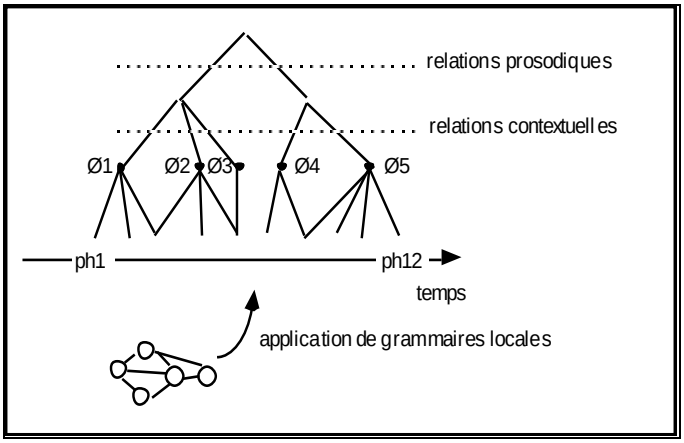

Figure 6 : Les phones Ph sont structurés par des grammaires phonétiques contextuelles. Les états de sortie des réseaux locaux "émettent" les symboles $Ø_i$.

Dans le système DIRA [Tattegrain, 1990] par exemple, cette méthode est mise en œuvre de la manière suivante :

- analyse morphologique : segmentation en phones homogènes sur un critère de stabilité d'indices acoustiques,
- analyse sous-symbolique : application locale de réseaux phonétiques qui tiennent compte de la structure interne des phonèmes et du contexte immédiat
- analyse symbolique : filtrage des hypothèses concurrentes en fonction des contraintes prosodiques et phonotactiques

Un réseau phonétique Rj est un réseau à une entrée et une sortie défini par le 5-uplet suivant:

$R_j = \{ j , S(j) , T , so_j , sf_j \}$

avec j: identificateur du réseau,

S: ensemble des états,

T: ensemble des arcs (ou transitions t_i),

so_j: état initial,

sf_j: état final.

Un tel réseau décrit une grammaire contextuelle phonétique locale.

Les états représentent toutes les réalisations possibles des différentes phases acoustiques des phonèmes.

Une transition ti est définie par:

$t_i = \{s_k, s_l, p_i, C_i, A_i\}$

avec s_k et s_l : extrémités de l'arc t_i,

p_i : score attaché à la transition si celle-ci est parcourue,

C_i : liste de contraintes devant être vérifiées lors du parcours de l'arc t_i,

A_i : liste d'actions à effectuer en cas de succès.
Par convention le score global $p = \sum p_i = 0$ si les contraintes ne peuvent être vérifiées par le contrôleur de réseau lors d'une reconnaissance locale, sinon p est la moyenne des scores p_i pour toutes les transitions i parcourues.

Les contraintes C_i sont classées en trois catégories:
- les conditions de réalisation d'une phase acoustique (par exemple si l'énergie descend à un niveau suffisamment bas alors l'état "closion-fin" peut être atteint, fig. 7)
- les contraintes induites par le contexte (par exemple l'état "friction-vocalique" ne peut être atteint que si le phonème précédemment reconnu est une consonne vocalique ou un début de syntagme avec pause, fig. 7)

Les actions A_i sont soit des procédures (calcul de paramètres, prédicats évaluables, etc.), soit des déroutements vers des règles à examiner de façon préférentielle — ce qui est en quelque sorte une forme de connaissance articulatoire.

Contraintes et actions peuvent maintenant être mises sous forme de règles de production, les contraintes en constituant les prémisses et les actions, les conclusions. Mettre en correspondance la micro-structure acoustique et la macro-structure phonétique, revient donc à cheminer dans un réseau en parcourant les états de gauche à droite selon les règles actives à chaque pas. Ce cheminement est contrôlé par le mécanisme d'application des règles.

A titre d'exemple nous décrivons ci-après le réseau des fricatives (fig.7). Ce réseau contient 7 états représentant 5 phases acoustiques et 2 états fictifs:
début: état d'entrée,
friction vocalique (début de la friction après une voyelle ou une consonne voisée),
début closion (petit chute d'intensité en début de friction),
closion fin (petite chute d'intensité en fin de friction),
friction sourde (friction sans voisement),
friction sonore (friction avec voisement),
fin: état de sortie.

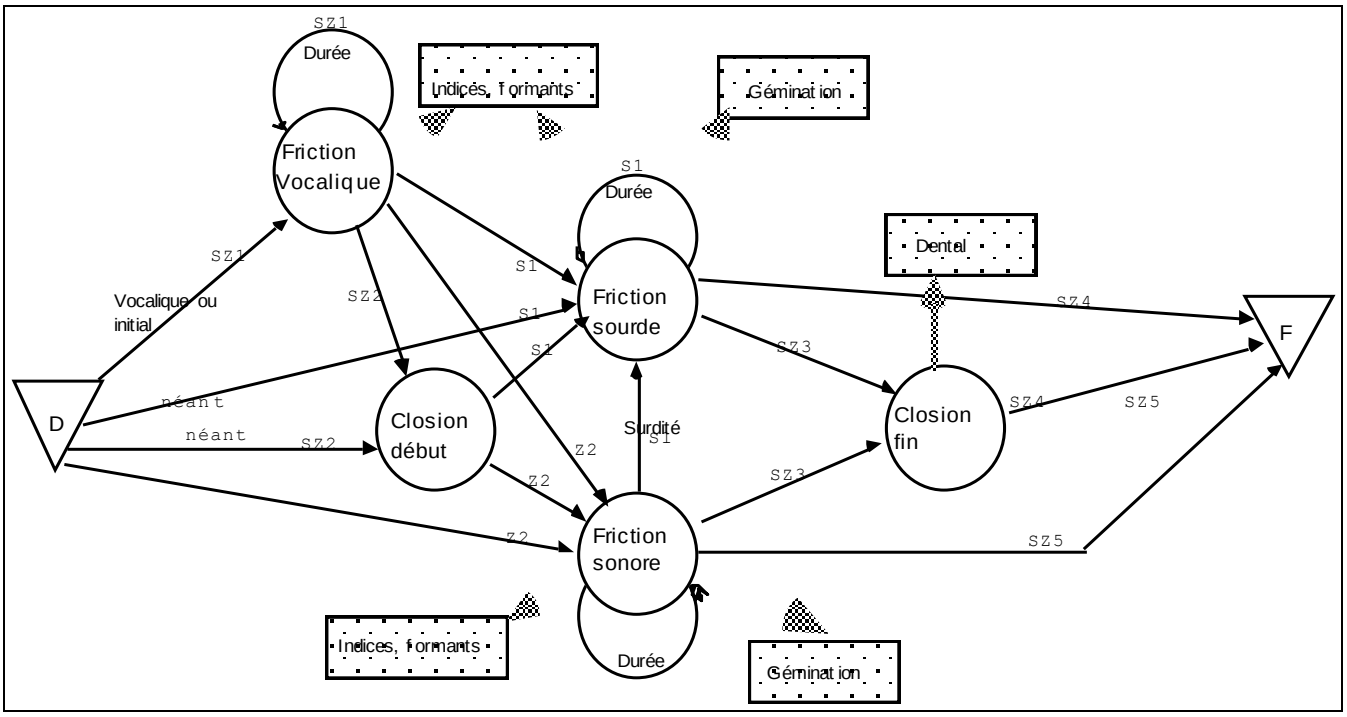

Figure 7 : Réseau des fricatives. Les notations SZij, Zij et Sij renvoient aux règles de transition pour les fricatives. Les états sont cerclés —sauf l'état de début et de fin—, les actions procédurales sont encadrées.

Ce réseau montre la diversité des réalisations d'une fricative ou en d'autres termes la syntaxe des phases acoustiques : succession de plusieurs frictions sourdes ou sonores, ou closion suivie de frictions sourdes ou sonores, etc.

La règle qui contrôle les transitions vers le nœud "friction vocalique" est la suivante:

```
SZ1-Règle Friction_vocalique
        ! transition Vocalique-fricative ou friction sonore à l'initiale
  si (Indice(Grave)>'+' OU (Fo≠0 ET Crête_max<1000 Hz))
        ! le phone candidat doit être "grave" ou présenter du voisement avec un formant en-dessous de 1000 Hz
  ET bruitplusseuil < Energie < Bruit + (3/4)* Signal-sur-bruit
        ! et être dans une fourchette d'énergie moyenne
  ET ∂(Aigu)≥'+'
        ! puis devenir progressivement plus aigu
  ET (Etat(précédent)=Friction+Vocalique
        ! la transition peut être une boucle
        OU(Contexte_antécédant='vocalique'ETEtat_précédent='néant')
        ! le contexte précédent est soit vocalique
        OU (Contexte_antécédant='pause'
            ET Etat(précédent)='néant'
        ! soit un silence
            ET pente(intensité)>pente1 ))
        ! il faut alors que dans ce cas l'intensité ait augmenté de façon significative
  alors Etat(courant)<-Friction_vocalique; p <- 1
  ET Action-proc(frontière_début<-phone_courant)
        ! on mémorise les frontières
  ET Action-règle(SZ1 OU SZ2 OU S1 OU Z2)
```

! on active ces règles pour cheminer dans les transitions suivantes

En résumé, le problème est localement bien posé par ce type de méthode, les contraintes de succession des phones sont formulées explicitement par des règles (c'est un modèle équivalent aux chaînes de Markov non cachées, le réseau émet un symbole de macro-classe pour une séquence de phones donnée en entrée), Les niveaux morphologiques, sous-symbolique et symboles sont clairement explicités. Le défaut majeur de cette approche est dans l'acquisition des connaissances qui ne peut pas se faire de manière automatique par apprentissage sur de grands corpus. On a donc recours à des expertises humaines qui ne garantissent pas la cohérence et la convergence des bases de connaissance. De ce fait, les performances de cette méthode restent globalement inférieures aux chaînes de Markov, sauf pour des phrases longues dans lesquelles les processus stochastiques se désynchronisent rapidement lorsqu'il leur manque un contrôle explicite de haut niveau.

2.2.2.3. Le modèle "phonèmes chevauchants"

Ce modèle considère que les phonèmes n'ont pas de frontière nette mais qu'ils « émergent » continument les uns par rapport aux autres au cours du temps. Le lieu de maximum d'émergence est un lieu privilégié où l'on peut espérer trouver les caractères invariants du phonème {Atal, 1983].

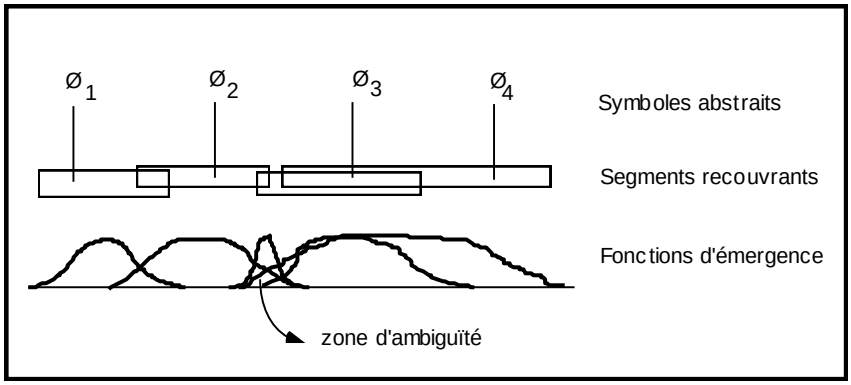

Fig 8 : Les phonèmes tentent d'être localisés directement sur le signal, au moyen d'une fonction d'émergence qui peut être interprétée soit comme le résultat du chevauchement des phonèmes dûs à la coarticulation, soit comme le résultat d'une incertitude sur la localisation.

Dans ce modèle la volonté de segmenter directement en "phonèmes" en sautant l'analyse morphologique, est clairement affirmée [Bimbot, 1988], [Deléglise, 1991] : on recherche à identifier une fonction d'émergence (fig. 8) qui peut être interprétée soit comme le résultat d'un phénomène de chevauchement de phonèmes, soit comme une incertitude de localisation d'un phonème dans le signal, soit comme une contribution pondérée de ce phonème au signal. Le phénomène de coarticulation se manifeste dans la succession recouvrante des fonctions ainsi obtenues. La méthode se résume donc à décomposer e_k sur la base $Ø_i$:

$e_k = \sum H_i.Ø_i$ où les fonctions H_i sont compactes et où les $Ø_i$ sont les prototypes acoustiques des phonèmes à identifier. Ces fonctions représentent la "contribution" du phonème $Ø_i$ à l'élément e_k qui s'en déduit dès qu'on lui a fixé des frontières. On peut aussi choisir le « centre » du phonème, c'est-à-dire le lieu de son émergence maximum, pour calculer un vecteur-paramètre caractéristique.

Dans cette approche tous les niveaux d'analyse sont confondus en un seul, puisque par la même opération on localise et on identifie le phonème ; il en résulte parfois des cas d'ambiguïté comme celui de la fig. 8 : un segment à fort recouvrement peut être identifié comme un phonème alors que ce n'est qu'une phase acoustique. Par ailleurs la notion de niveaux de structuration disparait car on s'attache à ne pas distinguer localisation et identification. Les résultats de cette méthode sont toutefois intéressants — elle est, de plus, simple à mettre en œuvre — chaque fois qu'un phonème se réalise par une seule phase acoustique (consonne liquide surtout). Les performances reposent entièrement sur la constitution du dictionnaire des Øi pour lequel des algorithmes de quantification vectorielle peuvent être mis en œuvre.

2.2.2.4. Le modèle "multicanal" et le connexionnisme

Jusqu'ici les éléments e_k étaient monodimensionnels. Or la multiplicité des traitements que l'on peut appliquer sur le signal et la nature multiparamétrique des données que l'on peut en extraire, nous amènent maintenant à examiner des modèles plus sophistiqués dans lesquels les éléments e_k sont distribués sur plusieurs canaux d'analyse. Cela engendre un niveau de complexité supplémentaire dans le décodage.

• L'analyse morphologique par "événements"

Dans ce modèle [Abry et al., 1985], le signal est examiné à travers des canaux qui réfèrent explicitement à des modèles d'organisation articulatoire et/ou perceptif. Les paramètres véhiculés dans ces canaux sont déjà fortement connotés au niveau phonétique et plus généralement linguistique. Nous les appelons indices acoustiques. Sur chacun d'eux on mène une analyse de même nature (repérage de ruptures, de points remarquables sur les courbes de variation, etc.) pour positionner des événements significatifs (débuts, fins). Les symboles associés aux événements sont de facto asynchrones. Une grammaire peut exprimer les règles d'organisation des événements en traits sur lesquels jouent le contexte et la prosodie, il faut une deuxième grammaire de structuration des traits pour aboutir aux phonèmes (Fig. 9).

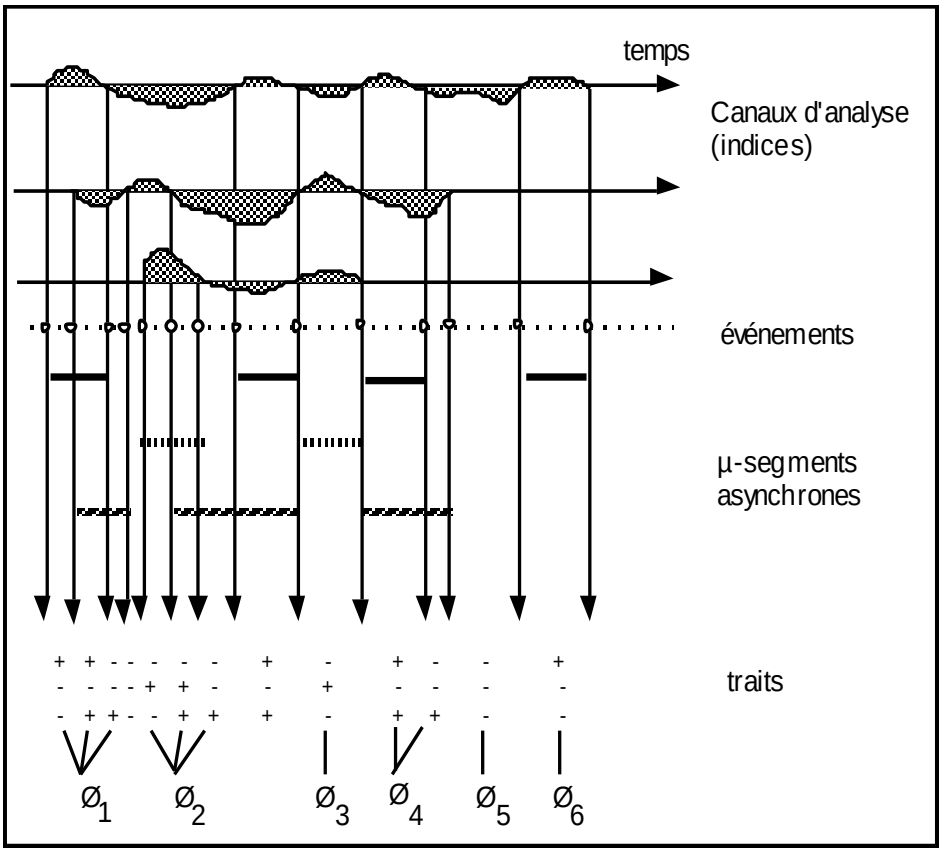

Figure 9 : Le passage des indices aux phonèmes via les événements et les traits. Une grammaire est nécessaire pour assurer le passage d'un niveau de structuration au suivant.

Le calcul des traits à partir des indices est très délicat (il faut une normalisation par locuteur), les événements "apparition" et "disparition" de traits ne sont pas toujours significatifs pour la localisation des phonèmes car la portée d'un trait déborde souvent sur plusieurs phonèmes en raison des effets contextuels. Le rôle d'une grammaire de traits a pour but précisément de raisonner à la fois sur la localisation et l'identification. Pour cela des grammaires d'association sont nécessaires (voir ci-après, par des réseaux de neurones formels par exemple).

• L'analyse morphologique par "cibles"

Ce modèle stipule que le locuteur cherche à atteindre des cibles acoustiques pour construire son message et le communiquer à l'allocutaire qui en possède un modèle de décodage. Ces cibles ne sont pas toutes atteintes (défauts de prononciation, vitesse d'élocution, bruits, etc.). Elles peuvent être visualisées comme points privilégiées d'un espace de paramétrisation adéquat [Caelen, 1986]. L'évolution de la parole se présente sous forme de trajectoire dans cet espace. Il existe deux variantes (a) monocanal et (b) multicanal (fig. 10). Nous ne nous intéresserons qu'au cas (b).

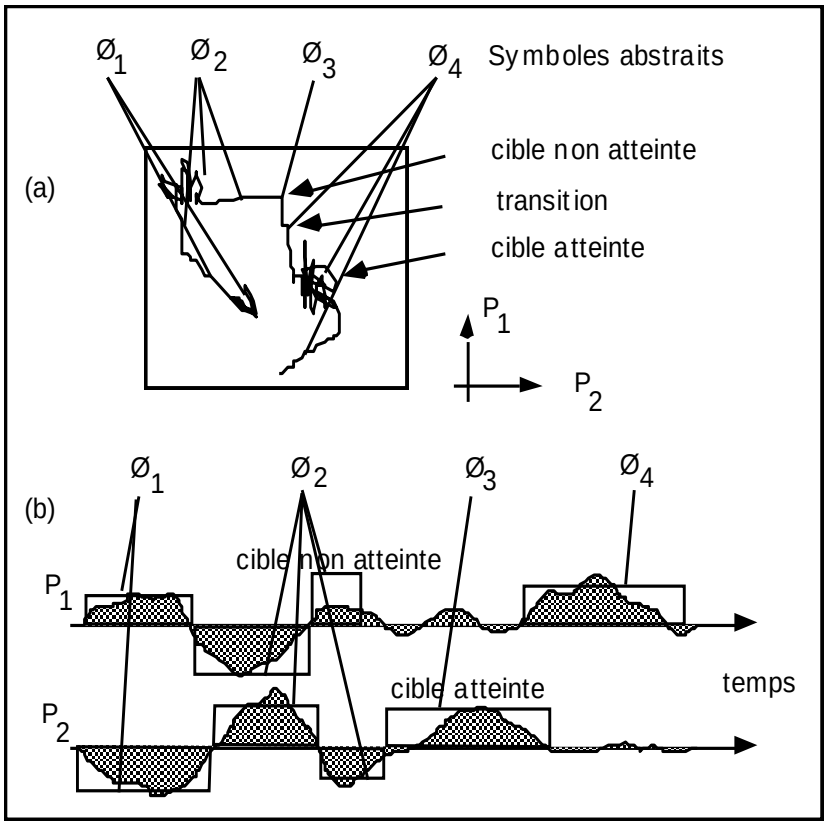

Figure 10 : Analysée sur différents canaux, la parole peut être représentée dans un espace multidimensionnel $\{P_1(t), P_2(t)...P_3(t)\}$ et analysée globalement par sa trajectoire dans l'espace $\{P_1, P_2...P_3\}$ (a) ou sur chacun des paramètres P_k (b). Sur ces deux représentations émergent les notions de cibles.

La détection des cibles se fait dans chaque canal séparément avec le modèle mécanique suivant [Piterman et al., 92] :

$$\frac{d^2\mu(x)}{dt^2} + k_1.D(\mu(x) - x_1) + k_2.D(\mu(x) - x_2) + a.\frac{d\mu(x)}{dt} = 0$$

où x_1 et x_2 sont deux attracteurs (cible et contexte), $\mu(x)$ la position (connue à travers sa densité de probabilité), k_i et a les paramètres intensités des attracteurs et frottement, D une fonction de distance. La formule la plus simple consistant à prendre $D(x)=\mu(x)=x$. Ce modèle renvoie à la mécanique d'un articulateur (système du second ordre) tentant à chaque instant d'atteindre une cible. Il possède une inertie, une élasticité et est soumis à des forces de frottement.

Une fois les cibles détectées, le problème du décodage se ramène à celui des événements, du moins formellement. Il est tout aussi nécessaire d'associer des symboles de pré-catégorisation.

•. Le traitement sous-symbolique

Le concept de réseau de neurones formels semble tout indiqué pour les grammaires d'association : on commence à bien connaître les propriétés de discrimination et d'adaptation de ces réseaux. Le problème consiste ici à associer des événements asynchrones parallèles à un symbole. Les solutions adoptées par les réseaux formels classiques pour la catégorisation de séquences consistent la plupart

du temps à adopter une métaphore spatiale du temps : une séquence d'événements temporels est transformés en une séquence spatiale en ajoutant une dimension paramétrique supplémentaire (fig. 11). Si la couche d'entrée du réseau est totalement connectée à la cartographie paramétrique de l'entrée ainsi réalisée, rien n'indique au réseau que la "trace" du nième paramètre d'entrée (figurée en gris sur la figure) est effectivement l'évolution temporelle de ce paramètre : on voit donc qu'il faudra fournir un nombre impressionnant d'exemples à un tel système pour apprendre à caractériser la dynamique du phénomène observé.

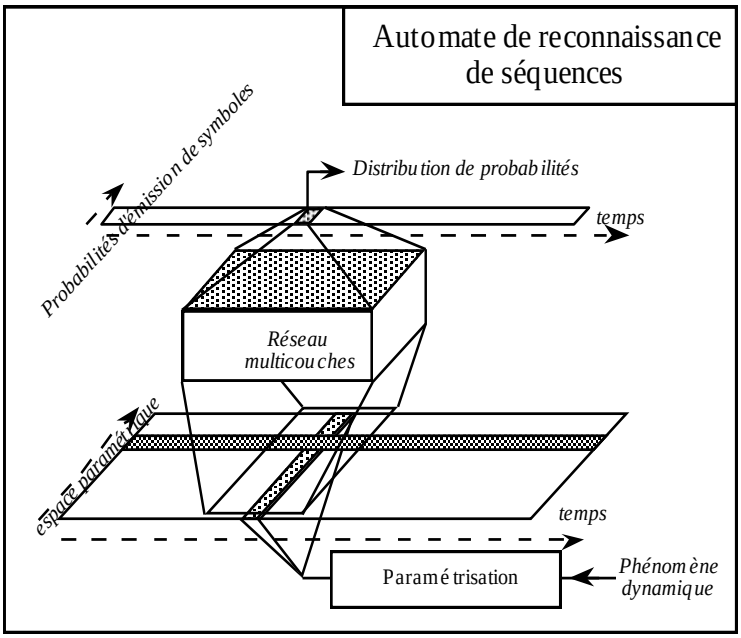

Figure 11 : Le traitement de séquences par les réseaux multi-couches classiques : la métaphore paramétrique du temps. En gris clair, est figurée les niveaux de représentations d'une trame d'entrée. En gris foncé, est figurée la "trace" ou l'évolution temporelle d'un paramètre.

Un exemple désormais classique de métaphore spatiale est le réseau NETTALK [Sejnowsky et al., 1987] : entraîné à effectuer la transduction graphème-phonème, son entrée est constituée d'une fenêtre de 7 caractères encadrant le caractère à transcrire, sa transcription phonétique étant présentée en sortie. Les deux chaînes d'entrée et de sortie avancent de manière synchrone. La chaîne de sortie peut présenter des cases vides lors par exemple de la transcription de la chaîne CH en contexte R : à C on fait correspondre le phonème /k/ puis à H on fait correspondre un phonème vide.

a) Les réseaux à délais

L'idée de Waibel [Waibel etal., 1988] a été d'avoir immédiatement identifié le problème de l'explicitation du temps par l'emploi de connections partielles. Les TDNN ("Time-delayed Neural Nets") ne sont simplement qu'une série de capteurs hautement parallélisés qui vont ne s'intéresser qu'à une partie de la fenêtre d'observation. Les étages supérieurs de traitement sont ainsi capables, par cette contrainte structurelle, d'effectuer des comparaisons des sorties à divers instants (calculs d'équations aux différences).

Les TDNNs tentent de répondre aux deux problèmes suivants :

a) pouvoir prendre en compte les relations temporelles entre événements d'entrée,
b) rendre invariant en translation temporelle cette identification (la fenêtre ne devrait pas avoir besoin d'être précisément centrée sur l'événement à analyser).

Ces deux problèmes sont résolus par la structure même des connexions (fig. 12 et 13) mais aussi par une particularisation de l'algorithme d'apprentissage : la duplication des traitements avec un décalage d'une unité de temps est accompagnée d'un lissage des changements de poids lors de la séance d'apprentissage. Si chaque ensemble d'unités à délais temporels est chargée d'observer et de détecter le même phénomène, leurs poids doivent être égaux : ils sont donc appris normalement d'une manière indépendante puis moyennés sur toute la fenêtre et celle-ci est alors affectée à chacun des poids (wi, wj...).

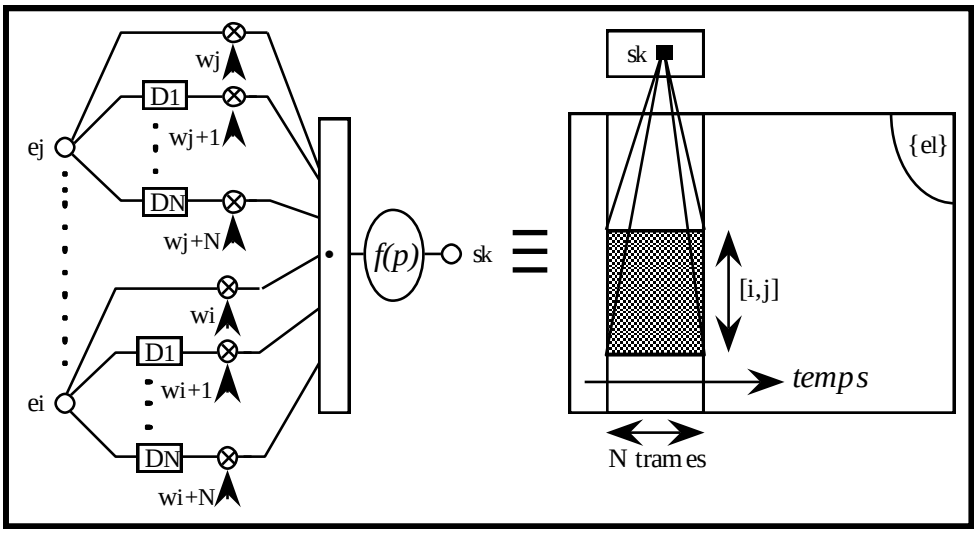

Figure 11 : Une unité de traitement TDNN.

Le réseau final est figuré ci-dessous : l'assemblage d'unités TDNN se fait de la même manière sur toutes les couches, la couche de sortie se chargeant d'effectuer la décision finale en intégrant toutes les sorties sur la fenêtre totale.

b) les réseaux à rétroaction

Une solution très différente de la métaphore spatiale à décalage est de représenter le temps par l'effet qu'il a sur le traitement lui-même : il faut pour ceci donner au réseau des propriétés dynamiques qui agissent sur lui-même. La plupart des travaux sur les réseaux classiques se sont attachés à convertir une série de paires d'observations (entrée-sortie) en un système de prédiction de nouvelles données : la capacité générale d'interpolation (généralisation) des réseaux est souvent mise en avant pour les comparer avec les modèles stochastiques. L'environnement est la source essentielle des stimuli et non le récipiendaire d'actions : les réseaux classiques résolvent des problèmes directs ("forward problems") — dans le sens, où ils cherchent à interpoler par un ensemble de fonctions non-linéaires un morphisme entre deux espaces. Or la plupart des séquences observables sont le sous-produit d'un système dynamique sous-jacent : on peut voir alors le réseau comme un système ayant à résoudre un problème inverse. Comment retrouver les caractéristiques du système dynamique générateur de séquences à partir de ses réalisations ? (par exemple, sachant que le système est du deuxième ordre, comment

retrouver ses caractéristiques à partir de la réponse du système à un pic de Dirac....).

Jordan [Jordan, 1988] a proposé une architecture générale de réseaux dits séquentiels pour faire ce type d'opérations. L'idée est de créer des niveaux de traitement appelés unités d'états, qui reçoivent l'entrée d'un certain nombre de cellules du modèle direct au temps t (cycle) t, effectuent un certain traitement (comme un neurone formel) et délivrent leur activation au temps t+1. Ces unités d'état peuvent recevoir leur propre activation, l'activation des unités de sorties [Jordan, 1989b] voire d'unités cachées [Elman, 1989], [Watrous et al., 190. Ce type d'architecture a été étudié dans un certain nombre de tâches avec un succès certain.

2.2.3 Discussion

Dans toutes ces approches, l'un des aspects — morphologique, sous-symbolique et symbolique — est toujours favorisé par rapport aux deux autres :

- le modèle "trame" et HMM n'a pas de traitement morphologique ni de raisonnement sur le niveau symbolique,
- le modèle "phone" à grammaire phonétique a des difficultés pour rassembler et utiliser des connaissances explicites aux niveaux sous-symbolique et morphologique,
- les modèles "réseaux formels" font des associations et des discriminations mais pas de raisonnement. Ils conviennent bien au niveau sous-symbolique (nous n'entrerons pas ici dans les querelles entre intelligence artificielle orthodoxe et neurosciences, qui n'apportent à ce débat que des arguments négatifs aux deux positions extrêmes où les uns prétendent qu'on ne peut pas raisonner avec une architecture de réseau formel et où les autres prétendent pouvoir tout faire avec ces réseaux).

L'apprentissage joue un rôle important, non parce qu'il s'agit de masquer l'ignorance de l'expert du domaine mais parce qu'il est nécessaire d'aboutir à des systèmes robustes au bruit et aux locuteurs, en un mot adaptables. Pour cela il est nécessaire d'avoir des systèmes qui ont la mémoire longue, qui établissent des analogies entre des situations déjà rencontrées, qui puissent faire des raisonnements et des synthèses de connaissances sans simplement se contenter d'emmagasiner des quantités importantes de données. Une base de règles figée n'est pas suffisante pas plus qu'un système markovien ou un réseau formel à court terme : il faut un niveau de métaconnaissance et d'auto-observation.

Du côté du raisonnement il en est de même : la substance sonore est un code organisé qu'il s'agit de décoder. Elle a ses règles, celle du langage, que seul l'être humain parmi les êtres vivants est capable de comprendre. Certes, ces règles sont inconscientes mais elles existent. Vouloir s'en passer par le "tout stochastique" conduit à des échecs retentissants dès que l'on veut dépasser la "surface" du code et relier la forme et le contenu.

Enfin, du côté de la morphologie, même constat : le traitement du signal doit être guidé par des objectifs de décodage, comme la recherche de configurations (ceci est encore vrai pour les êtres vivants chez lesquels on a trouvé des détecteurs de traits, par exemple, chez la chauve-souris) ou la détection de paramètres ad hoc (ce qu'il ne faut pas confondre avec la détection d'invariants). Bien sûr la transformée de Fourier reste très utile mais peut-on s'en contenter ? Il ne suffit pas non plus de

trouver des paramètres “statistiquement” meilleurs que d’autres : il faut que ces paramètres soient “explicatoires” pour pouvoir entrer dans un raisonnement.

2.3. Les différentes architectures des systèmes de reconnaissance

Les systèmes de compréhension automatique de la parole intégrant l'ensemble des traitements acoustiques et linguistiques sont apparus vers 1970, date qui correspond avec une intensification des recherches en IA (Intelligence Artificielle). A cette époque, les systèmes tentent d’utiliser toutes les sources de connaissances possibles (phonétique, lexique, syntaxe-sémantique, pragmatique, prosodie) en les faisant intervenir au moment le plus opportun dans le processus général de décodage. Le problème de l’organisation des systèmes se pose donc clairement, à la fois pour représenter les connaissances utiles et pour planifier ou gérer leur utilisation. Il faut remarquer que ce problème est rendu plus difficile du fait que les connaissances ne sont pas, *a priori*, homogènes entre elles et qu’elles interviennent à des niveaux différents. Des architectures logicielles sophistiquées ont été proposées qui ont posé le problème fondamental de la stratégie d'interaction entre les différentes sources de connaissances. Ce problème n’est pas encore bien résolu de nos jours, mais de nouvelles voies de recherche apparaissent à travers les architectures distribuées qui permettent un certain parallélisme des processus qui peut rendre caduques les stratégies de planification séquentielle utilisées jusque-là. Avant, d’étudier en détail les possibilités offertes par ces architectures, nous allons examiner les propriétés des architectures les plus caractéristiques devenues maintenant classiques. Sur la figure 12 sont présentés les différents types d'organisation de connaissances dans les systèmes de reconnaissance automatique de la parole [Reddy, 1975] à l'époque des recherches les plus intenses sur cette question (1970-1980). Pour les comparer entre elles, nous définissons les critères suivants :

• Ctl = le contrôle,
• F = la fiabilité
• P = la prédictibilité,
• C = la complexité,
• M = la maintenance,
• TR = le temps de réponse,
• Perf = les performances,
• Fonc = les fonctionnalités.

Dans les systèmes hiérarchiques ascendants (fig. 12.a) l'organisation est très simple : les données acoustiques se propagent du niveau bas (acoustico-phonétique) aux niveaux supérieurs (linguistiques) par abstraction progressive des hypothèses jusqu'à ce que le sens de la phrase soit obtenu. Ce type de système est relativement inefficace parce que pour assurer une certaine fiabilité des informations transmises il faut transférer une grande quantité d'hypothèses d'un niveau à l'autre (stratégie dite en “largeur”), ce qui multiplie inutilement la combinatoire de recherche en faisceau et ne fournit aucune assurance sur la présence de la bonne hypothèse dans la liste propagée. Les processus sont séquentiels ou au mieux en "pipe-line". Par contre, la mise en œuvre de tels systèmes ne pose pas de problème de génie logiciel particulier.

(a) Système hiérarchique ascendant = Organisation taylorienne

- Ctl = gestion "pipe-line" de processus, contrôle simple par les données
- F = repose sur le module le plus fragile (et les niveaux bas)
- P = comportement entièrement prédictible
- C = interchangeabilité des modules si les structures de données sont bien définies
- M = maintenance ramenée à la maintenance des modules de base
- TR = traitement séquentiel donc long
- Perf = recherche en largeur (coûteux)
- Fonc = pas de retour arrière possible en cas d'erreur d'hypothèses, pas d'adaptation globale, pas de guidage sur les "attentes"

Dans les systèmes hiérarchiques descendants ou génératifs (fig. 12.b) la méthode utilisée par chaque niveau est l'analyse par synthèse. En partant du haut, chaque niveau propose des hypothèses à vérifier par le niveau immédiatement inférieur. A son tour, ce niveau, après une série de vérifications sur ses propres connaissances ou en communicant les hypothèses aux niveaux qui lui sont inférieurs s'il manque d'informations, donne les scores des hypothèses proposées au niveau supérieur. Ainsi une décision peut être prise à un niveau quelconque généralement sous la forme d'une réfutation puisque les propositions sont faites au niveau supérieur. Ce type de système est relativement répandu à cause de la simplicité à exprimer les connaissances linguistiques de manière générative. Cependant, les limites de ce type de système sont bornées par les limites des représentations linguistiques explicitées dans le système qui interdisent à leur tour pratiquement toute prise en compte des inattendus.

(b) Système hiérarchique descendant = Organisation de marché (offres-demandes)

- Ctl = gestion "pipe-line" d'hypothèses
- F = repose sur la puissance du module le plus haut
- P = comportement entièrement prédictible
- C = raisonnement sur hypothèses
- M = maintenance plus coûteuse que pour le type (a)
- TR = traitement séquentiel mais moins long que le type (a)
- Perf = recherche en profondeur d'abord (risque d'impasse)
- Fonc = retour arrière, pas d'adaptation globale, guidage par les "attentes", décision distribuée, limitation par la puissance du générateur d'hypothèses de chaque niveau

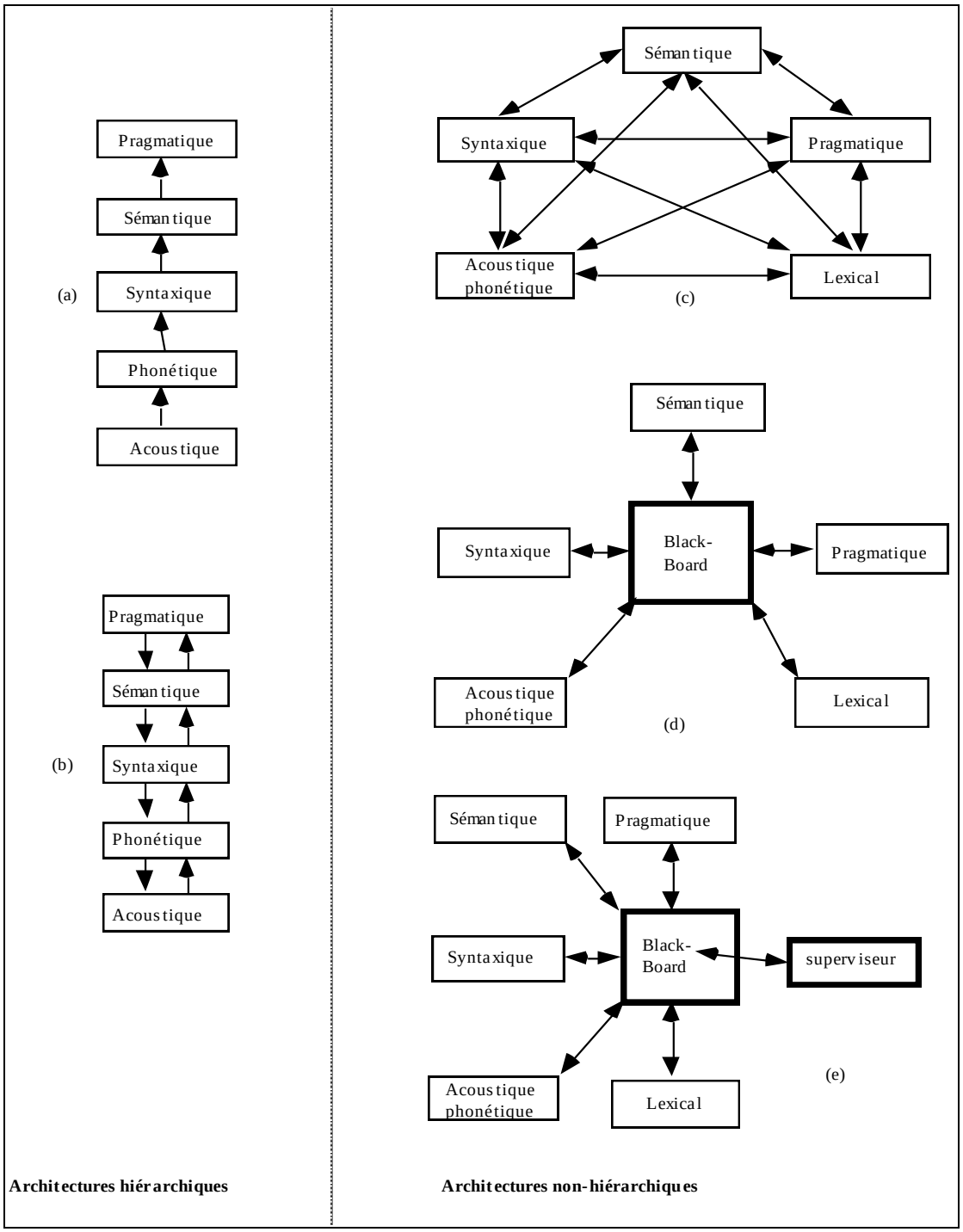

Fig. 12 : Les différents types d'organisation de sources des connaissances dans un système de reconnaissance automatique de la parole. (a) hiérarchique, (b) génératif, (c) non hiérarchique, (d) structure de Black-Board (accès asynchrone), (E) structure de Black-Board avec superviseur (accès synchrone). Les flèches indiquent le sens des échanges d'informations entre les modules.

Les systèmes du type “non hiérarchique” offrent des processus de coopération entre toutes les sources de connaissance ce qui évite les défauts de propagation des hypothèses des systèmes hiérarchiques (fig. 12.c). Ce type d'architecture, dite multi-agents, pose évidemment le problème du contrôle du flux

d'informations et de communication des agents.

Une classe importante de tels systèmes est constituée par les systèmes à structure de Black-Board (BB) (Hearsay II par exemple [Lowerre 79]) organisés autour du paradigme du "comité d'experts" (fig. 12.d). Chaque expert peut envoyer des hypothèses à un autre expert à travers le Black-Board qui est une mémoire commune évolutive et qui présente à chaque instant l'état de la situation. Toutes les informations sur les résultats de la reconnaissance à un instant donné, sont regroupées dans le BB dont la structure peut refléter ou non les différents niveaux (phonétique, lexical, syntaxique....). Les experts peuvent lire et écrire de manière anonyme dans un ordre quelconque (stratégie opportuniste) dans le BB dès qu'ils ont quelque chose "à dire", c'est-à-dire qu'ils peuvent ajouter une information dès qu'ils peuvent la construire au vu des données du BB ou au contraire en supprimer une si elle n'est pas compatible avec les connaissances qu'ils ont.

Les avantages de ce type de structure BB sont :

1. de minimiser la duplication des informations,
2. que les résultats fournis par un expert sont accessibles à tout moment et sans ordre hiérarchique par les autres experts,
3. qu'on peut définir les liens entre les hypothèses des différents niveaux et donc réduire les calculs en cas de retour en arrière ("backtracking"),
4. que les étapes de la reconnaissance étant représentés sous forme de graphe, il est possible de poursuivre une stratégie plus générale et d'invoquer des experts en des points où on localise des informations intéressantes ou, au contraire, des difficultés particulières.

(d) Société d'experts = Organisation de séances de "brainstorming"

- Ctl = autocontrôle, stratégie opportuniste de chaque expert,
- F = un expert peut détruire une info. Utile à un autre agent,
- P = comportement non prédictible,
- C = simplifiée par hiérarchisation du BB, complexité individuelle des modules
- M = en apparence facile si les experts n'échangent que des résultats et non des états sur leur raisonnement
- TR = non-contrôlable (remise en question sans fin des hypothèses), semi-parallèle
- Perf = ilots de confiance, sens de parcours quelconque (ascendant-descendant), réduction de la quantité de données échangées, focalisation locale sur un problème résistant,
- Fonc = focalisation des ressources sur les problèmes difficiles, souplesse totale du raisonnement

L'inconvénient de la structure BB est qu'un expert peut détruire une situation ou des connaissances qu'il a jugées sans intérêt alors qu'un autre expert aurait eu besoin de ces connaissances pour son raisonnement. Un autre inconvénient majeur est que l'on n'est pas assuré de la convergence du processus, le comportement de l'ensemble du système n'étant pas prédictible.

Pour éliminer ce dernier inconvénient, un superviseur peut être ajouté à la structure de BB pour la contrôler et planifier les actions des experts. Ce superviseur doit posséder une méta-stratégie de contrôle pour guider le système de compréhension dans son ensemble en évitant les conflits entre les

différents experts. Il doit donc avoir une compétence non plus sur le problème de compréhension ou de reconnaissance eux-mêmes mais sur le comportement des experts individuellement et lors de leurs interactions. Il doit de ce fait connaître leur mode de raisonnement. Cette structure BBS (Black-Board Supervisé) est présentée sur la fig. 12.e : le superviseur a ici deux fonctions, (a) gérer les flots de données dans le BB et (b) ordonnancer les actions des experts. Ainsi le problème de la stratégie est de la compétence du superviseur : cela renvoie le problème à un autre niveau et pose celui de savoir quelle est la meilleure méta-stratégie possible pour le superviseur. Une autre question est de définir la granularité des experts, leurs fonctions et compétences, et de modéliser leur raisonnement pour le rendre visible aux autres experts via le superviseur. Le système DIRA [Caelen 90] est un bon exemple de système de compréhension de la parole de type BBS.

(e) Expertise planifiée = Organisation supervisée d'une salle de classe

- Ctl = contrôle des tours d'intervention, des conflits, planification des tâches
- F = repose sur le contrôleur de plans
- P = prédictibilité assurée par le superviseur
- C = gestion des stratégies très complexe
- M = maintenance difficile car tous les modules interfèrent
- TR = séquentiel ou parallèle donc très variable selon les problèmes posés
- Perf = recherche en profondeur guidée par les buts
- Fonc = retour arrière possible, adaptation globale, guidage par les "attentes", décision centralisée, auto-connaissance du système sur lui-même, apprentissage possible

Devant les échecs relatifs des systèmes BB, on a cherché de 1980 à 1990 des architectures plus simples et plus homogènes, fondées comme le système HWIM, sur une représentation unique par réseau en homogénéisant toutes les sources de connaissances — on considère en effet que le concept de *séquence* est apte à rassembler les décompositions de mots en phonèmes aussi bien que de phrases en mots. Cette représentation convient particulièrement bien aux modèles markoviens généralisés. Mais, depuis l'apparition de techniques mixtes, notamment l'utilisation de réseaux de neurones pour la classification avant ou pendant la reconnaissance, ce concept unificateur de séquence n'est plus aussi fécond pour l'intégration de plusieurs techniques. Aussi voit-on depuis quelques années un regain d'intérêt pour les architectures logicielles dans les systèmes de reconnaissance. Ce regain d'intérêt correspond également aux récentes avancées de l'IA en matière de systèmes multi-agents qui fondent le domaine de l'IAD (Intelligence Artificielle Distribuée), c'est-à-dire aux systèmes de la fig. 12.c que nous avions abandonnés pour un temps.

Une des particularités des systèmes multi-agents est le fait qu'ils peuvent être distribués et autonomes, coopérer pour résoudre un même problème, ces deux propriétés rendant leur fonctionnement parallélisable et concurrent. Ainsi ils offrent quelques attraits pour traiter la robustesse d'une part — par coopération de traitements complémentaires en parallèle — et l'efficacité d'autre part — par intégration de traitements concurrents. Ainsi se retrouve-t-on de nouveau en face des questions du choix des modules, de la définition de leur granularité, du contrôle du système, de la coopération-concurrence des agents, de l'organisation "sociale" de leur travail, etc. On distingue actuellement en IAD deux grandes classes d'agents :

- les agents réactifs, qui répondent de manière réactive, qui sont à "grain fin" et qui ont des capacités de raisonnement limitées,

- les agents cognitifs à “gros grain” qui ressemblent beaucoup aux experts et qui ont des capacités de raisonnement.

Le principe général des systèmes multi-agents est qu’ils travaillent sans contrôle externe apparent, à la manière des sociétés d’animaux ou humaines (ou de modèles biologiques ou neurobiologiques). La notion de comportement émergent est essentielle dans ces systèmes. Certains agents peuvent se reproduire ou se dupliquer pour augmenter le parallélisme ou “réunir des forces” supplémentaires pour résoudre un problème. Bref les sources d’inspiration et les métaphores sont nombreuses dans le domaine de l’IAD ce qui à la fois nuit quelque peu à sa crédibilité mais qui d’un autre côté lui donne une richesse qu’il ne faut pas rejeter *a priori*.

(c) Multi-agents = Organisation par répartition du travail et coopération

- Ctl = émergentiste sous contraintes, les conflits se résolvent spontanément
- F = repose sur le bon choix des agents et leur compétence
- P = comportement non prédictible
- C = interchangeabilité théorique des agents
- M = difficulté dans la gestion des messages
- TR = temps de stabilisation parfois long
- Perf = parallélisme
- Fonc = décision émergente, adaptation globale, guidage à la fois sur les "attentes" et les entrées

Pour explorer ce domaine et ses possibilités en traitement de la parole, nous avons tenté de mettre en œuvre un système de compréhension automatique de la parole (MICRO) fondé sur les concepts de l’IAD appliqués à la neuropsychologie. Nous partons du constat que l’humain a des facultés d’adaptation aux phénomènes inattendus et spontanés que n’a pas la machine : la plupart des systèmes actuels de reconnaissance de la parole sont construits sur un principe hiérarchique découpant les fonctions de classification, de filtrage et d’interprétation, en blocs indépendants et figés — ainsi en est-il des systèmes à bases de connaissances hétérogènes mais aussi des systèmes statistiques ou des systèmes auto-organisateurs. Il est bon de constater qu'en dépit de leurs bonnes performances d'ensemble, ces systèmes restent incapables d'affronter les situations inattendues ou de s’adapter à des conditions ou locuteurs nouveaux, facultés pour lesquelles l'être humain est particulièrement efficace.

2.4. Les systèmes fondateurs

Les systèmes de reconnaissance développés aux Etats-Unis, dans le cadre du projet ARPA-SUR (Advanced Research Projects Agency of the Department of Defense - Speech Understanding and Recognition) [Klatt, 1977] constituent une référence dans le domaine de la parole, surtout parce qu'ils représentent une avancée décisive autant que spectaculaire. Ce projet (1971-1976) a suscité une intense activité dans le domaine. Les principaux systèmes qui ont été développés sont :

1. Systems Development Corporation : SDC system.
2. Bolt Beranek and Newman Inc. : HWIM.
3. Carnegie-Mellon University : HEARSAY II.
4. Carnegie-Mellon University : HARPY.

2.4.1. System Development Corporation : SDC

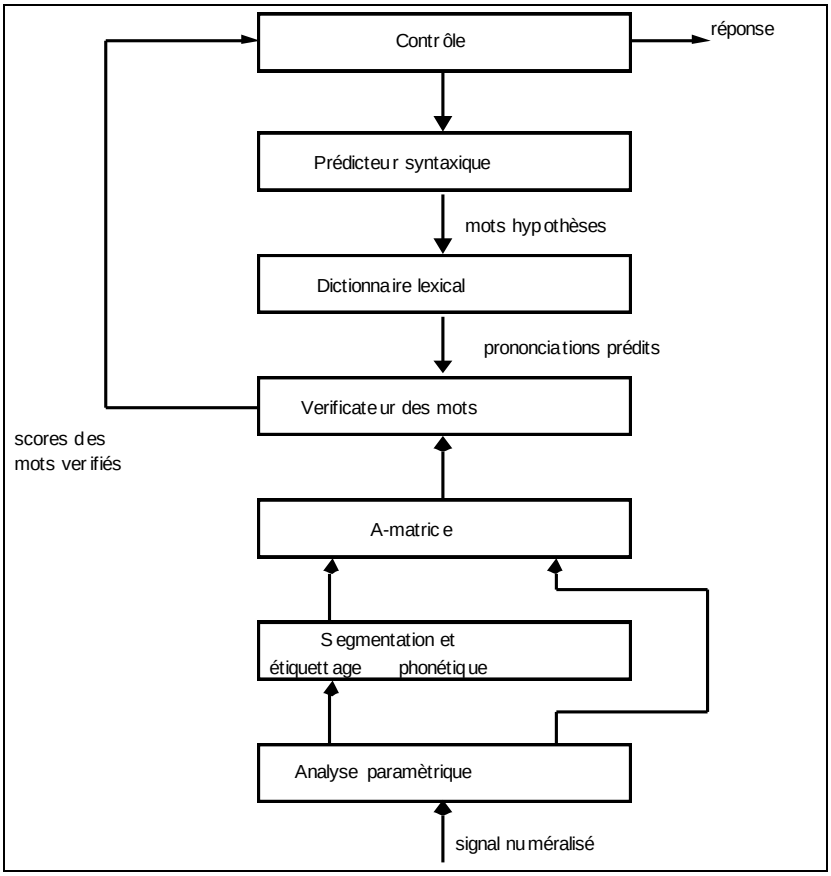

Fig. 13 : Schéma général du système SDC.

La fig. 13 montre l'organisation du système de compréhension de la parole SDC (System Development Corporation) [Ritea, 1975]. D'abord le signal de parole est paramétré (calcul des formants, et autres paramètres), ensuite les transcriptions phonétiques sont obtenues sur les segments homogènes du signal avec plusieurs alternatives. Ces informations phonétiques sont placés dans une matrice phonétique (A-matrice) pour les utiliser dans la stratégie descendante. Cette stratégie commence par la prédiction de la liste des mots possible au début de la phrase. L'étape de contrôle cherche les variantes phonologiques des mots lexicaux et les met sous forme d'un réseau phonétique comme il est illustré dans la figure 14. Ensuite les réseaux phonétiques sont envoyés, l'un après l'autre, au vérificateur lexical pour tester le score de correspondance avec la matrice phonétique (A-matrice), le signal d'entrée est traité de gauche à droite.

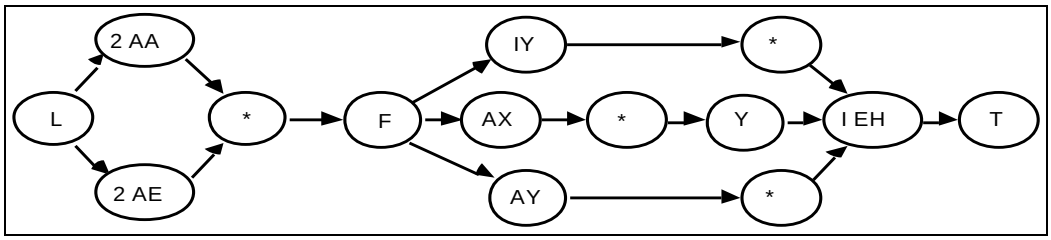

Fig. 14 : Représentation du mot "Lafayette" dans le dictionnaire lexical. (*) est un marqueur de séparation des syllabes.

Le vérificateur des mots s'appuie sur la structure syllabique de mot. Il examine la matrice acoustique à partir de la voyelle prédite ou d'un de ses allophones propres et tente de cheminer vers les consonnes adjacentes. En raison de la difficulté d'avoir une correspondance exacte, le vérificateur lexical dispose de techniques tolérantes pour calculer la probabilité d'avoir un mot donné dans la matrice phonétique. Le bloc de contrôle décide de la validité des mots trouvés en fonction du score avant de poursuivre la recherche dans la phrase. Il génère alors les mots possibles. La stratégie de contrôle dans le système SDC est très proche de celle de (Hearsay I speech understanding system) [Reedy et al, 1973] et d'un autre système développé au laboratoire Lincoln [Klovstad et al, 1975].

La performance du système SDC est de 65% pour un lexique de 200 mots. En résumé, on peut dire que la stratégie utilisée est une stratégie ascendante au niveau acoustique et une stratégie descendante depuis les niveaux syntaxique, sémantique et lexical.

3.5.2. Bolt Beranek and Newman Inc. : HWIM

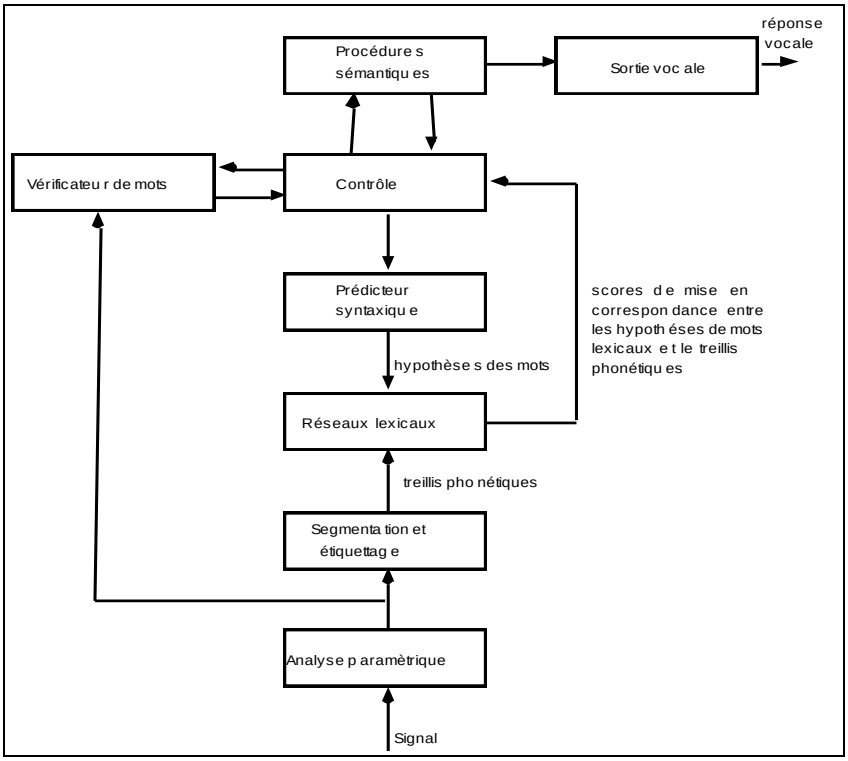

Fig. 15 : Schéma général du système BBN HWIM.

L'organisation générale du système Hwim (Hear What I Mean) est illustré sur la fig. 15 [Woods et al, 1976]. Le fonctionnement de ce système peut être résumé comme suit :

- les formants et autres paramètres sont calculés sur le signal de parole échantillonné.
- ces paramètres sont utilisés pour déterminer la transcription phonétique sous forme de treillis (plusieurs alternatives d'étiquettes pour chaque segment du signal).
- l'identification de la phrase commence par la recherche dans le treillis des hypothèses de mots lexicaux (n'importe où dans la phrase). Ces hypothèses de mots vont être considérées comme des points d'ancrage. Ces points d'ancrage sont utilisés pour construire des hypothèses pour amorcer des

calculs sur des parties plus longues de la phrase. Cette étape donc fait la mise en correspondance entre le treillis phonétique et les hypothèses de mots lexicaux. La mise en correspondance est obtenue pour tous les mots proposés en utilisant un réseau de décodage lexical. Ce réseau contient les mots et toutes ses variantes phonologiques [Klovstad, 1977], [Woods et al, 1976] (voir un exemple sur la fig. 16).

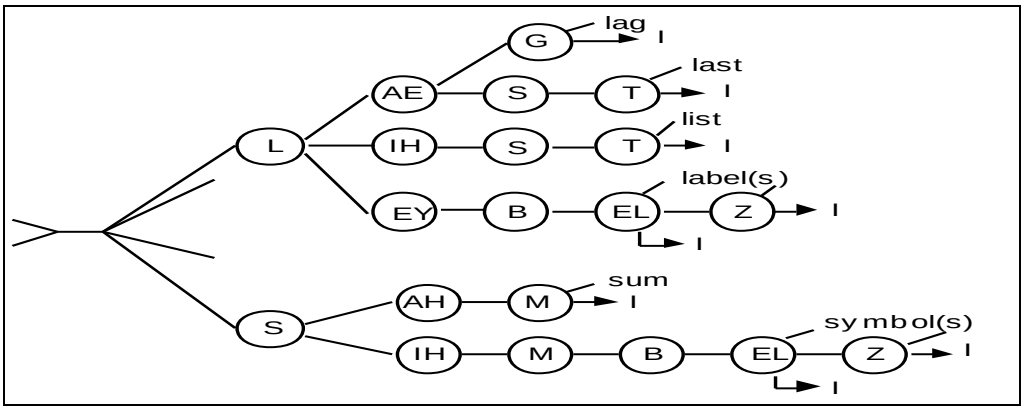

Fig. 16 : Le réseau lexical dans le système HWIM de BBN.

- le mot qui a le plus grand score est ensuite envoyé au vérificateur de mots qui retourne sur les données paramétriques pour faire des mesures indépendantes du score de la mise en correspondance de mot. La méthode de vérification adoptée est l'analyse par la synthèse [Klatt, 1975]. L'étape de vérification est très importante pour éliminer les erreurs de segmentation.

- le score de la vérification est combiné avec le score de la mise en correspondance lexicale. Si le score combiné est suffisant ce mot est envoyé au prédicteur syntaxique qui propose les mots qui peuvent apparaître à gauche et à droite de ce mot en utilisant des contraintes grammaticales. Un ATN (Augmented Transition Network Grammar) [Woods,1970] est utilisé pour caractériser les contraintes syntaxiques et sémantiques.

- chaque mot qui a un bon score, est combiné avec le premier mot qui constitue un point d'ancrage et la vérification est faite pour les deux mots ensemble. Ensuite, les hypothèses concernant des membres de phrases sont développées. Quand la phrase est complète la structure profonde de la phrase est constituée et envoyée aux procédures sémantiques pour calculer la réponse possible. Celle-ci est synthétisée en utilisant la synthèse de la parole par règles.

La performance de l'étiquetage phonétique du signal est de 52%. La taille du vocabulaire utilisée est de 1096 mots pour une application d'agence de voyages. La performance totale est de 44%. D'autres types de stratégies ont été développés dans ce système :

- Recherche d'un point d'ancrage puis développement des mots possibles à gauche et à droite.
- Recherche d'un point d'ancrage puis développement des mots possibles à gauche suivi d'un redémarrage avec une stratégie de gauche à droite.

En résumé, on peut dire que le système BBN HWIM a une stratégie purement ascendante au niveau acoustique-phonétique et une stratégie mixte aux niveaux lexique, syntaxe et sémantique.

2.4.3. Carnegie-Mellon University : HEARSAY II

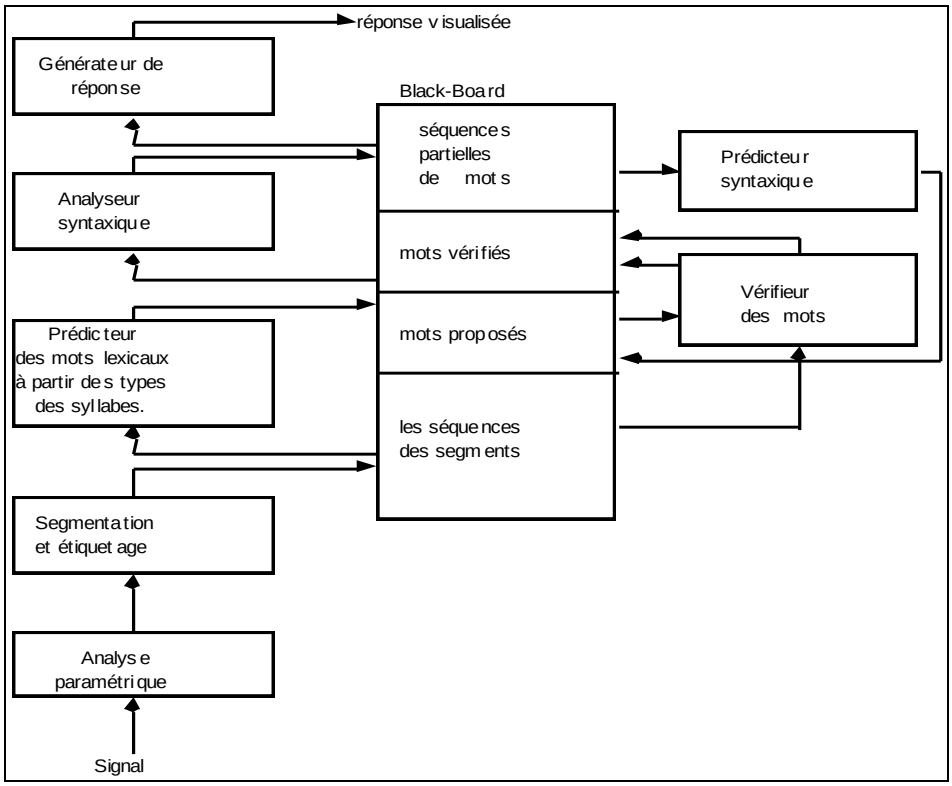

Fig. 17 : Schéma général du système HEARSAY II.

La fig. 17 illustre le schéma général du système CMU HEARSAY II [Lesser et al, 1975 ; Lesser et Erman, 1977]. Dans HEARSAY II le processus de la reconnaissance est très proche de celle de HWIM mais la philosophie d'organisation et le schéma général sont différents : c'est un système très modulaire. Le système HEASAY II est constitué de sources de connaissances asynchrones qui communiquent entre elles par le Black-Board. Les sources des connaissances sont activées selon l'état des informations dans le blackboard. Ces informations sont divisées en plusieurs catégories : des séquences d'étiquettes, des syllabes, des mots lexicaux prédites, des mots acceptés, des parties d'hypothèses de phrases. Un module accepte les informations d'un niveau quelconque et essaye de donner de nouvelles informations au niveau supérieur par analyse ascendante et au niveau inférieur par analyse descendante.

Dans un premier temps les paramètres de passages par zéro et d'énergie sont utilisés pour segmenter le signal de la parole en traits articulatoires [Goldberg et al, 1976]. Un prédicteur de mots donne tous les mots qui ont la structure syllabique compatible avec la représentation phonétique partielle. Par exemple les mots lexicaux qui correspondent à la séquence de traits (fricative, occlusive, voyelle, occlusive). Dans l'étape ascendante 70% des mots, trouvés par le prédicteur des mots, sont corrects. D'autres mots vont être trouvés par prédiction descendante par les autres étages.

Un élément de vérification donne un score à chaque mot lexical prédit. Ce score est déterminé par la comparaison entre les spectres de prédiction linéaire de mots et du signal. Le lexique utilisé, pour la vérification, est similaire au système Harpy. Les mots lexicaux dans le lexique sont définis sous la forme de spectre. Les mots prédits sont vérifiés par la comparaison des spectres probables avec les spectres sur le signal. Les mots avec de bons scores sont envoyés à l'analyseur syntaxique qui

commence à les ranger en hypothèses de phrase partielles. Les parties de phrases acceptables sont ensuite développées. La stratégie utilisée est de compléter la partie de phrase à gauche et à droite en même temps ou bien compléter la partie gauche jusqu'au début du signal puis redémarrer ensuite avec une stratégie de gauche à droite. La taille de vocabulaire utilisé est de 1000 mots pour une application de recherche de documents. La performance de la reconnaissance est de 77%, celle de compréhension est de 91%.

La structure de Hearsay II est très intéressante pour l'implantation sur des machines parallèles où on peut implanter chaque source de connaissance indépendamment. L'interaction entre ces sources se fait par le Black-Board et des variables globales. Historiquement Hearsay II a été le premier système multi-expert.

2.4.4. Carnegie-Mellon University : HARPY

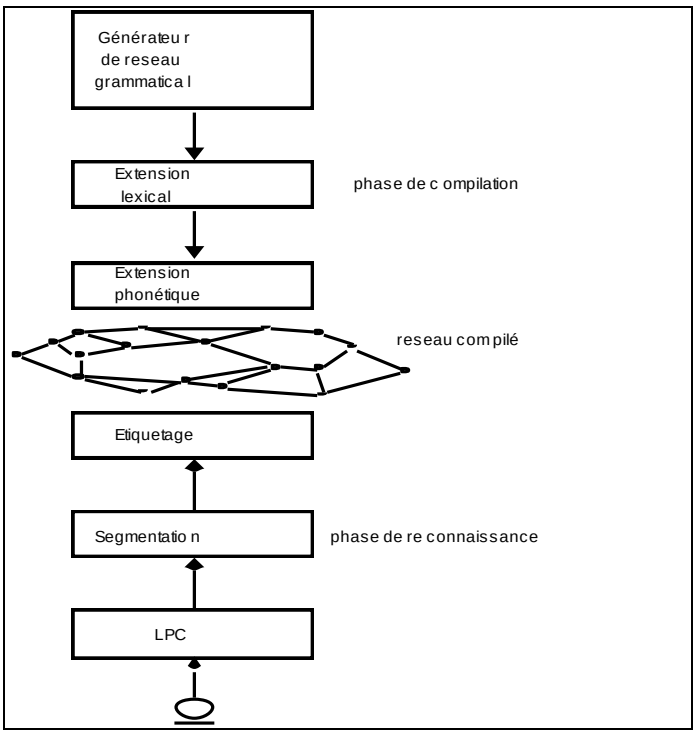

Fig. 18 : Le schéma général du système HARPY.

Le système HARPY est un système où toutes les connaissances (phonétiques, lexiques, syntaxiques-sémantiques) sont compilées dans un réseau [Lowerre, 1976]. Ce réseau contient 15000 états. Les transitions entre les différents états donnent toutes les phrases possibles. Les mots lexicaux dans le réseau sont représentés avec leurs variantes phonologiques.

La phrase à l'entrée est découpée en segments acoustiques stationnaires. Ensuite ces segments sont comparés avec des références préenregistrés. La méthode d'analyse utilisée est la prédiction linéaire (LPC) et la distance entre spectres est la distance spectrale d'Itakura.

Chaque segment acoustique est attribué à un état dans le réseau et la stratégie utilisée est de trouver le chemin ayant le meilleur score dans le réseau. La stratégie de recherche employée dans le système

HARPY est la recherche en faisceaux restreinte à quelques alternatives ayant des scores proches du meilleur score pour restreindre le temps de recherche.

La stratégie utilisée par HARPY est une stratégie de vérification. La taille de lexique est environ de 1000 mots et la performance est de 95%. Avec ce score HARPY a atteint la meilleure performance parmi les systèmes de cette époque. C'est ce qui explique qu'ensuite la majeure partie des recherches se soient concentrées sur des systèmes de ce type ayant donné plus tard des architectures fondées sur les réseaux et chaînes de Markov.

2.4.5. Systèmes plus récents

Dans les années 1985, SPHINX a été considéré comme le meilleur système de reconnaissance de la parole continue [Lee, 1988]. Il est basé sur une approche stochastique HMM. SPHINX a un vocabulaire de 1000 mots lexicaux avec un branchement lexical égal à 60. L'accès lexical est guidé par la prosodie et utilise une stratégie de regroupement d'îlots de confiances. Les résultats de reconnaissance sont très encourageants :

- 70% sans l'utilisation d'une grammaire (jusqu'au 95.8% avec une grammaire pour la reconnaissance multi-locuteur).
- 74% sans l'utilisation d'une grammaire (jusqu'au 96.1% avec une grammaire) avec adaptation au locuteur. SPHINX est utilisé pour des applications de gestion de bases de données.

BYBLOS est un système de reconnaissance de la parole continue. Il utilise une approche stochastique HMM (chaînes de Markov cachées) pour les modèles de phonèmes [Chow, 1987], [Kubala, 1988] avec un dictionnaire phonétique et une grammaire d'états finis pour un langage à complexité moyenne. Les paramètres markoviens sont automatiquement estimés à partir des données d'apprentissage. Pour chaque mot dans le dictionnaire, le modèle de chaque phonème est combiné avec les modèles de phonèmes adjacents, pour construire les modèles markoviens de mots. Ceux-ci sont ensuite utilisés pour reconnaître la phrase prononcée. Les connaissances phonétiques, lexicales et syntaxiques sont précompilées dans un réseau markovien. Ces connaissances sont utilisées dans une stratégie descendante pour limiter la recherche. La taille de vocabulaire est de 1000 mots. 80% des phrases testées sont correctes.

Les laboratoires français ont développé de leur côté de nombreux systèmes et ont contribué fortement aux avancées des recherches dans le domaine. L'école française s'est surtout distinguée par son approche analytique du problème de la reconnaissance.

A l'heure actuelle, on commence à trouver des produits commerciaux (Philips, Apple, IBM, Dragon, Microsoft, etc.) fondés pur la plupart sur des méthodes markoviennes. On constate la même tendance dans les principaux centres de recherche.

BIBLIOGRAPHIE

Décodage acoustico-phonétique

Abry, C., Benoît, C., Boé, L.J., Sock, R. (1985) Un choix d'événements pour l'organisation temporelle du signal de parole. Actes des 14èmes Journées d'Etude sur la Parole, Paris, 133-137.

Atal, B.S., (1983) Efficient coding of LPC parameters by temporal decomposition. *IEEE*, transactions on ASSP n° 24.

Autesserre, D., Rossi, M., (1985) Propositions pour une segmentation et un étiquetage hiérarchisé. Application à la base de données acoustiques du GRECO communication parlée. Actes des 14èmes Journées d'Etude sur la Parole, Paris, 147-151.

Bailly, G., Jordan, M., Mantakas, M., Schwartz, J.L., Bach, M. & Olesen, M. (1990) Simulation of vocalic gestures using an articulatory model driven by a sequential neural network, J. Acoust. Soc. Am., 87, S105.

Baum , L.E. (1972) An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes, *Inequalities*, 3, 1-8.

Bernstein, N. (1967) *The coordination and regulation of movements*, Oxford, Pergamon Press.

Bimbot, F., Chollet, G., Deléglise, P., Montacié, C. (1988) Temporal decomposition and acoustic-phonetic decoding of speech. Proc. ICASSP, 425-428.

Bourlard, H. & Wellekens, C.J. (1989) Links between Markov Models and multilayered perceptrons, *Advances in neural information processing systems*, 1, Morgan Kaufmann, 502-510.

Bridle, J.S. (1984) Stochastic models and template matching : some important relationships between two apprently different techniques for automatic speech recognition, *Proc. Inst. of Acoustics*, Autum Conf., 1-8.

Caelen, J., Caelen, G. (1981) Indices et propriétés dans le projet ARIAL II. Actes du séminaire "Processus d'encodage et de décodage phonétique", GALF-CNRS.

Caelen, J. (1986) Speech-segmenting and kinematics. Proc. of Montreal Symposium on Speech Recognition, McGill University, 77-79.

Deléglise, P. (1991) Une architecture logicielle pour le décodage acoustico-phonétique, application à la détection d'événements phonétiques. Thèse de doctorat d'état, Paris VI.

Elman, J.L. (1989) Structured representations and connexionist models, *CRL Technical report 8901*, Center for Research in Language, Univ. of California, San Diego.

Falletta, N. (1989) *Le livre des paradoxes*, Belfond, Paris.

Fodor, J. & Pylyshyn, Z. (1988) Connectionism and cognitive architecture : a critical analysis, in *Connections and symbols* (Pinker S. & Mehler J. Eds), MIT Press, Cambridge, 3-71.

Fowler, C.A. & Turvey, M. (1980) Immediate compensation for bite-block speech, *Phonetica*, 37, 306-326.

Hare, M., Corina, D. & Cottrell, G. (1988) Connectionist perspective on prosodic structure, *Center for Research in Language Newsletter*, Univ. of California, 3-2.

Huang, W., Lippmann, R.P. & Gold B. (1988), a neural net approach to speech recognition, *IEEE Conf. on Acoust., Speech and Sig. Proc.*, 99-102.

Johansson, G. (1950) *Configurations in event perception*, Almquist & Wiksell, Uppsala.

Jordan, M. (1988) Supervised learning and systems with degrees of freedom, *COINS Technical report 88-27*, University of Massachusetts, Amherst, MA.

Jordan, M. (1989a) Indeterminate motor skill problems. In *Attention and Performance*, XIII, (M. Jeannerod, Editor). Hillsdale, NJ; Erlbaum.

Jordan, M. (1989b) Serial order: A parallel, distributed processing approach. In Advances in Connexionist Theory: Speech, (J.L. Elman & D.E. Rumelhart, Editors). Hillsdale, NJ; Erlbaum.

Jordan, M. (1990) Indeterminate motor skill learning problems. In *Attention and Performance*, XIII (Jeannerod, M., Editor), MIT Press.

Klatt, H.H. (1977) Review of the ARPA speech understanding project, *J. Acoust. Soc. Amer.*, 62, 1345-1366.

Larar, J.N., Schroeter, J. & Sondhi, M.M. (1988) Vector quantization of the articulatory space, IEEE Trans. on Acoust., *Speech and Sig. Proc.*, 36, 1812-1818.

Levinson, S.E. (1985) Structural methods in automatic speech recognition, *IEEE Proceedings*, 1625-1650.

Levinson, S.E. (1986) Continuously variable duration hidden Markov models for automatic speech recognition, *Computer, speech and Language*, 1-1, 29-45.

Lindblom, B., Lubker, J. & Gay, T. (1979) Formant frequencies of some fixed-mandible vowels and a model of speech-motor programming by predictive simulation, *Journal of Phonetics*, 7, 141-161.

Liporice, L.A. (1982) Maximum likelihood estimation for multivariate observations of Markov sources, *IEEE Trans. on Information Theory*, IT-28, 5, 729-734.

Lubensky, D. (1988) Learning spectral-temporal dependencies using connectionnist networks, *IEEE Conf. on Acoust., Speech and Sig. Proc.*, 418-421.

Marr, D. (1982) *Vision: a computational investigation into the human representation and processing of visual information*, Freeman, San Fransisco.

Marteau, P.F. (1988) Cibles et trajectoires acoustiques. Application à la segmentation et à l'étiquetage automatique du signal de parole. Thèse de l'INPG, Grenoble.

Mehler, J. & Dupoux, E. (1990) *Naître humain*, Editions O. Jacob, Paris.
Meloni, H. & Rémi, B. (1987) Reconnaissance des formes et segmentation, *16èmes Journées d'Etudes sur la Parole*, 208-212.
Piterman, M. & Caelen, J. (1992) Modélisation, par un système dynamique, de trajectoires acoustiques unidimensionnelles. Actes des journées d'étude sur la parole, SFA-GCP, Bruxelles, 177-182.
Poggio T. (1984) Low-level vision as inverse optic, *Symposium : Computational model of hearing and vision*, Tallinn, 123-127.
Sakoe, H. & Chiba, S. (1978) Dynamic programming algorithm optimization for spoken word recognition, *IEEE Trans. on Acoust., Speech and Sig. Proc.*, 26-1, 43-49.
Sejnowski, T.J. & Rosenberg, C.R. (1987) Parallel networks that learn to pronounce english text, *Complex Systems*, 1, 145-168.
Sternberg, S., Wright, C.E., Knoll, R.L. & Monsell, S. (1980) Motor programs in rapid speech: additional evidence. In *The perception and production of fluent speech* (Cole, R.A., Editor). Hillsdale, NJ; Lea.
Stetson, R.H. (1928) Motor phonetics. A study of speech movements in action. In *The Netherlands Archives of Experimental Phonetics* (Kelso, J.A.S. & Munhall, K.G., Editors), Little, Brown & Cie; Boston, 3, 1-216.
Tattegrain, H. (1990) *Un système expert pour le décodage acoustico-phonétique de la parole*, Thèse de doctorat, INP, Grenoble.
Touzet, C. (1990) Contribution à l'étude et au développement de modèles connexionnistes séquentiels, Thèse de doctorat, Université de Montpelier II, 111p.
Vigouroux, N., Caelen, J. (1985) Segmentation en vue de l'organisation d'une base de données acoustique et phonétique. Actes des 14èmes Journées d'Etude sur la Parole, Paris, 152-155.
Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. & Lang, K. (1986) Phoneme recognition using time-delay neural networks, *Technical report TR-1-006*, ATR Interpreting Telephony Research Labs.
Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. & Lang, K. (1988) Phoneme recognition: neural networks vs. hidden markov models, *IEEE Conf. on Acoust., Speech & Sig. Proc.*, 107-110.
Watrous, R.L., Ladendorf, B. & Kuhn, G. (1990) Complete gradient optimization of a recurrent network applied to /b/, /d/, /g/ discrimination, *J. Acoust. Soc. Am.*, 87-3, 1301-1309.
Wellekens, C.J. (1986) Global connected digit recognition using Baum-Welch algorithm, *IEEE Conf. on Acoust., Speech & Sig. Proc.*, 1081-1084.
Whalen, D.H. (1990) Coarticulation is largely planned, *Journal of Phonetics*, 18, 3-35.
Williams, R.J. & Zipser, D. (1989) A learning algorithm for continually runing fully recurrent neural networks, *Neural Computation*, 1, 270-280.
Zue, V. (1986) The role od analysis by synthesis in phonetic Recognition. Proc. of Montreal Symposium on Speech Recognition, McGill University, 69-71.
Zue, V., Glass, J., Phillips, M., Seneff, S. (1989) Acoustic segmentation and phonetic classification in the Summit system. *IEEE*, proc. ICASSP.Baker, J. (1975)

Reconnaissance et compréhension de la parole

Caelen, J., Nasri, M.K., Reynier, E., Tattegrain, H. (1990). Architecture et fonctionnement du système DIRA. De l'acoustique aux niveaux inguistiques. Revue TS, vol. 7 N° 4, p. 345-366.
Carbonell, N., Damestoy, J.P., Fohr, D., Haton, J.P., Lonchamp, F. (1986a). "Design and implementation of an acoustic-phonetic decoding expert system". IEEE-ICASSP, Tokyo, Japan.
Carbonell, N., Pierrel, J.M. (1986b). "Architecture and knowledge sources in human computer oral dialogue system". Proceedings of NATO Workshop, Corsica, France.
Carbonell, N., Pierrel, J.M. (1987). "Task-oriented dialogue processing in human-computer voice comunication". NATO ASI Series, VOL 46.
Chomsky, N. (1965). "Aspect of the theory of syntax". MIT Press, Cambridge, Mass., USA.
Chow, Y. et al (1987). "BYBLOS: The BBN continuous speech recognition system". IEEE ICASSP.
Goldberg, H. G. and Reddy, R. (1976). "Feature extraction, segmentation and labeling in the Harpy and Hearsay-II systems". J. Acoust. Soc. Am. Vol. 60, S11 (A).
Haton, J. P. (1985). "Intelligence artificielle en compréhension automatique de la parole: état des recherches et comparaison avec la vision par ordinateur". TSI, Vol. 4-3, pp. 265-287.
Klatt, D. H. (1975). "Word verification in speech understanding system". pp. 321-341 in Reddy ed.
Klatt, D. H. (1977). "Review of the ARPA speech understanding project". J.A.S.A., Vol. 62, No. 6, December .
Klovstad, J. W., and Mondshein, L. F. (1975). "The CASPERS linguistic analysis system". IEEE Trans. Audio

Electroacoustic AU-16, 184-197.
Kubala, F. et al (1988). "Continuous speech recognition results of the BYBLOS system on the DARPA 1000-word resource management database". IEEE ICASSP pp. 291-294.
Lee, K-F. (1988). "Large vocabulary speaker-independent continuous speech recognition : The SPHINX system". Ph.D. dissertation, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
Lesser, V. R., Fennel, R. D., Erman, L. D., and Reddy, D. R. (1975). "Organization of the Hearsay-II speech understanding system". IEEE Trans. Acoust. Speech Signal Process. ASSP-23, pp.11-23.
Lesser, V. R. and Erman, L. D. (1977). "A retrospective view of the Hearsay-II architecture". IJCAI-77.
Lowerre, B. T. (1976). "The Harpy speech recognition system". Ph.D. thesis (departement of computer science, Carnegie-Melon University)
Mariani, J. (1982). "The ESOPE continuous speech understanding system". IEEE-ICASSP, pp. 1637-1640.
Mercier, G., Quinton, P., Vives, R. (1977). "Dialogue homme-machine avec KEAL". Recherches Acoustiques, Vol. 4, pp. 187-206.
Mercier, G. et al (1980). "The KEAL speech understanding system". In Spoken Language Generation and Understanding, J.C. Simon, editor, D. Reidel.
Morris, C. W. (1938). "Foundations of the theory of signs". Chicago, USA.
Newell, A., Barnett, J., Forgie, J.W., Green, C.C., Klatt, D.H., Licklider, J.C.R., Munson, J., Reddy, D.R. et Woods, W.A. (1973). "Speech understanding systems". NORTH-HOLLAND Press, Amsterdam.
Perennou, G. (1980). "ARIAL II: System for speech recognition". 5th IJCPR, Miami, December.
Pierrel, J. M. (1981). "Etude et mise en oeuvre de contraintes linguistiques en compréhension automatique du discours continu". Thèse d'Etat, Université de Nancy 1.
Pierrel, J. M. (1982). "Utilisation des contraintes linguistique en compréhension automatique de la parole continue". TSI, vol. 1, n°5, pp. 403-421.
Reedy, D. R., Erman, L. D., and Neely, R. B. (1973). "A model and a system for machine recognition of speech". IEEE Trans. AU-21, 229-238.
Reddy, D. R., Erman, L.D. (1975). "Tutorial on system organization for speech understanding". Acdemic Press, New York, pp. 457-459.
Ritea, B. (1975). Automatic speech understanding systems". Proceedings of the 11th IEEE Computer society Conference, Washington, DC, pp. 319-322.
Sheryl, R. Y., Alexander, G. H., Wayne, H. W., Edward, T. S. and Philippe, W. (1989). "High level knowledge sources in usable speech recognition systems". Communications of the ACM, Vol 32, No. 2, Feb.
Woods, W. A. (1970). "Transition network grammars for natural language analysis". Commun. Assoc. Comput. 13, pp. 561-602.
Woods, W., Bates, M., Brown, G., Bruce, B., Cook, C., Klovstad, J., Makoul, J., Nash-Webber,B., Schwartz,R., Wolf, J., and Zue, V. (1976).. "Speech understanding systems : final technical progress report". Bolt Beranek and Newman, Inc. Cambridge.
Woods, W. A. & Zue, V. (1976). "Dictionary expansion via phonological rules for a speech understanding system". pp. 561-564 in Teacher.

Modèles stochastiques pour la reconnaissance automatique de la parole

1. Introduction

Pour les systèmes de reconnaissance automatique de la parole (RAP), une de directions de recherche les plus importantes d'aujourd'hui se base sur les modèles stochastiques de langage. La RAP basée sur les modèles stochastiques est réalisée par un décodage probabiliste qui consiste à choisir dans la multitude des événements linguistiques possibles, celui qui correspond aux données avec la meilleure probabilité. Cela implique que le modèle doit pouvoir, pour chaque événement possible, attribuer un score à l'hypothèse qu'un événement linguistique a généré les données observées.

Nous nous proposons ici de présenter les principes fondamentaux des chaînes Markov, de modèles n-gramme ainsi que les algorithmes de type Viterbi et Baum-Welch qui peuvent être utilisés dans certaines étapes de la reconnaissance.

2. Modèles HMM pour la RAP

En général, une chaîne de Markov réside en un ensemble d'états et des transitions entre les états. Chaque état correspond à un symbole, et pour chaque transition il y a une probabilité associée. À la sortie d'une chaîne Markov on produit des symboles par des transitions entre les états. Les modèles de ce type peuvent être utilisés pour étudier les phénomènes avec des symboles observés arrangés en séries temporelles. Pour la reconnaissance automatique de la parole, ce modèle est trop restrictif et il doit être complété de sorte qu'il sera capable de traiter les cas où les observations sont des fonctions probabilistes des transitions. Ces modèles (avec l'extension mentionnée) sont connus comme des modèles Markov cachés (HMM – Hidden Markov Models). Pour les HMMs, les symboles de sortie sont probabilistes. Ça veut dire que tous les symboles sont possibles dans tous les états, chacun avec sa propre probabilité. A chaque état on associe une distribution de probabilité pour tous les symboles possibles. Par conséquent, un HMM est composé d'un processus non-observable (la chaîne Markov) et un processus observable qui lie les vecteurs acoustiques extraits du signal original avec les états de la chaîne Markov.

La figure 1 présente un HMM avec cinq états.

Ce modèle peut représenter une unité (phonème, mot etc.) et les transitions permises. Ce graphe peut être vu comme un modèle de production ans lequel chaque transition correspond à l'émission du signal (ou vecteur de paramètres). Pour chaque état s_j correspond une distribution de probabilité $P(e_k/s_j)$ (la probabilité de la production d'événement e_k quand il y avait lieu une transition de s_j). Chaque arc a une probabilité $a_{ij} = P(s_j/s_i)$, ça veut dire la probabilité de la transition d'état i vers l'état j.

Avec un HMM on peut modéliser des phonèmes, des mots ou une proposition complète. Pour le traitement de très grands vocabulaires, les HMM représentent des phonèmes, parce que de cette manière on réduit la quantité de données nécessaires pour l'entraînement et aussi l'espace de stockage pour modéliser les mots. Pour les petits vocabulaires, les HMM modélisent souvent des mots.

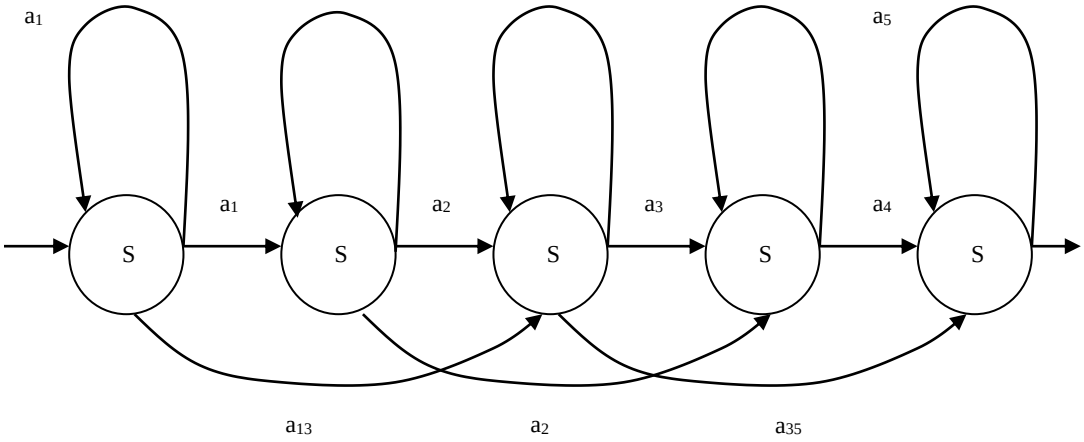

Figure 1

Pour tous les HMMs on doit résoudre trois problèmes :

- ***Le problème d'évaluation*** : étant donné un modèle et une séquence d'observations, quelle est la probabilité de ce modèle pour produire la bonne séquence. La solution la plus efficace est donnée par l'algorithme *forward-backward.*
- ***Le problème du décodage*** : étant donné un HMM et une séquence d'observations, quelle est la meilleure séquence d'états pour obtenir les observations respectives. C'est l'algorithme de Viterbi qui trouve la séquence d'états optimale.
- ***Le problème d'apprentissage*** : étant donné un HMM et une séquence d'observations (appelée aussi séquence d'entraînement), comment peut-on ajuster les paramètres du modèle pour maximiser la probabilité de production de la séquence. Une très bonne solution pour ce problème peut être l'algorithme Baum-Welch.

2.1. Le problème d'évaluation

Soit les notations :

- $O = (o_1, o_2, \ldots, o_T)$ la séquence d'observations.
- N le numéro d'états.
- a_{ij} probabilité de la transition d'état i vers l'état j.
- $b_i(o_t)$ probabilité que l'état i émet l'observation o_t.
- L l'ensemble de paramétres qui caracterisent le HMM (les probabilités de transitions, les probabilités d'émission et la distribution initiale des états).

Le meilleur algorithme qui trouve la probabilité de ce modèle pour produire la séquence respective (notée $P(O/L)$) est l'algorithme *forward-backward.*

Notons $a_t(i)$ la probabilité *avant*. Elle représente la probabilité d'observer les premiers t vecteurs quand on est dans l'état i et au moment t. On peut calculer cette probabilité avec la formule suivante :

$$a_{t+1}(j) = \left[\sum_{i+1}^{N} a_t(i) a_{ij} \right] b_j(o_{t+1})$$

$$1 \le t \le T-1$$

$$1 \le j \le N$$

Par conséquent, la probabilité *P(O/L)* sera donnée par :

$$P(O/L) = \sum_{i=1}^{N} a_T(i).$$

Pour un moment donné *t*, on calcule la variable *avant* pour tous les états. Puis, on fait les itérations pour *t = 1, 2, ..., T-1*. Des calculs similaires peuvent être réalisés en partant de l'état final vers l'état initial.

Exemple : Considérons le HMM illustré ci-dessous :

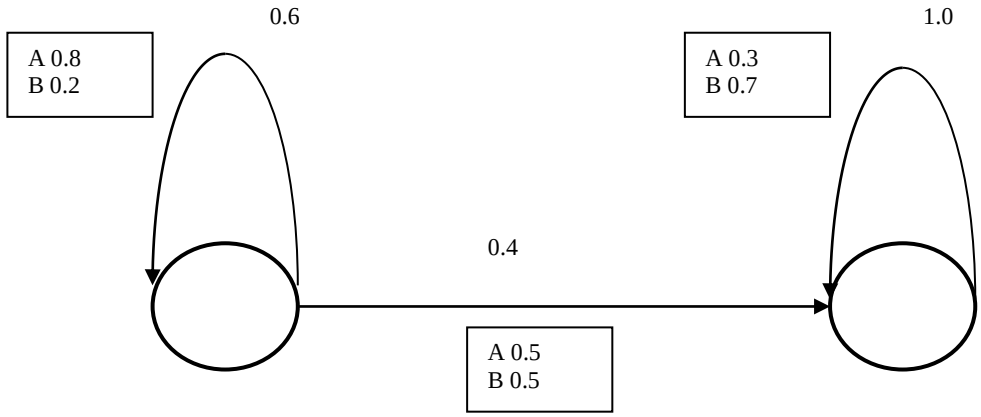

Figure 2

La figure présente les probabilités de produire les symboles A ou B quand une telle transition a lieu. On veut calculer la probabilité d'observation pour la séquence AAB an utilisant l'algorithme *forward*. On a illustré cette méthode par la figure 3 où chaque transition possible est représentée avec sa propre probabilité multipliée par la probabilité de production de symbole observé. On a aussi illustré les probabilités d'être dans un état en temps et qu'on produise le symbole donné. Ces probabilités ont été calculées (en accord avec l'algorithme *forward*) par sommer toutes les probabilités qui viennent vers un état, chacune multipliée par la probabilité de l'état précédente.

Dans notre cas, la probabilité avant P(O/L) est 0.19 (0.03 + 0.16, la somme entre les valeurs obtenues dans les états finales).

2.2. Le problème du décodage

Comme on a déjà annoncé, le problème du décodage peut être résolu en utilisant *l'algorithme de Viterbi*. Cette méthode trouve la meilleure séquence d'états pour obtenir les observations considérées. Pour calculer cette séquence optimale (pour les premières *t*

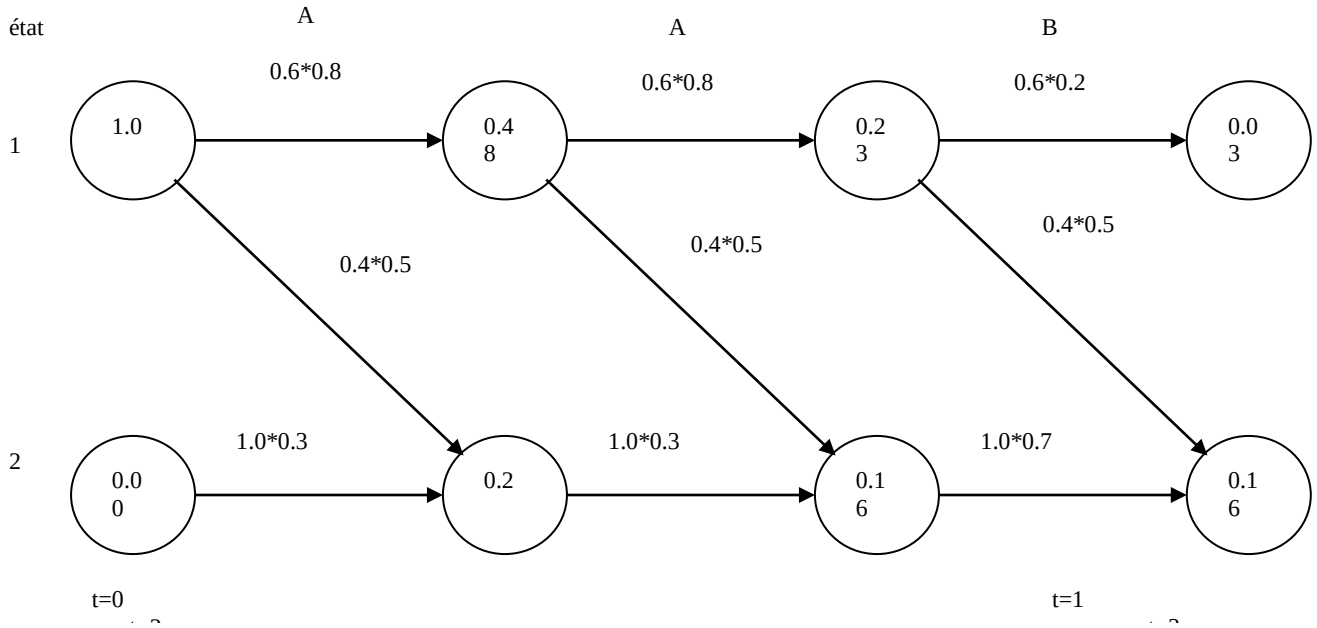

Figure 3

observations), on doit travailler avec $\delta_t(i)$ qui représente le meilleur score au moment *t* et en état *i* :

$$\delta_t(i) = \max_{q_1, q_2, ..., q_{t-1}} P\left[q_1, q_2, ..., q_{t-1}, q_t = i, o_1, o_2, ..., o_t \middle| L\right]$$

Le score $\delta_t(i)$ peut être calculé en utilisant la récursivité suivante :

$$\delta_t(j) = \left[\max_{\substack{1 \le i \le N \\ 2 \le t \le T}} \delta_{t-1}(i) a_{ij} \right] b_j(o_t) .$$

On peut exemplifier cette méthode sur le même exemple d'évaluation. Trouvons la meilleure séquence d'états pour obtenir la séquence AAB. L'algorithme de Viterbi est très similaire de l'algorithme forward. C'est à dire qu'au lieu de sommer les probabilités de tous les états au moment précèdent, on somme seulement la plus probable transition et on ignore le reste. En revenant depuis l'état final le plus probable, on trouvera la meilleure (la plus probable) séquence d'états. Dans la figure 4, la grosse ligne continue est la voie la plus probable. C'est qu'on appelle un **treillis temps-état**.

Comme on peut voir, pour cette situation la meilleure séquence d'états est 1 – 1 – 2 – 2. C'est important à remarquer que pour des séquences d'observations partiales, la meilleure séquence d'états n'est pas obligatoirement la même. Par exemple, pour générer AA, la meilleure séquence d'états sera 1 – 1 – 1.

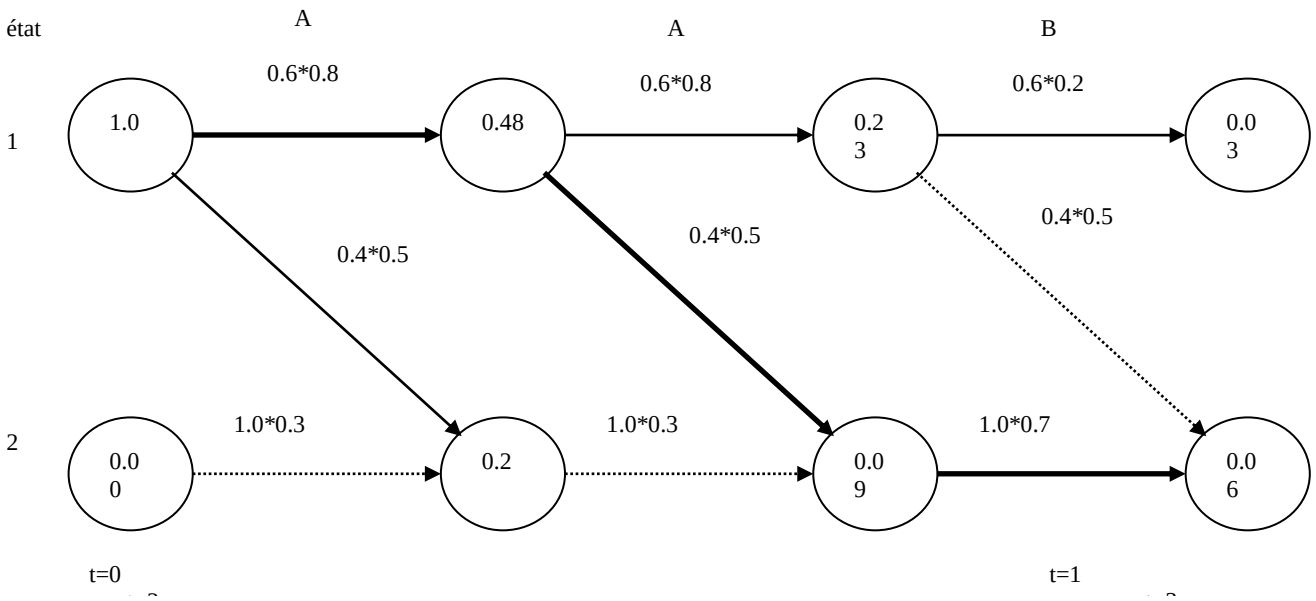

Figure 4

2.3. Le problème d'apprentissage

Le problème le plus difficile pour les HMM est comment ajuster les paramètres du modèle (L) pour obtenir la probabilité maximale de production d'une séquence donnée. On ne peut pas résoudre ce problème d'une manière analytique. On peut quand même choisir les paramètres de sorte qu'on maximise localement la probabilité en utilisant *l'algorithme Baum-Welch* (Baum-Welch re-estimation). Pour faire des commentaires sur cette méthode, il est nécessaire de définir une nouvelle variable :

$$\gamma_t(i,j) = P(i_t = q_i, i_{t+1} = q_j | O, L),$$

cela veut dire la probabilité de transition entre les états i et j, conditionnée par la séquence d'observations O et le modèle L. Par conséquent, le numéro prévu de transitions de l'état q_i peut être calculé selon la formule :

$$\sum_{t=1}^{T}\sum_{j=1}^{N}\gamma_t(i,j)$$

et le numéro prévu de transitions de l'état q_i vers l'état q_j :

$$\sum_{t=1}^{T}\gamma_t(i,j).$$

En utilisant les formules ci-dessus, on a été démontré qu'on peut recalculer (re-estimer) les paramètres du modèle. Donc, la probabilité de transition entre les états i et j sera donnée par la relation suivante :

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T} \gamma_t(i,j)}{\sum_{t=1}^{T}\sum_{j=1}^{N} \gamma_t(i,j)},$$

tandis que la probabilité d'émission aura la formule ci-dessous :

$$\overline{b}_j(k) = \frac{\sum_{t=1}^{T}\sum_{j=1}^{N} \gamma_t(i,j)}{\sum_{t=1}^{T}\sum_{j=1}^{N} \gamma_t(i,j)}.$$

Les dernières deux équations représentent les formules Baum-Welch.

En accord avec ces relations, on peut observer que les anciens paramètres sont remplacés par des nouveaux. On a démontré que ces nouvelles valeurs augmentent la probabilité de production de la séquence *O* (c'est à dire *P(O/L)*).

Supposons que S représente les valeurs actuelles. Les nouvelles valeurs S_1 conduisent à une meilleure probabilité *P(O/L)*.

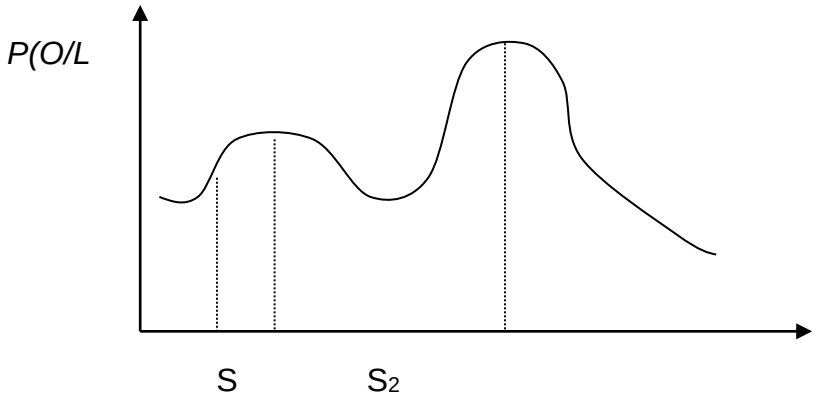

Figure 5

Mais il faut faire attention, parce qu'on peut arriver dans un maximum local avant d'arriver dans le maximum global.

2.4. Commentaires

En RAP, une observation correspond rarement à l'apparition d'un mot mais plutôt à une séquence de mots. Si le langage considéré est composé d'un ensemble limité de phrases, alors il est raisonnable d'avoir un modèle différent pour chaque apparition. Cependant, lorsque l'ensemble des phrases valides est important (où même infini), on doit considérer une autre approche. Un grand nombre de modèles

pose des problèmes à la fois pour l'apprentissage et pour la reconnaissance. Si le vocabulaire est composé de plusieurs centaines de mots et qu'un modèle spécifique est nécessaire pour chaque mot alors, afin d'entraîner l'ensemble des modèles il serait nécessaire d'avoir plusieurs occurrences de chaque mot dans le corpus d'entraînement. Il est vraiment impossible de collecter ces échantillons. De plus, avec cette approche, le langage de reconnaissance ne peut pas être étendu pour inclure les mots non observés parmi les données d'apprentissage. Ces problèmes peuvent être résolus en représentant les événements intéressants par la concaténation d'unités élémentaires issues d'une liste de taille gérable. Seulement ces unités auront des modèles spécifiques et les unités plus complexes seront alors modélisées par la composition (concaténation) des modèles des unités élémentaires. Cette structure convient particulièrement à la RAP basée sur des HMM pour quelques raisons :

- La production de la parole est en réalité basée sur l'émission d'une séquence d'éléments qui appartiennent à un ensemble limité : les phonèmes du langage.
- La structure du graphe des HMMs rend facile la construction de modèles composés.
- Si un ensemble complet de modèles est disponible, on peut construire un modèle pour chaque mot du langage, dès que sa transcription phonémique est connue.

La composition de modèle simplifie aussi l'apprentissage. Afin d'entraîner un modèle, on doit avoir un ensemble d'exemples disponible. Quand les unités élémentaires sont les phonèmes, il n'est pas possible de collecter des exemples isolés, puisque seuls les mots entiers ou des propositions peuvent être prononcés d'une manière naturelle. Donc, entraîner des modèles de phonèmes avec des données pertinentes nécessite la segmentation des données d'apprentissage, en spécifiant l'identité et la position de chaque phonème dans l'enregistrement. Donc, retenons qu'en utilisant la concaténation des modèles élémentaires, le traitement des données d'apprentissage est plus simple.

L'entraînement des HMMs consiste en une procédure itérative où, à chaque étape, les nouvelles valeurs des paramètres du modèle sont calculées en fonction des valeurs précédentes et des données d'apprentissage. Une itération de la procédure consiste en deux étapes :

1. Toutes les prononciations d'entraînement sont utilisées pour mettre à jour l'ensemble des valeurs intermédiaires, chacune correspondant à un élément du modèle.
2. Ces valeurs <<accumulées>> sont utilisées pour calculer un nouvel ensemble de paramètres. Cela permet d'utiliser la composition pour l'entraînement des modèles élémentaires pour des phrases en parole continue. Pour chaque prononciation d'entraînement, seule sa transcription phonémique doit être connue.

Donc, un HMM composé représentant toute la phrase peut être construit et les accumulateurs de ses constituants peuvent être mis à jour. Une fois les données d'apprentissage traitées, les paramètres de chaque modèle de phonème sont ré-estimés par l'exploitation de toutes les occurrences des phonèmes dans l'ensemble d'apprentissage.

3. Modèles de langage

Contrairement à la communication homme-homme, l'interaction homme-machine doit produire des instances de données structurées de façon déterministe. Le déterminisme, dans ce cas, signifie que le système informatique doit engendrer la même représentation toutes les fois qu'un même signal est traité. Les sources de connaissances utilisées par les machines sont des modèles de celles utilisées par l'homme pour produire ses messages. L'une de ces connaissances est le *modèle de langage*.

Les systèmes modernes sont basés sur des scores probabilistes attribués aux candidats hypothétiques. Un modèle probabiliste simple pour évaluer les hypothèses est représenté dans la figure 6.

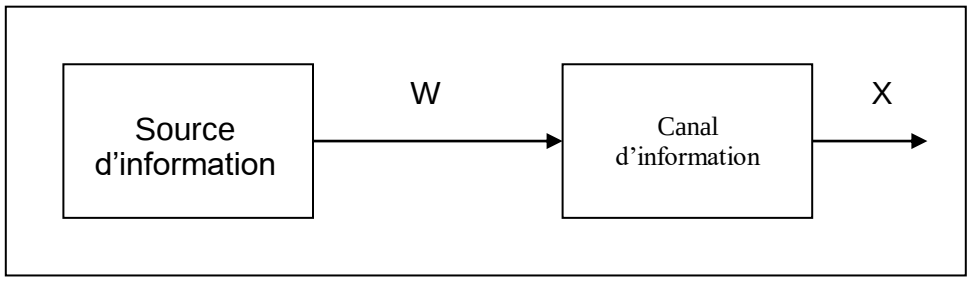

Figure 6

Il contient un décodeur qui considère une séquence d'observations acoustiques comme la sortie d'un canal d'information qui reçoit en entrée une séquence de symboles représentant l'intention du locuteur. Si ces symboles sont des mots, alors ils sont couramment représentés par la séquence

$$W = w_1 w_2 \ldots w_n.$$

Le signal *X* est la version codée de *W*. Comme on a déjà vu, le but de la reconnaissance est de reconstruire *W* en se basant sur l'observation de *X*.

On note :

- *P(W)* la probabilité d'une séquence de mots ;
- *P(X/W)* la probabilité d'observer X quand on a prononcé W,

on doit mentionner qu'en pratique, *P(X/W)* ne peut pas être évaluée directement à partir des données. On aura besoin d'un modèle acoustique, tandis que pour évaluer *P(W)* on aura besoin d'un modèle de langage.

En général, on peut calcule la probabilité d'une séquence de mots avec la formule :

$$P(W) = P(w_1, w_2, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_0, \ldots, w_{i-1})$$

où w_0 est choisi d'une manière appropriée pour la condition initiale. On peut observer que la probabilité de mot w_i dépend d'histoire donnée par les mots qui ont été déjà prononcés. On va noter cette histoire h_i. En raison de cette factorisation, la complexité du modèle augmente de sorte exponentielle avec la longueur d'histoire. C'est pour ça qu'en pratique on utilise des versions simplifiées, c'est à dire qu'on prend seulement une partie d'histoire pour influencer la probabilité d'un mot suivant. Il s'agit de modèles n-gramme. Les plus utilisés modèles sont les bigrammes (n=2) et trigrammes (n=3), dans lesquels les plus récents 1 respectivement 2 mots sont utilisés pour conditionner la probabilité du mot suivant.

3.1. Le modèle bigramme

Comme on a déjà mentionné, la probabilité d'une séquence de mots sera dans ce cas :

$$P(W) = \prod_{i=1}^{N} P(w_i | w_{i-1}) .$$

Considérons un modèle bigramme (illustré dans la figure 7). Ce modèle peut être vu comme un automate fini. Les arcs discontinus correspondent aux transitions entre différents mots en accord avec les probabilités du modèle. On peut observer que si on remplace les arcs avec les HMMs correspondants, on obtiendra un HMM composé dans lequel on peut calculer le chemin plus probable en utilisant l'algorithme de Viterbi. Les arcs discontinus n'auront pas des transitions de sortie associées. Il faut noter que la solution donnée par cet algorithme n'est pas optimale. C'est à dire qu'elle donne la probabilité d'une seule séquence d'états, et pas la probabilité totale d'émission de la meilleure séquence. Quand même, en pratique on a été observé que la probabilité du chemin donnée par cette méthode est tout près d'une probabilité totale.

Il faut aussi remarquer que la dimension du modèle augmente en fonction de vocabulaire. Cela peut conduire vers un très grand espace de recherche et ensuite vers un très long temps de traitement.

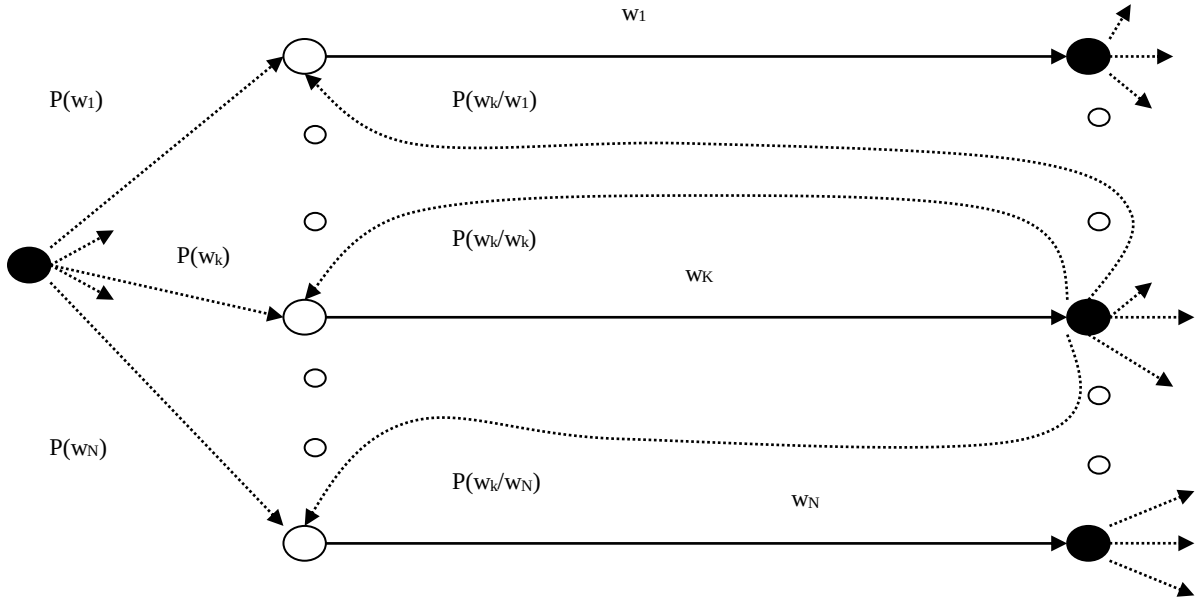

Figure 7

Quand on a un grand nombre d'états, à chaque instant, plusieurs états auront une probabilité accumulée beaucoup diminuée par rapport à la meilleure. On peut dire qu'en ces conditions, c'est peu probable qu'un chemin qui passe par des états comme ceux mentionnés peut devenir le meilleur chemin à la fin de la séquence. Ces observations conduisent vers une technique de réduction la complexité connue comme *beam search*. Cette méthode suppose qu'on peut négliger les états avec des scores accumulés sous le meilleur minus un seuil donné. De cette manière, on doit éviter les nœuds avec petits scores pendant la procédure. Il faut être mentionné que cette technique peut (en théorie) perdre le meilleur chemin, Mais en pratique, on a été démontré que si on choit correctement le seuil, on aura un gain en temps de 10 fois, tandis que les erreurs introduites seront complètement négligeables.

Quand le vocabulaire est très grand (dizaines des milles de mots) on doit considérer des différentes approches. Actuellement, il y a une variété d'algorithmes destinés au traitement de très grands vocabulaires. La plupart d'eux utilisent les techniques *multi-pass* (passages multiples). Chaque passage prépare les informations nécessaires pour le prochain, réduisant de cette manière l'espace de la recherche.

3.2. Le modèle trigramme

Dans ce cas, la probabilité sera calculée selon la relation suivante :

$$P(W) = \prod_{i=1}^{N} P(w_i | w_{i-2}, w_{i-1}).$$

Pour estimer les probabilités d'un trigramme, on peut utiliser un grand corpus des données (le corpus d'entraînement) pour calculer les fréquences de trigramme :

$$f_3(w_3 | w_1, w_2) = \frac{c_{123}}{c_{12}},$$

où c_{123} est le nombre de fois qu'on observe la séquence $\{w_1,w_2,w_3\}$ et c_{12} le nombre de fois qu'on observe la séquence $\{w_1,w_2\}$. Pour un vocabulaire de dimension V, on aura V^3 trigrammes possibles. Un nombre significatif de trigrammes ne sera pas vu dans le corpus d'entraînement, c'est à dire que ces trigrammes auront probabilité nulle. Pour résoudre ce problème, on a besoin d'estimer les probabilités des événements qui ne sont pas vus. On peut faire ça par une interpolation linéaire des fréquences de trigrammes, bigrammes et unigrammes et aussi par une distribution uniforme de vocabulaire.

$$P(w_3 | w_1, w_2) = \lambda_3 f_3(w_3 | w_1, w_2) + \lambda_2 f_2(w_3 | w_2) + \lambda_1 f_1(w_3) + \lambda_0 \frac{1}{V},$$

où $f_2( )$ et $f_1( )$sont estimés comme $f_3( )$ mais en tenant compte de bigrammes et unigrammes. Comme on verra dans les paragraphes suivants, les poids λ_i peuvent être estimés dans différentes manières, par exemple interpolation rationnelle ou la maximisation de probabilité de données *held-out* (qui sont différentes en rapport avec les données qui ont été utilisées pour estimer les probabilités pour les n-grammes). Il s'agit donc d'un problème d'estimation ML (Maximum Likelihood).

3.3 La perplexité

La perplexité est une mesure qui permet l'évaluation d'un modèle de langage. Quand on a deux modèles de langage, on veut les comparer. Une méthode peut être leur utilisation dans un système de reconnaissance. Ça sera toujours la meilleure méthode, mais malheureusement, en même temps trop chère. Pour éviter cette approche, on peut utiliser la théorie d'information, plus précisément, la valeur d'entropie calculée pour un nouveau texte, mais qui n'a pas été utilisé pour la construction du modèle de langage.

Notons *p* la vraie probabilité (qui n'est pas connue) d'un segment de *x* mots d'un nouveau texte de *k* mots. En ces conditions, la valeur d'entropie sera donnée par la relation suivante :

$$H = \lim_{n\to\infty} -\frac{1}{k}\sum_{x} p(x)\log_2 p(x).$$

On doit observer que si tous les mots d'un vocabulaire de dimension *V* ont la même probabilité, l'entropie sera

$$H = \log_2 V.$$

Pour n'importe quelle autre distribution de probabilité pour les mots qui composent le vocabulaire, l'entropie, satisfera l'inégalité :

$$H \le \log_2 V.$$

Pour calculer la probabilité du segment du texte choisi, on va utiliser la probabilité du modèle qui existe et qui sera notée. Notons que cette valeur diffère de celle notée *p* qui est inconnue et qui correspond au vrai modèle. On va calculer ensuite la valeur de la probabilité logarithmique (*logprob*) moyenne qui est définie selon la relation suivante :

$$lp_k = -\frac{1}{k}\sum_{i=1}^{k} \log_2 \bar{p}(w_i|h_i).$$

Il a été démontré que si on fait $k \to \infty$ on obtiendra l'inégalité ci-dessous :

$$\lim_{k\to\infty} lp_k = lp \ge H,$$

cela veut dire que la probabilité logarithmique moyenne ne peut pas être sous la valeur d'entropie. Le but de ces calculs est de découvrir quel modèle de langage a une valeur *logprob* moyenne plus près de la valeur d'entropie du texte.

Pour évaluer un modèle de langage, on utilise souvent la mesure dénommée *perplexité*. Elle est donnée par la formule :

$$T = 2^{lp}.$$

On peut dire que la perplexité est la mesure de la dimension d'un ensemble des mots de lequel on choit le mot prochain, étant donné qu'on observe l'histoire des mots prononcés. La perplexité d'un modèle de langage est dépendante de quel domaine on utilise pour choisir le texte. Par exemple, si on parle sur le domaine médical de langue anglaise, on obtiendra une perplexité de 60, pour le domaine de journalisme 105, tandis que si on prend la langue anglaise en général, on aura une valeur 247, ce qui correspond à une valeur d'entropie de 1.75bit /caractère.

3.4 Nouvelle approche pour les modèles n-gramme

Une nouvelle méthode a été proposée en 1996 pour améliorer les performances d'un système de reconnaissance. Cette approche a eu comme résultat aussi la réduction de la perplexité. Cette nouvelle approche essaie de surmonter les problèmes qui ont apparu dans les modèles classiques et qui sont causés par le dénombrement des unités (phonèmes, mots, phrases) dans le corpus d'entraînement. Normalement, toutes les apparitions d'une séquence de mots dans un corpus sont comptées comme des apparaissons d'une certaine phrase (qui les contient). Quand il y a des phrases plus fréquentes que leurs sub-phrases, on peut obtenir une réduction de performance pendant les tests qui travaillent sur les sub-phrases respectives (sans être dans le contexte de la phrase entière).

Par exemple, pour un modèle stochastique standard, quand une phrase "A B C" est très fréquente dans un corpus d'entraînement les probabilités estimées pour "C" (dans les situations quand "C" suivie une

phrase de type ”X B”) peuvent être artificiellement augmentées. C’est parce que ”B C” a une fréquence trop haute. Supposons qu’on a prononcé dix fois la proposition ”Les élèves sont allés à l’école”. Toutes les sub-unités auront la même fréquence. Donc, quand on estimera les probabilités pour les mots qui peuvent suivre la séquence ”sont allés”, on trouvera ”à l’école”, parce que ”à l’école” a reçu une trop grande valeur de probabilité.

Pour surmonter ce problème, une nouvelle base de données pour les fréquences des n-grammes a été créée. Dans cette base de données, les sub-phrases d’une phrase fréquente aura une probabilité réduite à 1 pour chaque apparition de la phrase fréquente. Donc on aura une probabilité additionnelle qui pourra être englobée dans le modèle de langage pour créer des trigrammes ou des modèles d’ordre supérieur.

Quand un mot ou une séquence des mots fait partie en même temps à deux phrases fréquentes, sa fréquence est réduite une fois, et pas une fois pour chaque phrase. Par exemple, si ”A B C” et ”B C D” sont des phrases fréquentes dans le corpus d’entraînement, la fréquence de la sub-phrase ”B C” sera normalement réduite une fois pour chaque apparition en ”A B C” et une fois pour chaque apparition en ”B C D”. Et encore, si la phrase ”A B C D” existe une fois dans le corpus d’entraînement, comme elle représente des apparitions pour ”A B C” et ”B C D”, la fréquence de ”B C” sera réduite deux fois et pas une fois. C’est pour ça qu’il est nécessaire de rajouter une apparition de ”B C” dans la base de donné.

De cette manière, on a obtenu une réduction de la perplexité de 4.9%. Il faut remarquer qu’il est besoin d’un temps de traitement assez long pour créer la nouvelle base de données.

3.5 L’interpolation rationnelle des prédicteurs d’un modèle stochastique de langage

Dans le cas des trigrammes, on peut exprimer la probabilité, selon la relation ci-dessous :

$$P(w_3|w_1,w_2) = \lambda_3 f_3(w_3|w_1,w_2) + \lambda_2 f_2(w_3|w_2) + \lambda_1 f_1(w_3) + \lambda_0 \frac{1}{V}.$$

Notons premièrement *v* l’histoire du mot. Dans ce cas on peut récrire *P* la probabilité avec la formule suivante :

$$P(w/v) = \sum_{i\in I} \lambda_i f_i(w/v).$$

Introduisons maintenant une fonction $g_i(v)$ dépendante d’histoire et qui sera utilisée pour obtenir un score concernant la qualité des prédicteurs et qui se basera sur l’histoire pour laquelle f_i est pertinente. Une interpolation linéaire donne la formule :

$$P'(w/v) = \sum_{i\in I} \lambda_i g_i(v) f_i(w/v).$$

Mais P’(w/v) n’est pas encore une distribution de probabilité, parce qu’elle ne respecte pas les conditions de normalité. On va faire une normalisation par le facteur :

$$P''(v) = \sum_{w\in V} P'(w/v) = \sum_{i\in I} \lambda_i g_i(v)$$

et on va obtenir le modèle d’interpolation rationnelle :

$$P(w/v) = \frac{P'(w/v)}{P''(v)} = \frac{\sum_{i\in I} \lambda_i g_i(v) f_i(w/v)}{\sum_{i\in I} \lambda_i g_i(v)}.$$

Supposons que l'optimisation des coefficients λ_i utilise un ensemble de données de validation $w=w_1 \ldots w_S$. Notons v_s l'histoire du mot w_s, $s=1 \ldots S$. Le vecteur des coefficients $\boldsymbol{\lambda}$ va être calculé par la maximisation d'une fonction logarithmique de probabilité, décrite par la formule suivante :

$$l_w(\lambda) = \log P(w) == \log \prod_{s=1}^{S} P(w_s / w_1^{s-1}),$$

où $w_1^{s-1} = w_1 \ldots w_{s-1}$. La condition de normalité :

$$\sum_{i\in I} \lambda_i = 1.$$

Les valeurs λ_i peuvent être optimisées en utilisant le gradient $\nabla_l(\boldsymbol{\lambda})$ avec des composantes données par la relation suivante :

$$\frac{\partial l}{\partial \lambda_i} = \sum_{s=1}^{S} \left\{ \frac{g_i(v_s) f_i(w_s / v_s)}{P'(w_s / v_s)} - \frac{g_i(v_s)}{P''(v_s)} \right\}.$$

La matrice Hessienne $\boldsymbol{H = H(\lambda)}$, ayant des éléments :

$$H_{ij} = \frac{\partial^2 l}{\partial \lambda_i \partial \lambda_j}$$

a une forme $\boldsymbol{H = H' - H''}$ et les éléments de ces deux matrices sont :

$$H'_{ij} = \sum_{s=1}^{S} \frac{g_i(v_s) f_i(w_s / v_s) g_j(v_s) f_j(w_s / v_s)}{(P'(w_s / v_s))^2}$$

$$H''_{ij} = \sum_{s=1}^{S} \frac{g_i(v_s) g_j(v_s)}{(P''(v_s))^2}$$

.

Ces éléments sont symétriques et positive-définis, tandis que $\boldsymbol{H(\lambda)}$ ne l'est pas et par conséquent on ne peut pas appliquer ici l'itération de type Newton

$$\lambda^{(k+1)} = \lambda^{(k)} + (H(\lambda^{(k)}))^{-1} \nabla_l(\lambda^{(k)}),$$

parce qu'il n'est plus garanti que la fonction de probabilité l_w va augmenter. C'est pour cela qu'on a utilisé $\boldsymbol{H'}$ au lieu de $\boldsymbol{H}$ pour obtenir une croissance monotone de l_w.

En ce qui concerne les fonctions g_i, elles peuvent être choisies en partant de f_i. Comme nous avons déjà montré, nous pouvons écrire f_i de cette manière :

$$f_i(w/v) = \frac{c_i(v,w)}{c_i(v)},$$

où c_i est le nombre d'apparitions de la séquence qui est entre parenthèses. La fonction g_i a été choisie comme ci-dessous :

$$g_i(v) = \frac{c_i(v)}{c_i(v) + C}, \text{ avec } C > 0$$

et ensuite, le modèle d'interpolation rationnelle a été obtenu (voir la relation suivante. Il faut remarquer aussi que les valeurs $g_i(v)$ varient dans l'intervalle (0, 1), ce qui a été illustré dans la figure 8. Pour $C \rightarrow 0$, on obtient $g_i(v)=1$, ce qui correspond à l'interpolation linéaire).

$$P(w/v) = \frac{\sum_{i \in I} \lambda_i \frac{c_i(v,w)}{c_i(v)+C}}{\sum_{i \in I} \lambda_i \frac{c_i(v)}{c_i(v)+C}} .$$

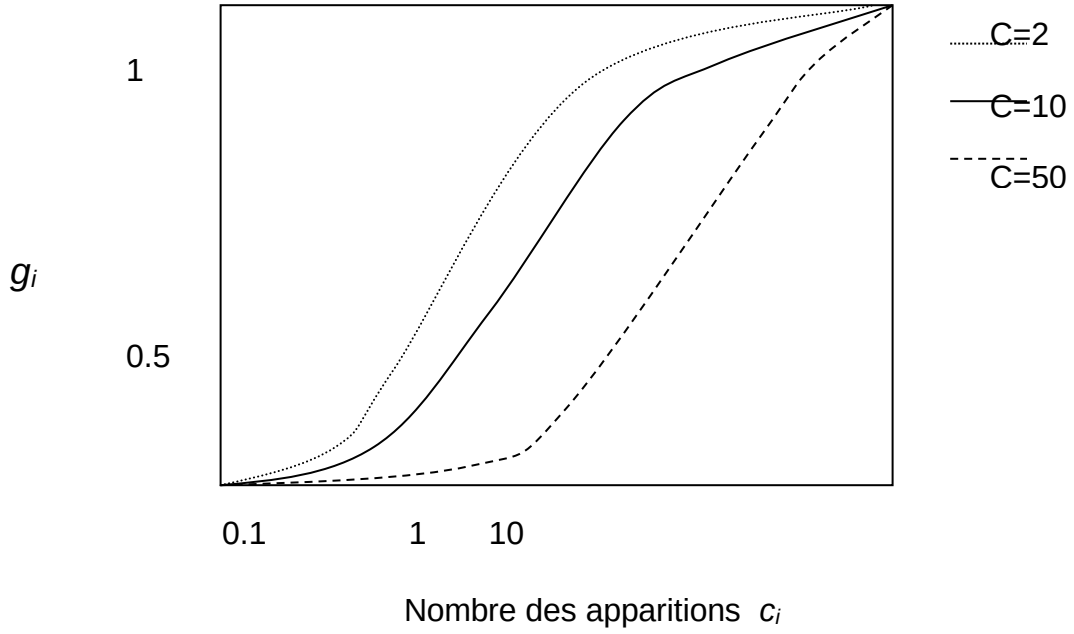

Figure 8

Avec ce nouveau modèle d'interpolation, on a obtenu des très bons résultats pendant le processus de reconnaissance, ainsi qu'en ce qui concerne la perplexité.

4. Le problème du nouveau mot

Même quand on utilise un très grand vocabulaire dans un système de reconnaissance automatique de la parole il n'est pas possible de couvrir tous les mots qu'un utilisateur ou un groupe d'utilisateurs prononce. Quand même, dans les systèmes actuels, quand on prononce un mot hors vocabulaire, un certain mot de vocabulaire existant sera extrait, c'est à dire une faute reconnaissance. De plus, on aura aussi des fausses reconnaissances dans la région avoisinante de ce nouveau mot.

Comme on peut voir dans la figure 9, ce problème peut être abordé en deux étapes :

- Détecter qu'un nouveau mot a été prononcé ;
- Ajouter le nouveau mot au vocabulaire déjà existant.

Ces deux étapes peuvent être résolues d'une manière indépendante. Pour détecter des nouveaux mots, il n'y a pas beaucoup de méthodes proposées. L'addition de nouveaux mots dépend de quel type de système de reconnaissance on utilise. Quand le système se base sur des mots entiers, une possibilité d'addition peut être l'entraînement du système avec les nouveaux mots. Mais quand on utilise un système basé sur des représentations phonétiques, l'addition aura besoin de prononciations. On a utilisé la transcription orthographique pour construire la représentation phonétique de nouveau mot. Par

exemple, en 1991, *Asadi et al.* Ils ont appliqué les règles d'un système texte-parole pour obtenir les transcriptions phonétiques de nouveau mot. Quand même, les transcriptions

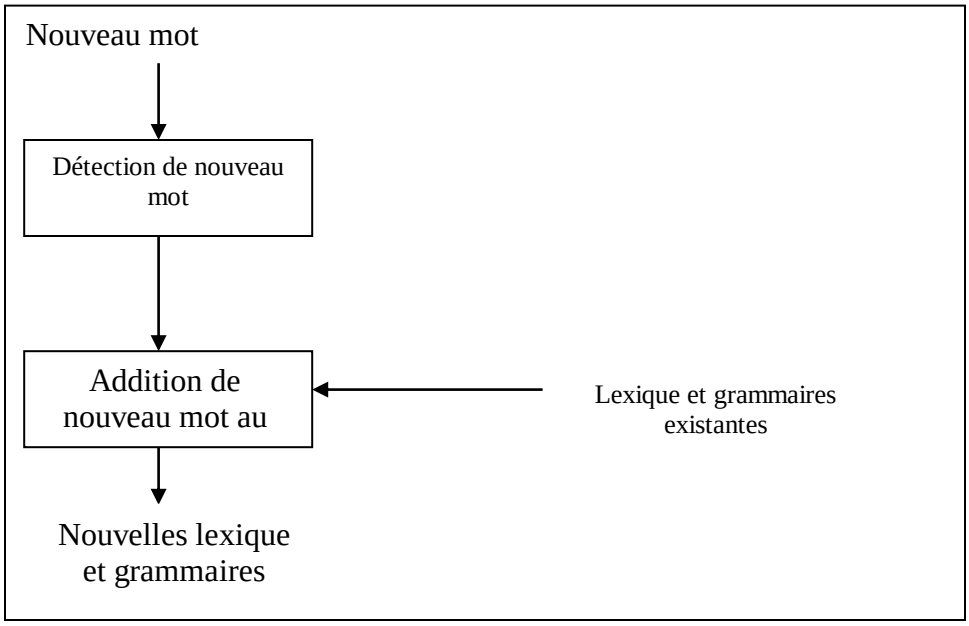

Figure 9

obtenues contenaient des fautes. C'est pour cela qu'un réseau de prononciations a été créé, ensuite, le locuteur a prononcé le nouveau mot et un système de reconnaissance phonétique a choisi la meilleure transcription phonétique. Donc, on peut dire qu'en cette approche, on a besoin de la transcription du mot. D'autres approches essayent d'extraire la transcription du nouveau mot de sa transcription phonétique (en supposant qu'on a un système qui peut extraire des transcriptions phonétiques).

4.1 Quelques approches pour modeler un nouveau mot

Les suppositions qui ont représenté la base de ces modèles sont les suivantes :

- Le modèle doit être général, c'est à dire qu'il doit représenter n'importe quel nouveau mot apparaît.
- Le modèle doit avoir un score meilleur sur les nouveaux mots par rapport aux mots déjà existants dans le vocabulaire.

Tous les modèles (proposés par Asadi, Schwartz et Makhoul) consistent en une séquence des phonèmes. Chaque phonème est représenté par une Chaîne Markov Caché (HMM) ayant trois états. Les états sont connectés de gauche à droit et ils possèdent des transitions revenant dans le même état. Il y a évidemment des probabilités associées à chaque transition. Le premier modèle a quatre phonèmes cinq états. Les quatre phonèmes sont identiques et ils ont une distribution spectrale plate et réglable (voir figure 10). Appelons ce modèle M1.

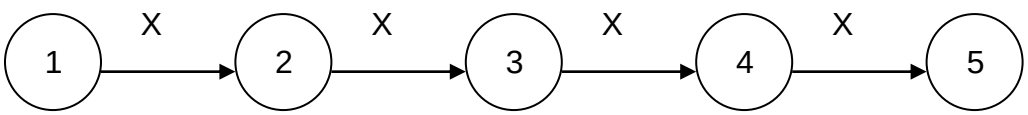

Figure 10

Le deuxième modèle est une approche qui accepte n'importe quelle séquence de phonèmes mais qui contient au moins deux phonèmes (voir figure 11). Le modèle a trois états et les transitions illustrées dans la figure 11. Il y a donc 3N arcs, où N est le nombre des phonèmes qui ont été utilisés. Nous allons appeler ce modèle comme M2.

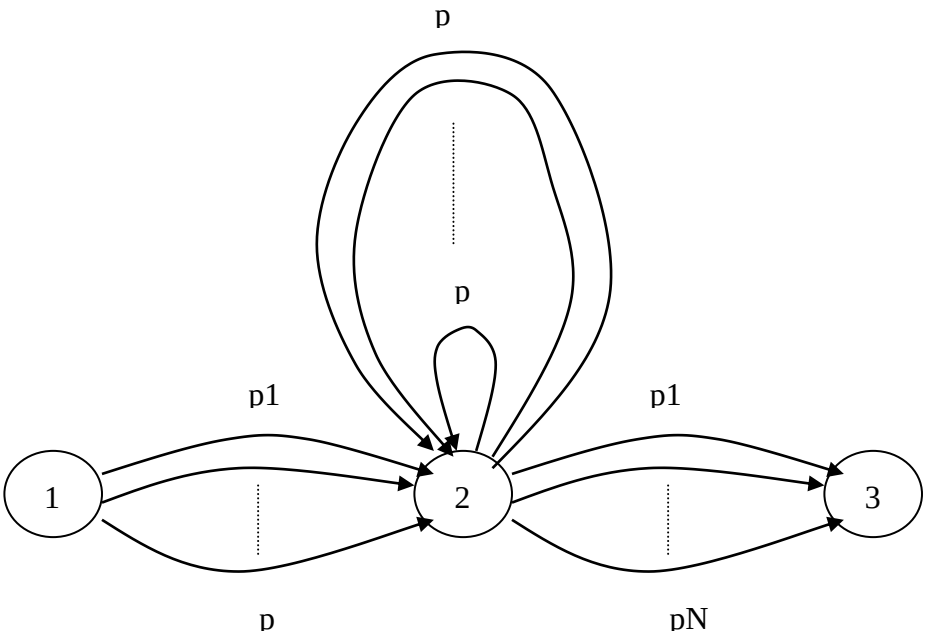

Figure 11

Le troisième modèle ressemble au deuxième. C'est à dire qu'on a cinq états, 5N arcs (représentant les transitions) et il accepte au moins quatre phonèmes. Ce modèle sera ensuite M3 (voir figure 12).

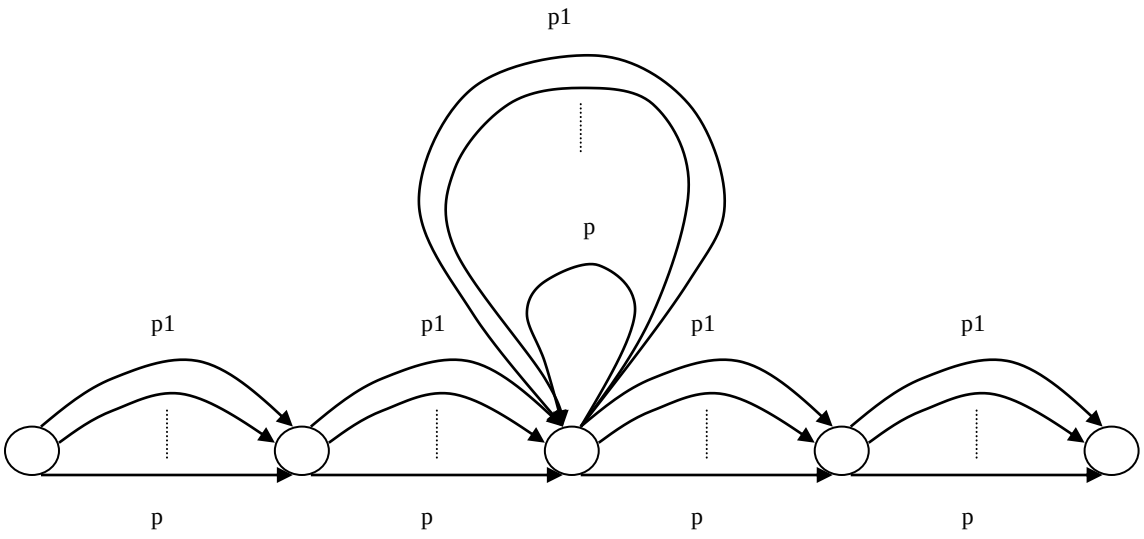

Figure 12

Le quatrième modèle est composé par des modèles des diphones. Il accepte n'importe quelle séquence ayant au moins deux diphones. Etant donné N (le nombre de phonèmes utilisés), il y a N+2 états (+2 parce qu'on modèle aussi les frontières initiales et finales) et N^2+2N arcs (voir figure 13). Ce modèle sera M4.

On a utilisé une grammaire statistique de classes. Cette grammaire est constituée par des nœuds de classes, arcs pour les mots et arcs entre classes. Chaque nœud possède un nombre d'arcs qui partent de lui-même. Les arcs des mots qui appartiennent à la même classe ont la même valeur de probabilité. Les probabilités de transition entre les classes dépendent sur le corpus d'entraînement.
Il y a deux types de classes :

- *Classes ouvertes* – les classes qui acceptent des nouveaux mots (par exemple noms des villes, vocabulaires techniques, vocabulaires médicaux, etc) ;
- *Classes fermées* – les qui n'acceptent pas de nouveaux mots (par exemples jours de la semaine, mois de l'année, les chiffres etc).

On a créé des modèles des nouveaux mots pour chaque classe ouverte en vue de pouvoir faire la distinction si le nouveau mot est un nom de jour ou un nom d'un mois (par exemple).

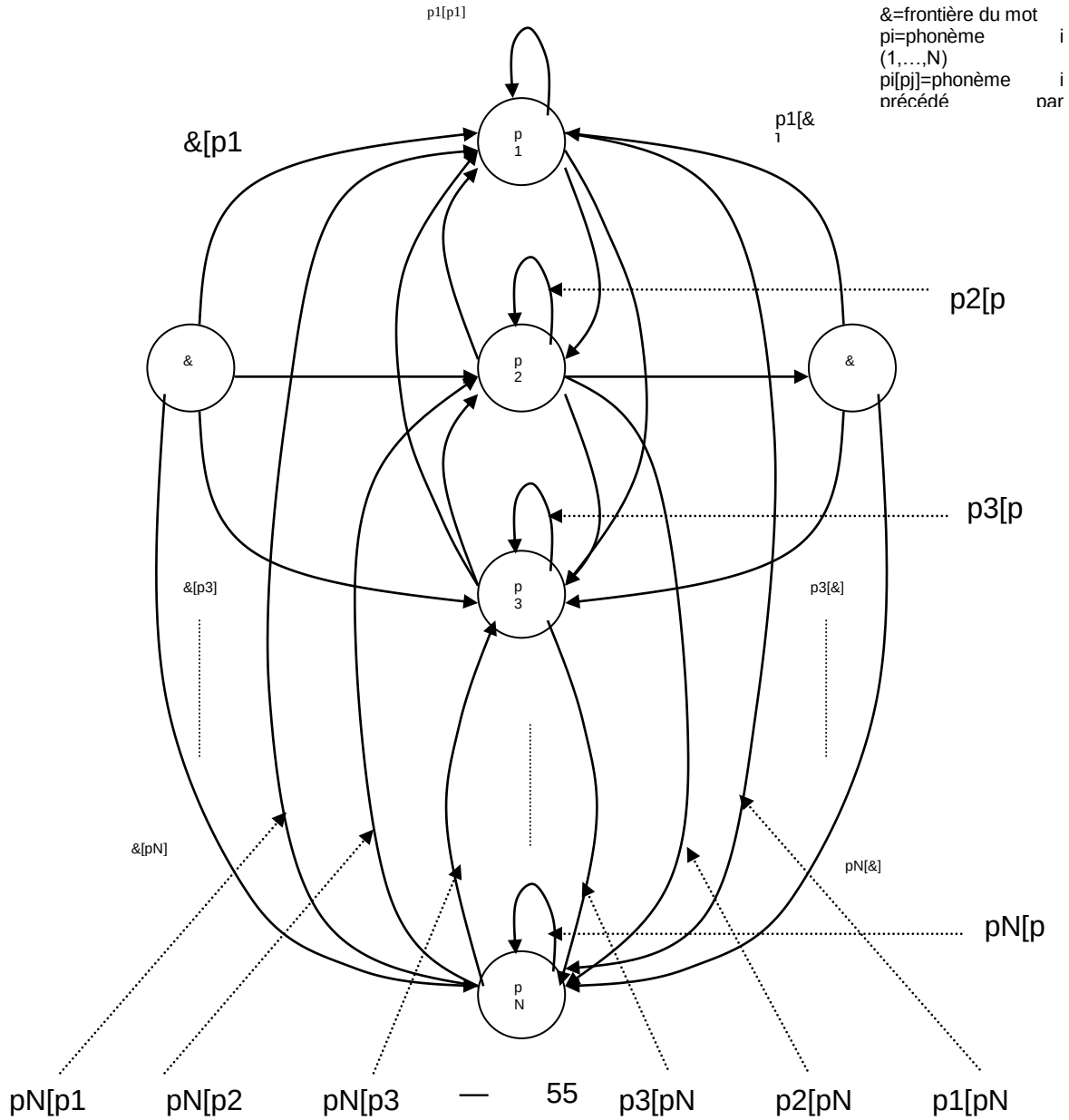

Figure 13

Des expérimentations ont été effectuées sur un vocabulaire de 1000 mots. Les nouveaux mots ont été simulés par effacer quelques mots de vocabulaire. On a utilisé 7 locuteurs, chacun d'eux prononçant 25 propositions de teste. Pour réduire la probabilité de sélectionner le nouveau mot d'une classe ouverte, une valeur *bias* a été introduite. Cette valeur est un facteur de multiplication pour les probabilités des arcs qui appartiennent au mot nouveau. De cette manière, on essaie de réduire le score de fausses alarmes.

En ce qui concerne le modèle M1, on a obtenu un score de détection de 67%, tandis que le score de fausse alarme a été de 51%. Donc, grâce à cette très grande valeur d'échec, le modèle M1 a été abandonné. Les résultats des expérimentes effectués avec les modèles M2, M3, M4 peuvent être synthétisés dans le tableau suivant :

Modèle	**Score détection (%)**	**Score fausse alarme(%)**
M2	74	3.4
M3	71	4.0
M4	76	8.6

En accord avec les résultats illustrés dans le tableau ci-dessus, on peut dire que le modèle M2 est le meilleur, grâce à sa valeur (la plus diminuée) de fausse alarme. Le modèle M3 est tout près de M2, mais comme on peut voir, M2 est meilleur sur tous les aspects. Par malheur, M4 possède le meilleur score de détection, mais en même temps le plus grand score de fausse alarme. C'est parce que le modèle diphonique M4 est bon pour les nouveaux mots mais il est un meilleur modèle pour les mots déjà existants dans le vocabulaire. C'est ensuite très difficile de régler la valeur *bias* de telle manière que le modèle détecte seulement les nouveaux mots. De plus, il est très clair que le modèle M4 est plus complexe en ce qui concerne le temps de traitement.

On a effectué aussi des expérimentations dans des situations dépendante et indépendante de locuteur. Le système a été entraîné sur un ensemble de 500 propositions qui ne contiennent pas de nouveaux mots dans le test.

Sur les figures 14 et 15 on a illustré le score de détection en fonction de score de fausse alarme obtenu pour la partie dépendante respectivement indépendante du locuteur.

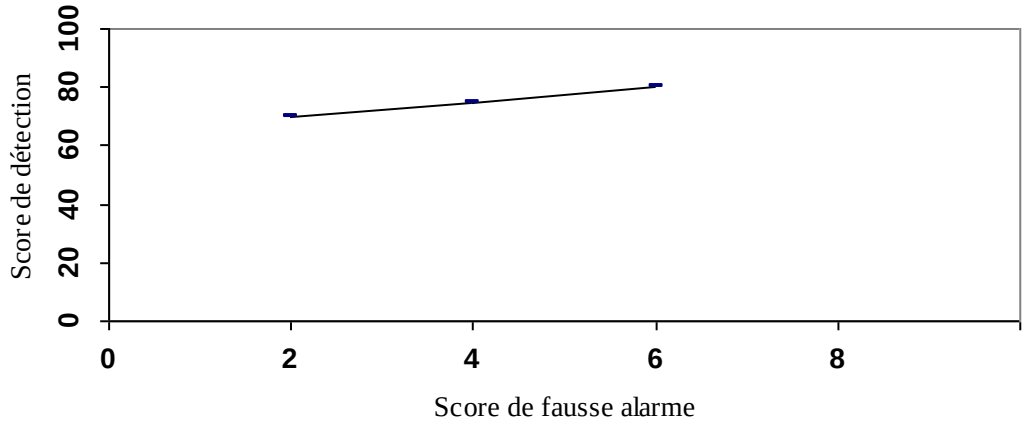

Figure 14

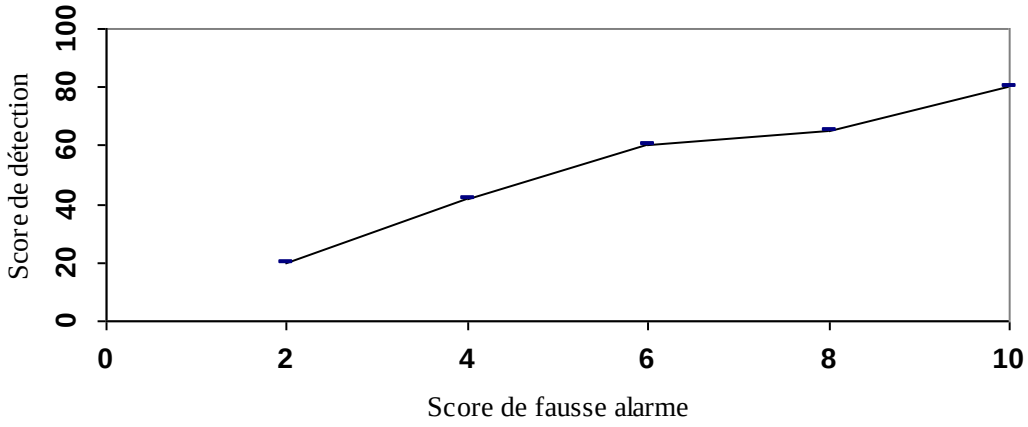

Figure 15

Les résultats pour la situation illustrée dans la figure 14 ont montré qu'on a 71% score de détection et moins de 2% fausse alarme, tandis que pour la situation de figure 15 on trouve le meilleur point avec 50% détection et plus de 2% fausse alarme. C'est à dire que cette solution pour détecter des nouveaux mots marche bien dans la situation dépendante de locuteur.

4.2 La reconnaissance phonétique

Comme on l'a déjà mentionné, quand un nouveau mot a été détecté, on veut l'ajouter au vocabulaire du système. Le système a besoin d'une transcription phonétique de ce mot pour être capable de créer un nouvel HMM. Quand le système sollicite la transcription orthographique, il vérifie si le mot est vraiment nouveau. Ensuite, le système cherche dans un dictionnaire phonétique et s'il trouve une transcription phonétique adéquate il utilisera ce qu'il a trouvé pour construire un nouveau modèle. Autrement, le système doit apprendre une transcription phonétique pour le nouveau mot en utilisant une méthode alternative.

Pour générer des transcriptions phonétiques suffisantes pour un processus de reconnaissance, Asadi *et al* ont combiné les possibilités de reconnaissance phonétique données par le système BYBLOS avec les possibilités de transcription phonétique offertes par le système DECTalk. Le tableau ci-dessous montre quelques résultats d'un processus de reconnaissance phonétique réalisés par le système BYBLOS. On a utilisé premièrement une grammaire (G1) dans laquelle n'importe quel phonème peut être suivi par n'importe quel phonème avec la même probabilité. Ensuite, on a utilisé une grammaire statistique (G2) basée sur les transcriptions phonétiques provenant de 600 propositions d'entraînement.

Unité	Grammaire	Phonèmes corrects	Score d'erreur
Phonème	G1	59.8	44.0
Phonème	G2	69.4	34.9
Diphone	G2	78.8	24.2
Triphone	G2	84.4	18.0

En accord avec ce tableau, on peut dire qu'on obtient des résultats meilleurs quand on utilise des modèles dépendants de contexte (diphones, triphones). Quand même, les résultats ne sont pas suffisants pour être utilisés dans la transcription des nouveaux mots. C'est pour ça qu'on a besoin d'une autre méthode.

L'approche suivante utilise les scores réalisés dans un processus de reconnaissance phonétique basé sur un système DECTalk (notons que les règles de transcription phonétique utilisées par ce système sont de type texte-parole). Les résultats sont illustrés dans le tableau suivant :

Transcription phonétique	Phonèmes corrects	Score d'erreur
DECTalk	88.4	12.5

Le tableau suivant présente les scores obtenus dans la reconnaissance de la parole dans deux situations :

Transcription phonétique	Score des mots erronés
Transcription manuelle	4.4
DECTalk	21.2

Donc ni les transcriptions phonétiques de DECTalk ne sont pas suffisantes pour reconnaître des nouveaux mots.

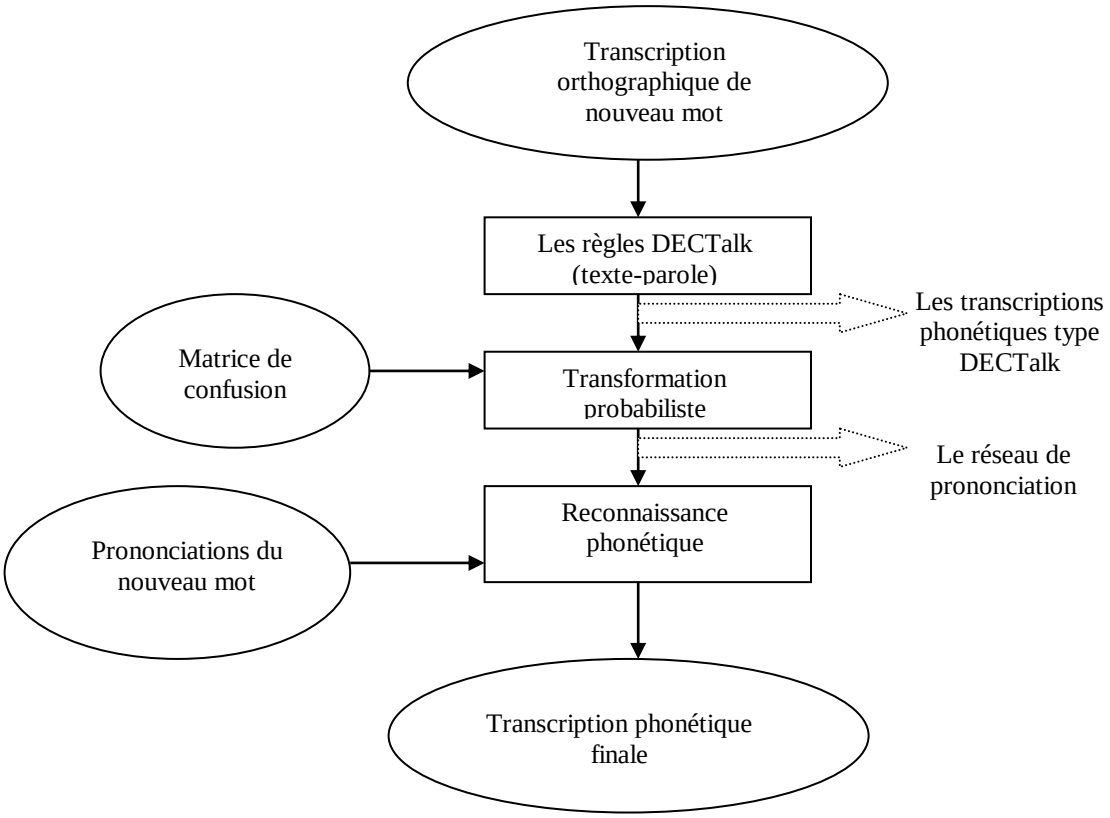

Figure 16

Pour obtenir une meilleure transcription phonétique, on a utilisé une combinaison probabiliste entre les deux sources des connaissances. Le système sollicite la transcription orthographique de nouveau mot. Le système passe la transcription au DECTalk qui produit une transcription phonétique initiale (ayant probablement des erreurs). Ensuite, une transformation probabiliste est appliquée, pour obtenir un *réseau de prononciation*. Le locuteur sera sollicité de prononcer le mot pour entrer dans un processus de reconnaissance phonétique. La figure 16 (ci-dessus) présente la méthode de transcription phonétique.

La transformation probabiliste se déroule en accord avec la description suivante :

- On a premièrement un ensemble de transcriptions orthographiques correctes pour un grand nombre de mots.
- On trouve les transcriptions phonétiques correspondantes en utilisant DECTalk.
- On calcule la matrice de confusion entre les deux ensembles des phonèmes (cette matrice contient les probabilités avec lesquelles DECTalk peut confondre les phonèmes).
- On construit le réseau de prononciation qui contient toutes les prononciations possibles pour le nouveau mot.
- On fait la reconnaissance phonétique d'une seule prononciation d'un nouveau mot, basée sur les contraints données par le réseau. En fin, la séquence phonétique avec le meilleur score sera la transcription phonétique choisie.

Il faut mentionner que dans cette situation les résultats ont été améliorés de sorte que le score d'erreur a été de 4.5%.

Quand un nouveau mot a été détecté et un modèle approprié a été créé, le mot dois être ajouté au vocabulaire du système. Pour réaliser cette tâche, le système doit apprendre la classe à laquelle le nouveau mot appartient. Le système présente à l'utilisateur une liste des classes (et un mot de chaque classe). L'utilisateur choisira la classe la plus appropriée où le nouveau mot sera placé. De plus, on a supposé que dans une classe tous les mots aient la même probabilité 1/N (N est le nombre des mots). Chaque classe aura la même propriété après l'addition d'un nouveau mot, c'est à dire que la probabilité sera après l'addition 1/(N+1).

Une autre approche pour ajouter des nouveaux mots à un vocabulaire a été proposée par Haeb *et al*. On a supposé qu'il n'y a pas des transcriptions phonétiques et qu'on a seulement quelques apparitions des nouveaux mots et un inventaire contenant les unités phonémiques qui ont été entraînées sur un corpus indépendant de locuteur. Cette approche essaie de transcrire le mot nouveau comme une séquence des unités les plus appropriées. Cette approche a deux versions :

- La première version trouve une transcription pour chaque apparition d'un nouveau mot et ensuite elle sélectionne une transcription qui a la meilleure probabilité de produire toutes les apparitions.
- La deuxième version trouve une "transcription moyenne" qui est ultérieurement "transposée" dans des unités phonémiques ou sub-phonémiques.

Considérons que $y^1, y^2, \ldots, y^n$ représentent les n apparitions d'un nouveau mot et S l'ensemble de toutes les séquences des unités. On essaie de détérminer la séquence plus probable pour chaque y^i, notée T^i, $i=1,2,\ldots,n$, $T^i \in S$:

$$T^i = \arg\max_{s \in S} P(y^i / s).$$

Ensuite, on trouve la transcription T_{mul} qui est la plus probable "productrice" des autres transcriptions. Cette valeur sera obtenue par la maximisation de produit des toutes les probabilités :

$$T_{mul} = \arg\max_{T^i \in T} \prod_{j=1}^{n} P(y^j / T^i).$$

Dans la deuxième version, on entraîne un modèle d'un mot en utilisant les probabilités d'émission de toutes les n apparitions. Ce modèle peut être interprété comme une *transcription moyenne* $\bar{\bar{y}}$. Les vecteurs d'observation de ce modèle sont les vecteurs moyens donnés par les densités de probabilité spécifiques pour chaque état. La transcription de nouveau mot sera calculée à l'aide de la séquence qui sera la plus probable "productrice" de la *transcription moyenne* :

$$T_{avg} = \arg\max_{s \in S} P(\bar{y} / s).$$

Notons aussi que la séquence T_{avg} pourra être elle-même un candidat additionnel T^{n+1} pour le modèle précédent.

Comme on l'a déjà vu, l'approche d'Asadi *et al* applique des contraintes à l'espace de recherche (dictionnaire d'un système texte-parole). Un avantage potentiel de la dernière méthode est que les prononciations insolites ont une plus grande probabilité d'être détectées, en comparaison avec l'approche d'Asadi.

Les unités courantes pour la reconnaissance automatique de la parole sont les phonèmes. Mais pour un système de transcription automatique, on peut utiliser aussi des autres unités. C'est la situation de Haeb *et al*. Leur système est basé sur des HMMs. Le modèle d'un phonème contient 3 "segments" type HMM. Un segment est représenté par une séquence de deux états avec la même fonction densité de probabilité. On utilise l'algorithme de Viterbi pour l'entraînement ainsi que pour la reconnaissance, c'est à dire que la probabilité d'un mot sera remplacée par la probabilité de la séquence la plus appropriée.

Les résultats expérimentaux obtenus avec cette approche ont montré que la transcription automatique et la transcription basée sur un dictionnaire conduisent à des scores d'erreur comparables. De plus, selon les résultats, quand le corpus est très grand, les unités sub-phonèmique offrent une meilleure transcription.

Bibliographie

[1] J. P. Haton et al, *Reconnaissance automatique de la parole*, Dunod Informatique 1991.

[2] H. Méloni (coord), *Fondements et perspectives en traitement automatique de la parole*, Aupelf-Uref, 1996.

[3] E. Keller (coord), *Fundamentals of speech synthesis and speech recognition*, John Wiley & Sons, 1994.

[4] J.C. Junqua et al, *Robustness in automatic speech recognition*, Kluwer AcademicPublishers 1995.

[5] R. A. Cole et al (coord), *Survey of the state of the art in human language technology*, Commision of the European Communities, Oregon Graduate Institute, 1995.

[6] L. Rabiner et al, *Fundamentals of speech recognition*, Prentice-Hall, 1993.

[7] E. Günter et al, *Rational Interpolation of maximum likelihood predictors in stochastic language modeling*, Proceedings Eurospeech 1997, vol 5, 2731-2734, Patras, Greece.

[8] R. Haeb-Umbach et al, *Automatic transcription of unknown words in a speech recognition system*, Proceedings ICASSP 1995, 840-843.

[9] A. Asadi et al, *Automatic detection of new words in a large vocabulary continuous speech recognition system*, Proceedings ICASSP 1990, 125-128.

[10] A. Asadi et al, *Automatic modeling for adding new words to a large vocabulary vocabulary continuous speech recognition system*, Proceedings ICASSP 1991, 305-308.

[11] H. Bourlard et al, *Optimizing recognition and rejection performance in wordspotting systems*, Proceedings ICASSP 1994, vol 1, 373-376.

[12] P. Fetter et al, *Improved modeling of Out Of Vocabulary Words in spontaneous speech*, Proceedings ICASSP 1996, 534-537.

[13] P. O'Boyle et al, *Improving N-gram Models by incorporating enhanced distributions*, Proceedings ICASSP 1996, 168-171.

[14] C. Sorin et al, *Levels in speech communication*, Elsevier 1995.