
CHAPITRE I

Reconnaissance de la parole

La parole est le moyen naturel d'échange d'informations entre les personnes. Cependant ce moyen d'échange n'est efficace que si la compréhension entre ces personnes est totale. Pour nous comprendre, il nous faut parler un même langage avec un niveau sonore suffisant. On laisse par la suite à notre cerveau le soin d'analyser les informations reçues et de réagir en conséquence. Depuis quelques décennies, on s'intéresse à introduire cette faculté dans une machine. L'échange d'information entre l'homme et la machine devenant alors complètement vocal. L'intérêt pour la reconnaissance de la parole a commencé vers les années 50 après l'apparition du spectrographe. Cette découverte a permis de mieux comprendre l'information propre à un signal de parole (temps, fréquence et amplitude) [Liénard, 77] et donc de développer des méthodes de reconnaissance.

De nombreux projets ont vu le jour depuis cette date. Par exemple le projet ARPA¹ en 1971 avait comme objectif une reconnaissance de la parole continue avec des locuteurs multiples, sur un vocabulaire de 1000 mots avec syntaxe et un taux d'erreur inférieur à 10% [Altwegg, 80]. On peut citer aussi, les projets ESPRIT¹ [Mariani, 87] (SUNDIAL [Peckham, 91], ARS, SAM, ...) du côté européen et DARPA¹ du côté américain [Baker, 87]. Ces projets ont permis aux chercheurs de mieux cerner la difficulté du problème et les ont ramenés à plus de réalisme et à restreindre leurs objectifs.

Difficulté du problème : La difficulté dans le traitement de la parole est due essentiellement à la variabilité de ce signal.

On distingue 4 classes de variabilité [Lea, 82] : intra-locuteur (vitesse d'élocution, état de santé, hésitations, etc.), inter-locuteurs (physiologie de l'appareil de production, accent, etc.),

¹ Ce sont des projets pluridisciplinaires comportant, entre autres, le traitement automatique de la parole.

canal de transmission (prise de son, bruit ambiant, ligne téléphonique, etc.) et le vocabulaire de l'application (langue, coarticulation entre mots, etc.).

Dans le cadre d'une application de serveurs vocaux interactifs (SVI) [Mathan, 91], on s'intéresse particulièrement à la variabilité inter-locuteurs et aux effets de la ligne téléphonique.

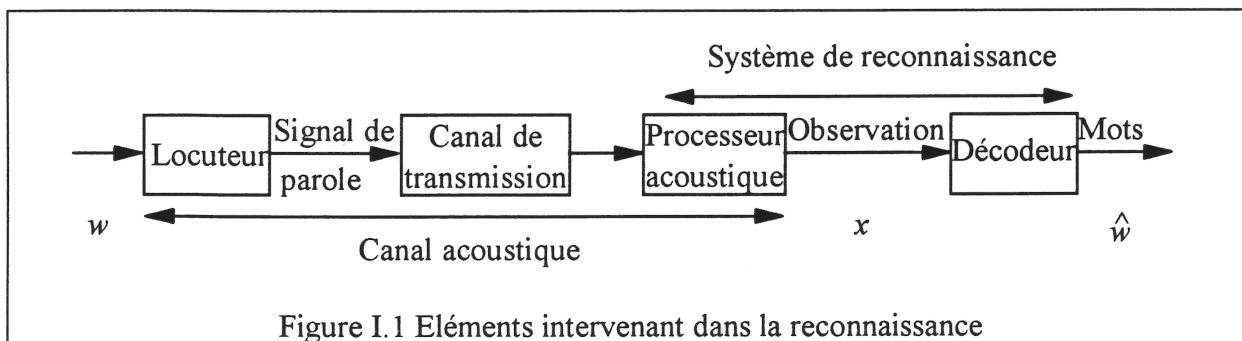
Ainsi, la conception d'un système de reconnaissance de la parole, passe par la résolution (ou le contournement) de toutes les difficultés citées plus haut. C'est plus simple à dire qu'à faire ! Les efforts engagés pour cette conquête et la ténacité des chercheurs devraient donner à réfléchir à ceux qui seraient tentés de prendre à la légère des résultats difficilement acquis depuis le début des années 80. Depuis cette date, les systèmes de reconnaissance n'ont cessé d'évoluer, d'une reconnaissance de mots isolés en mode monolocuteur vers une reconnaissance de parole continue en mode indépendant du locuteur acceptant une taille de vocabulaire modulée en fonction de la tâche à réaliser.

Les domaines d'utilisation des systèmes de reconnaissance sont vastes et variés. La réservation de places d'avion, l'aide aux handicapés, les transactions bancaires, les programmes des salles de cinéma, l'enseignement assisté par ordinateur etc.

Nous allons décrire très succinctement dans ce chapitre un système de reconnaissance et les techniques employées pour la reconnaissance de mots isolés ou connectés. On présente à la fin de ce chapitre les propriétés de quelques systèmes de reconnaissance et plus particulièrement le système de reconnaissance développé au CNET.

1 Système de reconnaissance

Un système de reconnaissance automatique de la parole est un système capable, par une analyse acoustique, d'extraire du signal de parole des informations utiles et compréhensibles par la machine et de reconnaître, à partir de ces informations, par un procédé de décodage, le mot, la phrase ou le texte émis par le locuteur.



Dans une approche statistique on considère que l'ensemble constitué par le locuteur, le canal de transmission et l'analyseur acoustique constitue un canal acoustique transformant un texte w en une observation X . Le rôle du décodeur est de déterminer la séquence $\hat{w}$ qui fournit la meilleure interprétation de cette observation.

1.1 Analyse acoustique

Cette étape sert à extraire le maximum d'informations pertinentes du signal vocal représentées par un minimum de paramètres. Parmi les types possibles de paramètres, on trouve l'énergie dans différentes bandes de fréquences (spectre), la fréquence fondamentale (pitch), le comptage des passages par zéro du signal et de sa dérivée etc.

L'analyse acoustique calcule un nombre de coefficients préalablement fixé toutes les 10 à 20 ms. L'ensemble de ces coefficients constitue un vecteur associé à une trame acoustique. Par extension, nous appellerons une trame acoustique le vecteur qui lui est associé.

Il existe différents types de coefficients. On peut citer les coefficients cepstraux (MFCC : Mel Frequency Cepstrum Coefficients), coefficients de prédiction linéaire (LPC : Linear Prediction Coefficients) ... [Calliope, 89].

1.2 Décodage des informations

Le rôle du décodeur est de déterminer à partir d'une suite de trames acoustiques, les mots ou phrases reconnus à l'aide d'un algorithme de comparaison. On distingue deux approches, une approche globale et une approche analytique. L'approche globale consiste à tenter de reconnaître le mot (ou la phrase) prononcé comme une entité unique, sans chercher à le scinder en ses composants phonétiques.

L'approche analytique, au contraire, tire parti de la structure et des propriétés phonétiques des mots pour isoler et identifier les constituants élémentaires (phonèmes, diphtongues, etc.) du message parlé, avant de les recombinaison pour déterminer le mot (ou la phrase) prononcé [Haton, 91].

Malgré cette distinction, il est possible d'utiliser le même algorithme pour identifier un mot dans une approche globale et un segment de parole dans une approche analytique.

L'approche globale implique une phase d'apprentissage et une phase de reconnaissance. La phase d'apprentissage sert à obtenir pour chaque mot du vocabulaire une représentation compacte, robuste aux variations de la voix du locuteur et de la prise de son. La phase de reconnaissance sert à déterminer le mot prononcé à partir des références obtenues dans la phase d'apprentissage.

2 Caractéristiques des systèmes de reconnaissance

Les systèmes de reconnaissance de la parole peuvent être distingués selon les 3 critères suivants :

Mode de fonctionnement : indépendant du locuteur (n'importe quel utilisateur peut utiliser le système sans apprentissage préalable), monolocuteur (un utilisateur à la fois et cela après apprentissage) ou plurilocuteur (un groupe restreint de personnes).

Mode d'élocution : mots isolés (un mot à la fois), mots connectés (des séquences de mots avec ou sans pause) ou parole continue (des phrases au sens habituel du terme).

Mode de décodage d'informations : approche analytique ou approche globale.

On s'intéresse dans ce qui suit à décrire les différentes techniques de reconnaissance dans le cas d'une approche globale. Les mêmes techniques peuvent être utilisées dans l'approche analytique.

3 Techniques de reconnaissance

Dans le cas d'une approche globale, trois techniques sont utilisées en phase de reconnaissance : l'alignement temporel, la méthode statistique et la méthode connexionniste.

3.1 Alignement temporel

L'alignement temporel permet de calculer la distance minimale entre une forme acoustique inconnue correspondant au mot à reconnaître et une forme de référence disponible pour chacun des mots du vocabulaire. Ensuite, la forme de référence fournissant la distance minimale détermine le mot reconnu. Cette méthode repose sur un algorithme de programmation dynamique [Bellman, 57] permettant l'ajustement temporel des mots à comparer : l'algorithme DTW (Dynamic Time Warping) [Sakoe, 71]. Des contraintes sont fixées préalablement sur la coïncidence des vecteurs de début et de fin de mots et sur les règles d'évolution entre 2 trames consécutives [Itakura, 75], [Sakoe, 78].

Cette méthode est généralement utilisée en mode dépendant du locuteur. En mode indépendant du locuteur, de nombreuses références sont nécessaires pour chacun des mots du vocabulaire. Ce nombre important de références par mot est dû à la grande variabilité du signal de parole. Il faut alors une taille mémoire importante pour stocker ces références, et le coût de traitement est également très élevé.

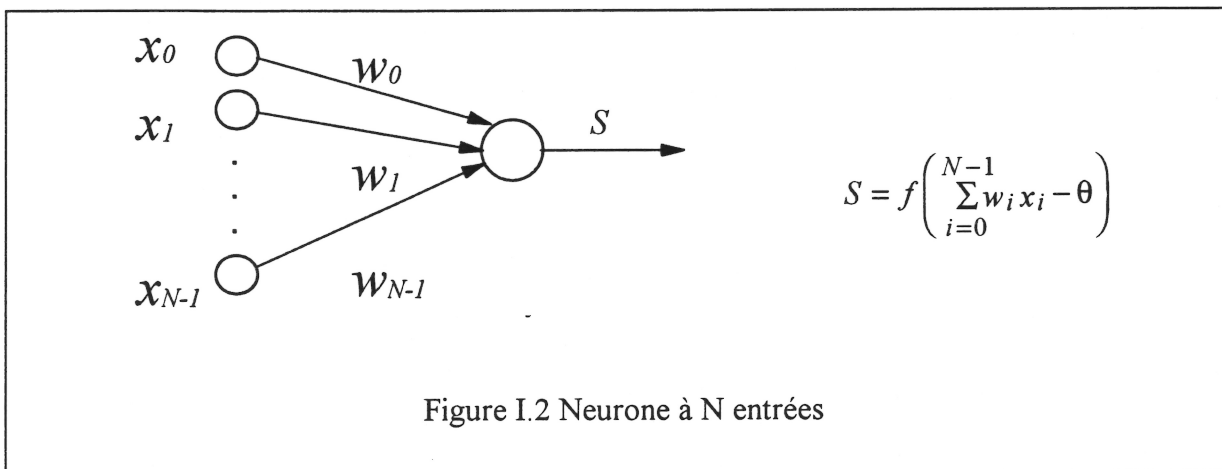
3.2 Méthode statistique

Pour palier aux problèmes rencontrés dans la méthode d'alignement temporel (en mode indépendant du locuteur), une méthode qui consiste à remplacer l'ensemble des références acoustiques résultant des différentes prononciations d'un mot, par un modèle de ces prononciations, a été proposée. Ainsi chaque mot du vocabulaire est modélisé statistiquement par un modèle de Markov caché [Jelinek, 76], [Levinson, 83], [Rabiner, 89]. Ce modèle permet une modélisation des variantes de prononciation des mots et ceci avec un coût d'occupation mémoire moindre. La section 4 présentera plus en détails les modèles de Markov cachés.

3.3 Méthode connexionniste

Il y a eu ces dernières années un regain d'intérêt pour les modèles connexionnistes à cause de leur pouvoir de discrimination et du développement de l'algorithme de rétropropagation du gradient pour l'apprentissage des paramètres [Lecun, 87], [Rumelhart, 86]. Cependant, en reconnaissance automatique de la parole, l'aspect séquentiel (trame par trame) est difficile à modéliser.

Un réseau de neurones est un ensemble de cellules interconnectées. Chaque connexion est affectée d'un poids. Le résultat obtenu à la sortie d'une cellule est la somme pondérée de ses entrées, filtrée par une fonction de base non linéaire (Figure I.2).



Chaque cellule du réseau est caractérisée par une fonction non linéaire et par un seuil d'activation. La figure I.3 illustre deux types de fonctions non linéaires, une fonction à seuil $(-1, +1)$ et une sigmoïde. Le paramètre adaptable θ est appelé le seuil de la cellule.

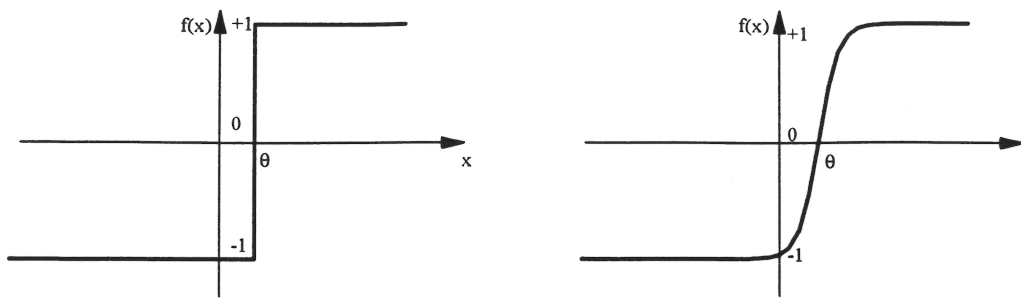


Figure I.3 Exemple de fonctions non linéaires

Un réseau de neurones est caractérisé par sa topologie (nombre de cellules, connexions entre les cellules, etc.) et les caractéristiques des cellules (fonctions de base). La phase d'apprentissage consiste à déterminer les poids des connexions minimisant l'erreur mesurée entre la sortie réelle du réseau et la sortie désirée.

En phase de reconnaissance, le vecteur d'observation inconnu (vecteur d'entrée) génère un vecteur en sortie. Ce vecteur est comparé aux vecteurs cibles associés aux classes. La classe associée au vecteur cible le plus proche définit la classe d'appartenance du vecteur d'entrée.

4 Introduction aux modèles de Markov cachés

Nous avons introduit dans le paragraphe 3.2 relatif à la méthode statistique, l'utilisation de modèles de Markov cachés pour représenter l'évolution acoustique d'un mot, dans une approche globale. Ce paragraphe décrit avec plus de détails un modèle de Markov caché et l'algorithme de reconnaissance utilisé dans l'approche statistique.

4.1 Modèles de Markov

Les modèles de Markov utilisés résultent de l'association d'un ensemble de fonctions de densité de probabilité et d'une chaîne de Markov d'ordre 1. Les fonctions de densité de probabilité fournissent des probabilités sur l'espace des trames acoustiques, tandis que la chaîne de Markov, formée d'états et de transitions entre ces états, sert de support à cet ensemble de distributions. Ces distributions peuvent être associées aux états ou aux transitions. Elles peuvent être choisies continues ou discrètes.

Pour un modèle de Markov on définit les paramètres suivants :

NQ le nombre d'états dans le modèle,

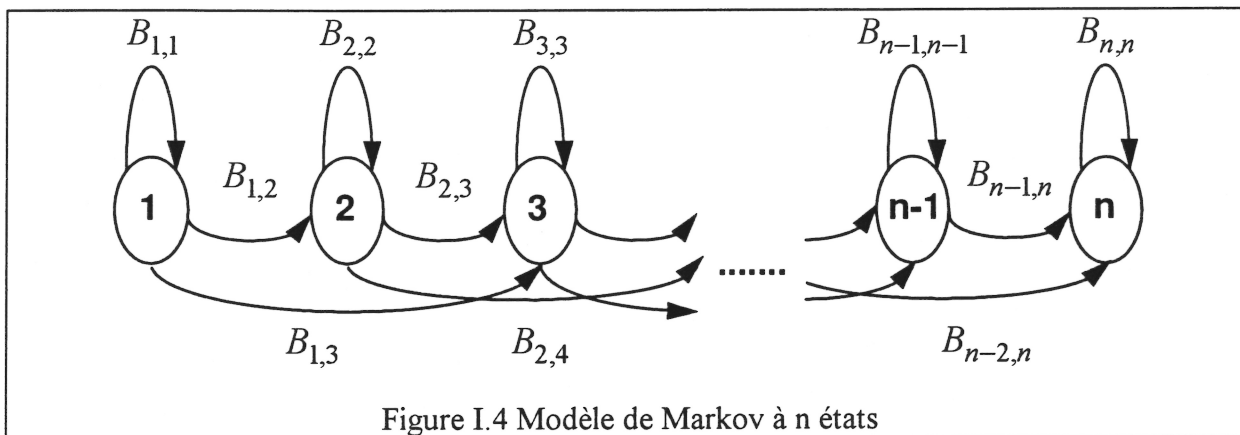
$\{q_i ; i = 1, \dots, NQ\}$ l'ensemble des états du modèle.

$\{t_{i,j} ; i, j = 1, \dots, NQ\}$ l'ensemble des transitions possibles entre les états du modèle.

$\{\pi_i; i = 1, \dots, NQ\}$	les probabilités initiales affectées à chacun des états du modèle.
$\{a_{i,j}; i,j = 1, \dots, NQ\}$	les probabilités de transition entre les états du modèle.
	$a_{i,j}$ représente la probabilité de passage de l'état q_i vers l'état q_j .
$\{B_{i,j}; i,j = 1, \dots, NQ\}$	les distributions acoustiques associées aux transitions.

Les distributions acoustiques $B_{i,j}$ fournissent les probabilités d'émission des trames acoustiques. Par analogie avec l'alignement temporel, la probabilité $P(X[\tau] / B_{i,j})$ d'émission d'une trame $X[\tau]$ pour une loi de probabilité $B_{i,j}$ joue le rôle d'une mesure de similarité entre la trame observée $X[\tau]$ et une trame de référence.

La figure suivante (I.4) illustre un modèle de Markov de Bakis (gauche droit, le plus fréquemment utilisé en parole) à n états. Les distributions acoustiques $B_{i,j}$ sont associées aux transitions.



Sur ce modèle les trois transitions partant d'un même état (sauf le dernier) correspondent à :

$t_{i,i}$	le bouclage sur l'état.
$t_{i,i+1}$	la transition vers l'état suivant.
$t_{i,i+2}$	le saut de l'état suivant.

4.2 Phase de reconnaissance

Dans une méthode statistique, la séquence optimale $\hat{w}$ à la sortie du décodeur (figure I.1), s'obtient en recherchant, parmi toutes les séquences w possibles, celle qui est la plus probable, soit :

$$P(\hat{w} / X) = \underset{w}{\text{Max}} P(w / X) \quad (\text{I.1})$$

Or d'après la formule de Bayes :

$$P(w / X) = \frac{P(X / w) \cdot P(w)}{P(X)} \quad (\text{I.2})$$

$P(X)$ ne dépend pas de la séquence w . La séquence $\hat{w}$ qui fournit la meilleure interprétation de l'observation est donc celle qui maximise la probabilité conjointe $P(X, w)$:

$$P(X, \hat{w}) = \underset{w}{\text{Max}} P(X / w) \cdot P(w) \quad (\text{I.3})$$

$P(w)$ représente la probabilité a priori, liée au langage de l'application, que la séquence w soit émise. Cette probabilité est déterminée à partir de la syntaxe de l'application. $P(X/w)$ est la probabilité, liée au canal acoustique, d'observer X quand la séquence w est prononcée. C'est le calcul de cette probabilité que nous allons maintenant aborder.

4.3 Probabilité d'émission des observations

Une observation X est composée d'une suite de vecteurs de longueur T . A chaque mot du vocabulaire w est associé un modèle M . On cherche à déterminer $P(X/M)$, la probabilité d'émission de l'observation X par le modèle M donné. Cette probabilité est calculée en sommant toutes les probabilités conjointes $P(X, s/M)$ sur tous les chemins s de longueur T :

$$P(X / M) = \sum_s P(X, s / M) = \sum_s P(X / s, M) \cdot P(s / M) \quad (\text{I.4})$$

$P(X/s, M)$ est la probabilité d'émission de l'observation X le long du chemin s et $P(s/M)$ est la probabilité du chemin s .

Le calcul direct de cette probabilité telle que formulée dans l'équation (I.4) étant impraticable, on utilise une autre méthode connue sous le nom "avant-arrière" [Baum, 72].

On peut s'intéresser aussi à la recherche du chemin optimal, c'est-à-dire celui qui fournit la probabilité d'émission de l'observation la plus élevée.

$$P(X, \hat{s} / M) = \underset{s}{\text{Max}} P(X, s / M) \quad (\text{I.5})$$

Cette probabilité est déterminée par l'algorithme de Viterbi [Viterbi, 67], [Forney, 73].

4.4 Algorithme de Viterbi

On calcule la fonction $\Phi(\tau, q_i)$ représentant la probabilité maximale d'émission des τ premiers vecteurs $X[1], \dots, X[\tau]$ de l'observation X sur tous les chemins de longueur τ arrivant à l'état q_i à cet instant. La probabilité initiale à l'instant 0 et pour l'état q_i est donnée par :

$$\Phi(0, q_i) = \pi_i \quad (\text{I.6})$$

On montre que $\Phi(\tau + 1, q_j)$ est obtenue par la formule récurrente suivante :

$$\Phi(\tau + 1, q_j) = \underset{q_i}{\text{Max}} \{ \Phi(\tau, q_i) \cdot a_{i,j} \cdot B_{i,j}(X[\tau + 1]) \} \quad (\text{I.7})$$

Si l'on désire connaître les états visités le long du chemin optimal, il faut définir une fonction supplémentaire $\Psi(\tau + 1, q_j)$ qui donne pour chaque instant $\tau + 1$ et chaque état q_j l'état q_i^* qui correspond au maximum de l'équation (I.7). Un retour en arrière à partir de l'état final permet de retrouver un à un les états visités. L'algorithme global est donné ci-dessous :

Initialisation	Pour $i = 1, \dots, NQ$ faire $\Phi(0, q_i) = \pi_i$
Itérations	Pour $\tau = 1, \dots, T$ faire Pour $j = 1, \dots, NQ$ faire $\begin{cases} \Phi(\tau, q_j) = \underset{q_i}{\text{Max}} \{ \Phi(\tau - 1, q_i) \cdot a_{i,j} \cdot B_{i,j}(X[\tau]) \} \\ \Psi(\tau, q_j) = q_i^* \end{cases}$
Résultat	$P(X, \hat{s} / M) = \underset{q_i}{\text{Max}} \Phi(T, q_i)$

Dans le cas d'une application comportant un seul état initial q_I et un seul état final q_F la probabilité initiale à l'instant 0 est donnée par :

$$\pi_I = 1 ; \text{ et } \pi_i = 0 \quad \forall i \neq I \quad (\text{I.8})$$

A l'instant final T , la probabilité conjointe du chemin optimal et de l'observation est donnée par la valeur de Φ pour l'état final q_F :

$$P(X, \hat{s} / M) = \Phi(T, q_F) \quad (\text{I.9})$$

4.5 Détermination des paramètres du modèle

L'ensemble des paramètres du modèle de Markov (probabilités initiales, probabilités des transitions, paramètres des fonctions de densité de probabilité) est déterminé dans la phase d'apprentissage.

Considérons un modèle de Markov caché M d'un mot w . Pour apprendre les paramètres de ce modèle, on dispose de KP prononciations de ce mot. On veut déterminer les paramètres optimaux $\hat{\theta}$ du modèle M connaissant l'ensemble des prononciations (ou observations) X^r . Ces paramètres optimaux sont obtenus en maximisant la probabilité d'émission de l'ensemble X^* des observations X^r

$$P(X^* / M(\hat{\theta})) = \underset{\theta}{\text{Max}} P(X^* / M(\theta)) = \underset{\theta}{\text{Max}} \left\{ \prod_{r=1}^{KP} P(X^r / M(\theta)) \right\} \quad (\text{I.10})$$

Dans un premier temps, on initialise les paramètres du modèle. L'initialisation peut être effectuée à partir d'une segmentation linéaire d'un ensemble de prononciations du mot. Ensuite à chaque itération i , on détermine les nouveaux paramètres θ_i du modèle de manière à augmenter la probabilité d'émission des données :

$$P(X^* / M(\theta_i)) \geq P(X^* / M(\theta_{i-1})) \quad (\text{I.11})$$

On arrête le processus itératif lorsqu'il n'y a plus de changement significatif sur les paramètres (convergence vers un extremum local).

Les algorithmes utilisés pendant la phase d'apprentissage sont l'algorithme de Baum-Welch [Baum, 72] ou celui de Viterbi.

L'algorithme de Baum-Welch utilise les informations le long de tous les chemins possibles du réseau, pondérées par la contribution du chemin à la probabilité de l'observation. Il repose sur l'algorithme de calcul "avant-arrière". L'algorithme de Viterbi, par contre, n'utilise que les informations présentes le long des chemins optimaux. De ce fait, il prend en considération beaucoup moins d'événements, et pour compenser, il faut accroître sensiblement la taille de l'ensemble d'apprentissage.

5 Reconnaissance de mots connectés

Les techniques citées dans le cas de la reconnaissance de mots isolés se généralisent pour la reconnaissance des mots connectés. Cependant d'autres problèmes interviennent dans ce cas : la segmentation de la phrase en un ensemble de mots (difficile de déterminer les frontières de début et de fin de mots) et l'effet de coarticulation (une déformation de la réalisation acoustique des mots, due à la présence de mots avant ou après).

Ce paragraphe décrit deux méthodes de reconnaissance de mots connectés qui ne réalisent pas une segmentation préalable de la phrase. Cette segmentation est implicitement réalisée pendant la phase de comparaison. Ces deux méthodes sont présentées, dans un souci de meilleure compréhension, dans un ordre non chronologique de leur apparition.

5.1 One-Pass

C'est le plus simple des algorithmes de reconnaissance de mots connectés. Il a été développé par [Bridle, 82] puis reformulé par [Ney, 84].

Soit un vocabulaire R constitué de V mots. Pour chaque mot est réalisée une référence (représentation la plus significative ou modèle de Markov caché) $R(i)$. On cherche à identifier l'observation X de longueur T dans l'ensemble des références (ou unités) R_S .

$$R_S = \{R(1), R(2), \dots, R(i), \dots, R(V)\} \quad (\text{I.12})$$

L'algorithme One-Pass est un algorithme synchrone. La progression à l'intérieur de chaque unité est réalisée de la même manière que dans la reconnaissance de mots isolés. A cela s'ajoute des contraintes inter-unités. Ces contraintes autorisent la succession de n'importe quel mot par un autre mot du même vocabulaire.

A la fin de la procédure, à l'instant T , on cherche l'unité qui fournit le meilleure score. On remonte le chemin suivi par cette unité pour retrouver la séquence de mots reconnus. Cette technique est d'une complexité $O(T)^1$.

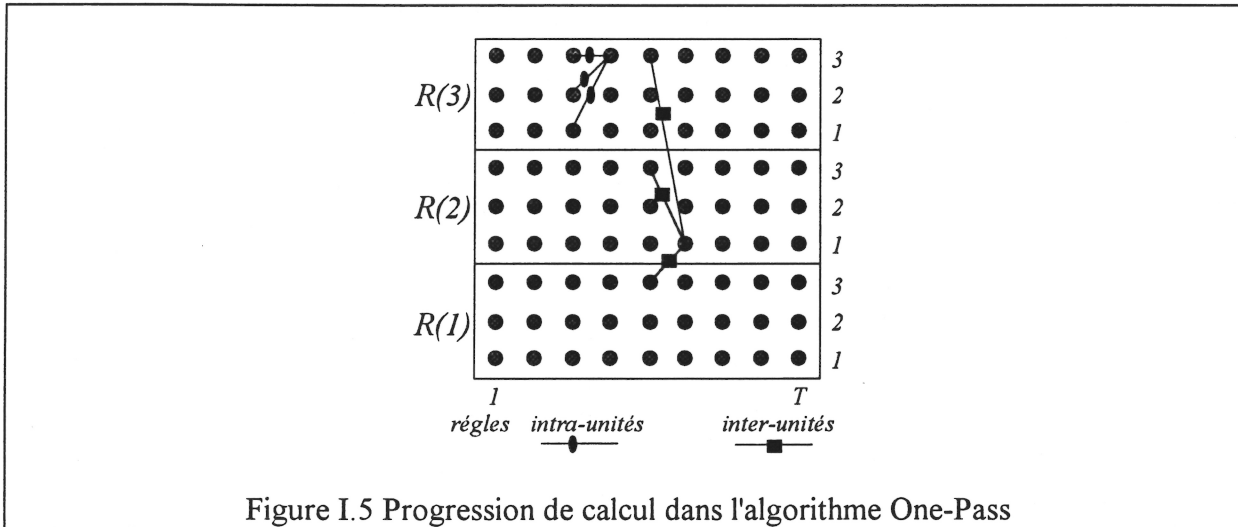


Figure I.5 Progression de calcul dans l'algorithme One-Pass

¹ $O(x)$ est une fonction proportionnelle à x .

La figure I.5 présente les deux contraintes de progression dans la phase de décodage, les contraintes locales à l'intérieur de chaque référence (règles intra-unités) du type Itakura ou fournies par la topologie des HMMs et les contraintes globales (règles inter-unités) qui définissent la syntaxe de l'application.

Nous représentons ci-dessous (figure I.6), deux séquences de mots possibles. La séquence en trait continu est la séquence reconnue car son score de probabilité final à l'instant T est le plus élevé.

La phrase prononcée correspond donc à $R(2) R(1) R(3)$.

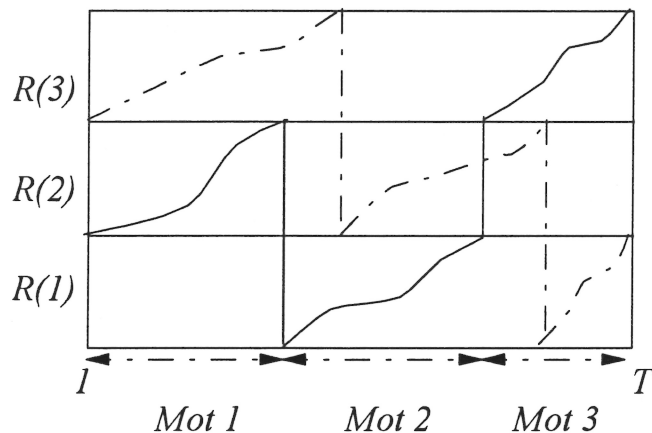


Figure I.6 Détermination de la séquence reconnue, par l'algorithme One-Pass

5.2 Level-Building

Cette technique a été développée par [Myers, 81-a]. On commence par construire L niveaux (L est connu préalablement et désigne le nombre maximum de mots connectés dans l'observation). Dans chaque niveau, on retrouve l'ensemble des références R_s . La procédure de reconnaissance dans chaque niveau est équivalente à une technique de reconnaissance de mots isolés. Les règles inter-unités dans le One-Pass deviennent des règles inter-niveaux dans le Level-Building. Sa complexité est de $O(L * T)$.

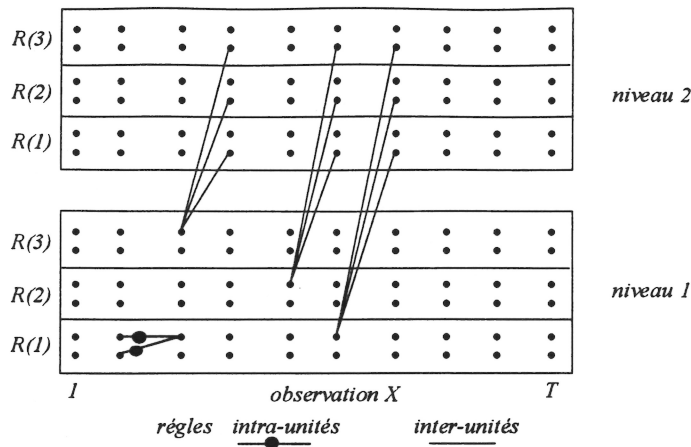


Figure I.7 Progression de calcul dans l'algorithme Level-Building

Nous avons représenté sur la figure (I.7) deux niveaux seulement (on n'autorise qu'une succession de deux mots). On a présenté dans ce cas les règles intra-unités et les règles inter-niveaux. La séquence de mots reconnue est retrouvée de la même manière que dans le One-Pass.

6 Quelques systèmes de reconnaissance

Ce paragraphe présente quelques systèmes de reconnaissance de la parole. Ceux utilisés aujourd'hui sont pour la plupart à base de HMM.

BYBLOS (BBN) [Chow, 87], [Kubala, 88] est un système de reconnaissance de parole continue dépendant du locuteur. Il est conçu pour un très large vocabulaire. Ce système est à base de modèles de Markov cachés.

MOZART (LIMSI) [Gauvain, 82] a été développé pour la reconnaissance de petits vocabulaires (quelques centaines de mots prononcés de manière isolée ou connectée). Depuis et afin d'aborder la reconnaissance de plus grand vocabulaires (de l'ordre de plusieurs milliers de mots) un nouveau système de reconnaissance a vu le jour, **AMADEUS** [Limsi, 89]. Ce système a été conçu pour la reconnaissance de 5000 mots en mode isolé et environ 300 mots en mode connecté. Cette taille peut-être étendue en multipliant le nombre de processeurs utilisés lors de la phase de reconnaissance [Lamel, 92]. Ces deux systèmes utilisent des processeurs de comparaison dynamique spécialisés et fonctionnent en mode dépendant du locuteur en temps réel. Depuis un système de reconnaissance a vu le jour. Ce système fonctionne en mode indépendant du locuteur, utilise une approche globale statistique et comme unité de reconnaissance le phonème en contexte. Des contraintes de types n-grams sont intégrées au niveau syntaxique [Limsi, 92].

Séraphine (CNET) [Gagnoulet, 82] est un système de reconnaissance fonctionnant en mode monolocuteur et permettant la reconnaissance d'un petit vocabulaire, 100 mots en mode isolé et des courtes phrases (maximum 10 mots). L'algorithme utilisé pour la phase de reconnaissance est la DTW. Par la suite un nouveau logiciel, appelé PHIL86 [Jouvet, 88], [Gagnoulet, 90] a été élaboré pour permettre la reconnaissance en mode indépendant du locuteur en utilisant l'approche par modèles de Markov cachés. Des améliorations ont été apportées depuis sur les différents modules du système de reconnaissance PHIL86 pour donner naissance à PHIL90 [Jouvet, 91]. La section suivante présente les différents modules de ce système de reconnaissance, qui a servi de support à notre travail.

SPICOS (Philips) [Paeseler, 89], [Ney, 92-a] est un système de reconnaissance de parole continue dépendant du locuteur. La taille du vocabulaire utilisé avoisine les 1000 mots. Les modèles de Markov cachés sont utilisés dans la description acoustique des phonèmes. L'originalité de ce système réside dans l'utilisation des fonctions de probabilités de Laplace au lieu des fonctions de probabilités gaussiennes habituellement employées.

SPHINX II (CMU) [Huang, 93] est un système de reconnaissance de la parole continue indépendant du locuteur pour un vocabulaire de 1000 mots environ. L'application est modélisée par des unités phonétiques plus courtes que le phonème (senone). Chaque unité est représentée au niveau acoustique par des modèles de Markov cachés semi-continus [Huang, 90]. Les locuteurs sont divisés en deux classes : hommes ou femmes.

TANGORA (IBM) [Averbuch, 86], [Averbuch, 87] est un système de reconnaissance monolocuteur conçu pour des vocabulaires de très grande taille (20000 mots). Il est réalisé à partir d'une approche globale à base de modèles de Markov cachés. Le développement de cartes spécialisées a permis l'intégration d'un tel système sur un PC-AT pour un vocabulaire de 5000 à 20000 mots. La reconnaissance est réalisée en temps réel et chaque prononciation est effectuée en mots isolés. Des applications ont été développées avec le système TANGORA pour la reconnaissance de la parole automatique dictée.

Une adaptation du système de reconnaissance TANGORA à la langue française a été réalisée par l'équipe IBM-France [Cerf-Danon, 92].

PARSYFAL [Cerf-Danon, 89] est un système de reconnaissance développé, pour la langue française, par l'équipe du centre IBM-France. La parole est prononcée en syllabes isolées [Merialdo, 87]. Le système utilise des dictionnaires contenant de 10000 à 200000 mots afin de couvrir la totalité du vocabulaire.

7 Système de reconnaissance PHIL90

Le système de reconnaissance développé au CNET (PHIL90), permet la reconnaissance de vocabulaires dont la taille reste limitée à une centaine de mots environ, de manière indépendante du locuteur. Il autorise la reconnaissance de mots isolés ou de mots connectés. Ce système, qui utilise un réseau compilé (une procédure d'optimisation du réseau est introduite dans l'étape de compilation), permet l'introduction de règles syntaxiques, lexicales et phonologiques. Il utilise l'approche par modèles de Markov cachés pour représenter l'ensemble des mots du vocabulaire. Les phases d'apprentissage et de reconnaissance sont réalisées par l'algorithme de Viterbi. Ce système a été essentiellement développé pour la réalisation de serveurs vocaux interactifs [Toularhoat, 92].

7.1 Modélisation statistique

Les modèles de Markov sont représentés par un ensemble d'états q_i , de transitions $t_{i,j}$ et de fonctions de densité de probabilité $B_{i,j}$. Ces fonctions de densité de probabilité sont associées aux transitions, et sont choisies continues et gaussiennes. Chaque fonction de densité de probabilité possède deux paramètres, le vecteur des moyennes $m_{i,j}$ et la matrice de covariance diagonale $\Sigma_{i,j}$.

7.2 Analyse acoustique

L'analyse acoustique calcule toutes les 16 ms (fenêtre de Hanning de 32 ms avec un recouvrement de 50%) 8 coefficients cepstraux obtenus à partir des sorties d'un banc de filtres uniformément répartis sur une échelle Mel (MFCC), complétés par un paramètre d'énergie, plus les dérivées temporelles d'ordre un et deux de ces 9 coefficients.

7.3 Modélisation d'une application

Une syntaxe décrit l'ensemble des phrases admises par l'application, l'exemple suivant (figure I.8) présente la syntaxe des chiffres isolés de 0 à 9 précédés et suivis d'un silence.

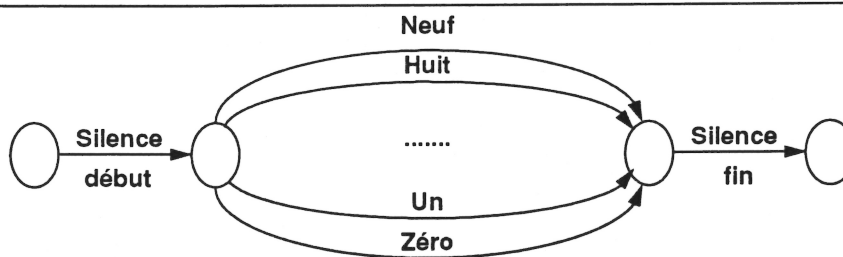


Figure I.8 Modélisation syntaxique d'une application

Chacun des éléments syntaxiques est découpé en unité syntaxique et mot syntaxique. L'unité syntaxique sert à la définition de la modélisation (mots, phonèmes, allophones, ...). Le mot syntaxique est l'étiquette identifiant l'unité syntaxique lui correspondant. La suite des mots syntaxiques présents le long du chemin optimal identifie la phrase reconnue. La figure suivante (I.9) illustre l'élément syntaxique du chiffre Zéro.

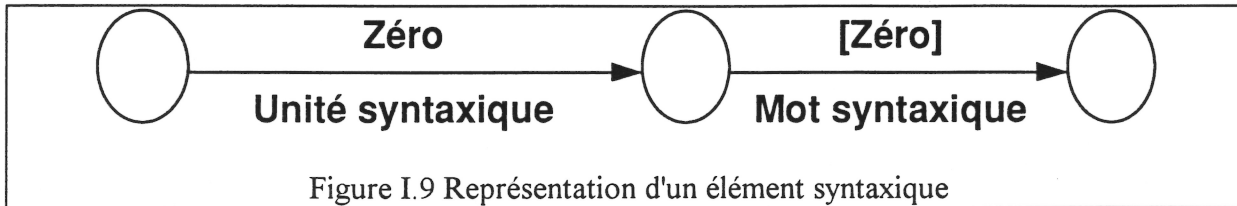


Figure I.9 Représentation d'un élément syntaxique

Chaque unité syntaxique peut être modélisée de différentes manières.

7.3.1 Modélisation par mots

Un élément syntaxique est représenté au niveau acoustique par un modèle de Markov caché à n états. La figure suivante (I.10) schématise un modèle régulier à n états employé pour modéliser le chiffre Zéro au niveau acoustique.

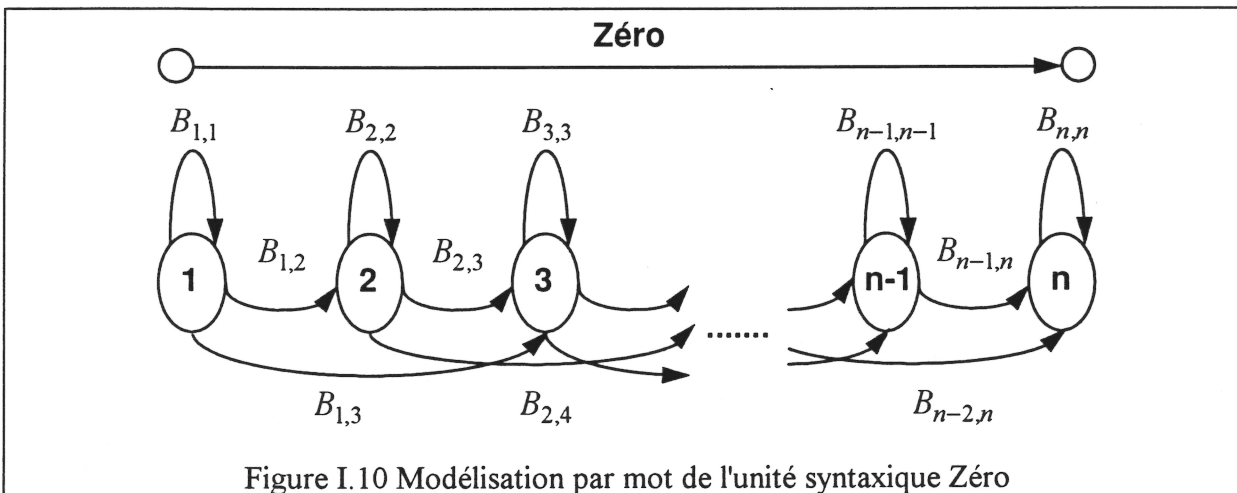
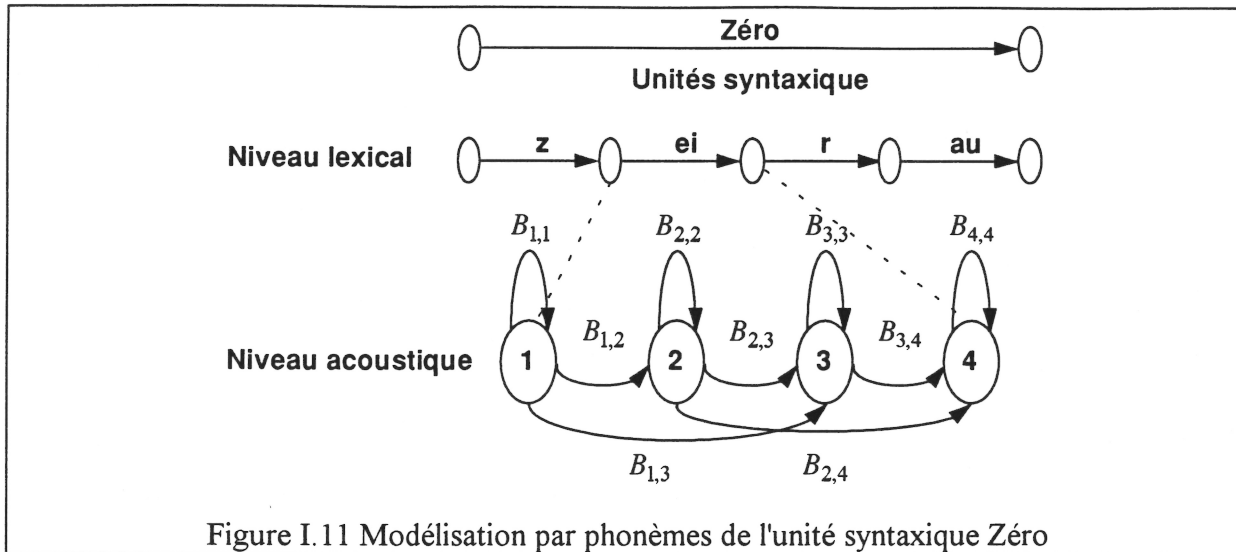


Figure I.10 Modélisation par mot de l'unité syntaxique Zéro

7.3.2 Modélisation par phonèmes

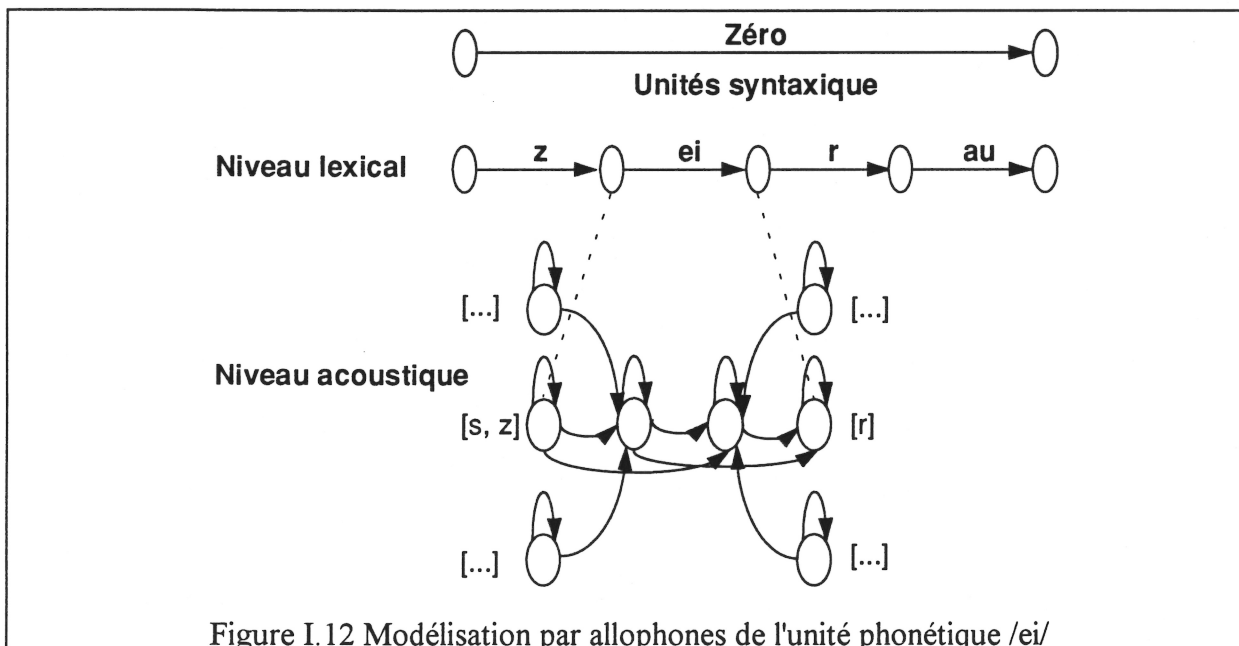
Une description lexicale transforme chaque mot du vocabulaire, en terme d'unités arbitraires plus petites ; des phonèmes par exemple (/z/, /ei/, /r/, /au/ pour le chiffre Zéro). Chaque unité phonétique doit être modélisée au niveau acoustique par un modèle de Markov caché.

La figure suivante (I.11) illustre le niveau lexical de l'unité syntaxique Zéro. On a représenté uniquement le niveau acoustique du phonème /ei/ par un modèle de Markov caché à 4 états (choix arbitraire, pour les besoins de la figure).



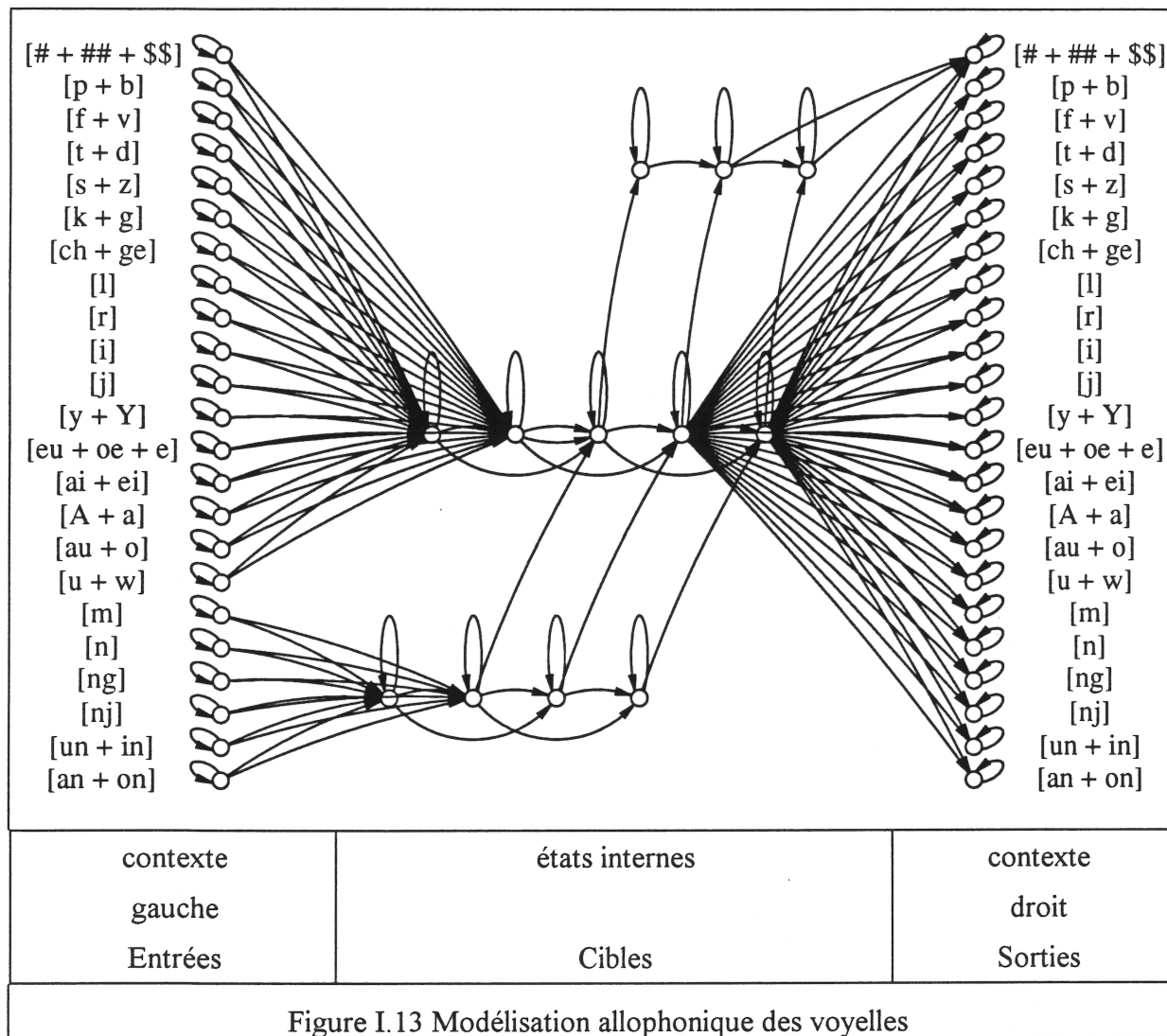
7.3.3 Modélisation par allophones

La réalisation acoustique d'un phonème étant très dépendante du contexte dans lequel il apparaît (c'est-à-dire des phonèmes qui l'entourent), il est naturel de prendre en considération ces influences contextuelles. C'est l'approche par allophones qui consiste à utiliser pour chaque phonème un modèle acoustique comportant plusieurs entrées et plusieurs sorties, choisies et validées en fonction du contexte dans lequel le phonème apparaît [Jouvet, 91].



Dans la figure I.12, le niveau acoustique schématise un modèle acoustique à 3 entrées et 3 sorties. Lorsque ce modèle est utilisé pour le phonème /ei/ dans le chiffre Zéro, seule l'entrée /z/ (qui précède /ei/) et la sortie /r/ (qui suit /ei/) seront validées.

En fait, les modèles utilisés comportent beaucoup plus de 3 entrées et sorties, ainsi que le montre la figure suivante (I.13) qui schématise un modèle employé pour les voyelles.



Les étiquettes '#', '##' et '\$\$' représentent respectivement la pause entre mots et les silences de début et de fin d'enregistrement.

Chaque entrée ou sortie est associée à un contexte droit ou gauche spécifique, c'est-à-dire à une classe de phonèmes ayant la même influence acoustique sur le son. Les entrées et les sorties modélisent, dans ce cas, les effets de coarticulation, tandis que la cible commune, qui peut-être très courte, voire omise dans le cas des liquides et des semi-voyelles, représente la partie stable d'un phonème. Les états internes peuvent être représentés par plusieurs cibles (voir figure I.13). L'équivalence du code API de chaque étiquette est donnée en annexe 1.