Table des matières

Liste des tableaux

Table des figures

Introduction générale

Le traitement de la parole est un vaste domaine de recherche qui demande l'intervention des experts de plusieurs spécialités. Malgré le développement remarquable des outils et les programmes informatiques, les systèmes à commandes vocales n'ont eu du succès que ces dernières années. Avec l'apparition de la nouvelle génération des smart phones les utilisateurs peuvent parler avec leurs téléphones avec des langues spécifiques. L'absence de la langue arabe parmi ces langues reflète la pauvreté des recherches sur la parole arabe.

Nous avons donc décidé, à partir de ce travail, de nous pencher sur la parole arabe en nous basant sur les travaux réalisés sur les autres langues. Vu que la parole peut être utilisée pour la commande vocale, la dictée, détection Parole/Non Parole, empreinte vocale et autres, nous avons choisi de comparer le traitement avec les mots connectés ; où chaque mot est pris avec sa forme globale, et les mots continus ; où chaque mot est découpé en unités atomiques. Ainsi, nous avons étudié l'influence de la taille de la base d'apprentissage sur le taux de classification dans la parole continue et isolée et mots connectés et continus.

Dans le premier chapitre, nous avons défini des notions linguistiques, le mécanisme de production de la parole, et la méthode MFCC pour automatiser la parole et d'extraire les paramètres pour ensuite faire la classification. Au deuxième chapitre, nous avons présenté les classifieurs célèbres utilisés dans différentes recherches au traitement automatique de la parole. Et enfin, le troisième chapitre, présente les expériences que nous avons réalisée avec les résultats obtenus pour finir avec la présentation d'une calculatrice vocale que nous avons développé en Java.

Chapitre 1

Traitement automatique de la parole

1.1 Introduction

Le traitement automatique des langues (T.A.L.) ou NLP (Natural Language Processing) est un domaine de recherche pluridisciplinaire, qui fait collaborer linguistes, informaticiens, logiciens, psychologues, documentalistes, lexicographes ou traducteurs, et qui appartient au domaine de l'Intelligence artificielle (I.A). Dans le monde nous trouvons plusieurs langues. De chaque langue dérivent plusieurs dialectes. A cet effet le traitement automatique de la parole est un domaine pour lequel un effort important a été approuvé au cours des cinq dernières décennies. Le traitement automatique de la parole ou Speech processing est l'un des filières du traitement automatique de la langue naturelle qui a comme objectif fondamental l'amélioration de la communication Homme-Machine. Selon Shannon dans sa théorie de l'information [1], un message représenté comme une séquence de symboles discrets peut quantifier son contenu d'information en bits, et le débit de transmission de l'information est mesuré en bits par secondes (bps). Mais en traitement de la parole l'information est d'une forme analogique continue « Speech Signal » ce qui est impossible de l'introduire directement dans la machine ; c'est pour cette raison qu'il faut faire des transformations (prétraitements) de numérisation de ce signal afin que nous puissions l'exploiter sur machine.

1.2 Le son naturel

Nous percevons les voix des personnes qui nous entourent, le bruit du vent ou de la cascade, le chant des oiseaux, les bruits de l'activité humaine tels que les moteurs. Nous entendons la musique produite par les instruments de musique, par la radio et les CD et diffusée dans des haut-parleurs, etc. Et si nous tentons d'émettre un son soutenu, une note chantée par exemple, et si nous sommes attentifs, nous sentons des parties du corps vibrer. Cela peut être dans la poitrine, dans le ventre, dans la tête, dans la gorge ou ailleurs. La voix produit des vibrations qui se répercutent dans le corps parce que la voix humaine est elle-même une vibration engendrée par les cordes vocales. Celles-ci vibrent sous l'effet de l'intention mentale. Elles sont mises en action ainsi que le souffle, par notre volonté. Il faut soit affiner notre sensibilité, soit nous mettre dans des conditions un peu excessives pour se rendre compte que tous les sons sont des vibrations. Ainsi, plaçons-nous à proximité d'un haut-parleur qui diffuse une musique très forte, par exemple lors d'un festival de 'Andalous' ou de 'Anachide' en plein air. Nous sentons immédiatement notre ventre vibrer sous l'effet du son. Nous voyons la membrane du haut-parleur vibrer elle aussi. C'est elle qui, par sa vibration, produit le son. Le son produit se propage dans tous les sens avec une vitesse variante selon la nature de l'environnement c'est-à-dire plus la matière est dense, plus la vibration sonore est plus rapide.

1.3 Perception du son

Une vibration mécanique de la matière et de l'air qui met en alternance le tympan ou le micro ne constitue pas en elle-même un son. Car c'est dans le cerveau que naît et se forme le son. Le son n'existe pas en-dehors de notre cerveau, de nous-même. L'oreille recueille les vibrations de l'air, les transforme en impulsion électrique au moyen des cellules nerveuses, impulsion qui est perçue et interprétée en son par le cerveau (fig. 1.1). Le son est donc essentiellement une perception. Si l'attention se dirige vers cette perception, la perception arrive à la conscience. Un son est un phénomène psychique, lié à la conscience des êtres vivants. Entre l'arrivée des signaux vibratoires aux oreilles

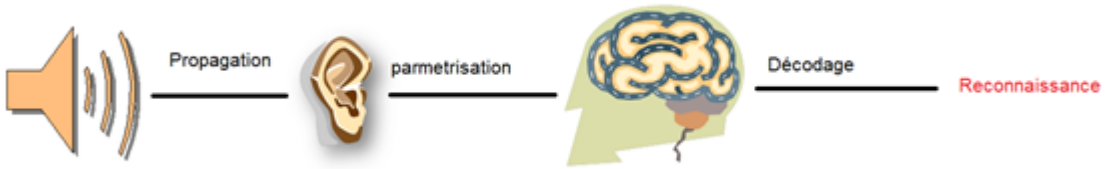

FIGURE 1.1 – Perception et analyse du son par l'être humain

et la sensation de son dans le cerveau, a lieu le phénomène de traitement des signaux par le système nerveux. Cela signifie que la vibration physique de l'air ne parvient pas de façon brute au cerveau. Elle est transformée.

1.4 Système de production de la parole chez l'être humain

La production des sons de parole se fait juste dans la partie mobile du conduit vocal sur laquelle on peut agir volontairement. En partant du bas en haut du conduit vocal (fig. 1.2) nous distinguons l'ensemble des organes suivants :

- Le Trachée : C'est le conduit élastique (fibro-cartilagineux) qui, chez les vertébrés, permet lors de l'inspiration, de conduire l'air depuis le larynx dans les bronches. Elle est constituée d'un épithélium respiratoire ainsi que de cellules musculaires lisses.
- Le larynx : organe essentiel de la phonation, constitue, avec les cordes vocales, la source vocale responsable de la production du flux laryngé (son périodique complexe).
- L'Epiglotte : C'est une structure cartilagineuse reliée au larynx qui coulisse vers le haut quand les voies aériennes sont ouvertes, et aide à fermer l'entrée de la trachée au moment de la déglutition. Elle descend légèrement vers le bas, afin d'entrer en contact avec le larynx qui s'élève, formant ainsi un verrou au-dessus

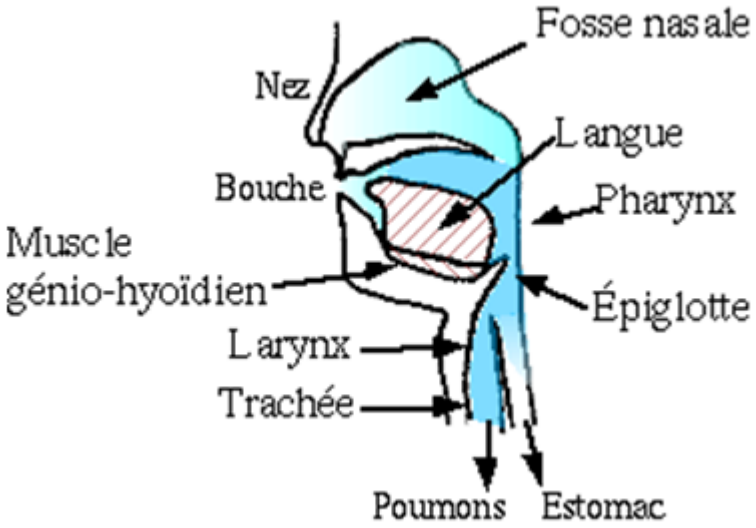

FIGURE 1.2 – Conduit vocal

du larynx.

- Le Pharynx : C'est un organe situé au fond de la cavité buccale, qui a pour rôle de modifier les sons produits dans le larynx par les cordes vocales. L'ouverture de la trompe d'eustache est également située au niveau du pharynx, et le relie à l'oreille interne.
- La Langue : est un organe situé dans la cavité buccale. Il intervient dans la parole par ces mouvements.
- Les dents : Par leurs formes jouent le rôle de filtre des sons venant de l'intérieur
- Les Lèvres : Par leurs mouvements d'ouvertures et de fermetures produisent des sons spécifiques.
- Le Nez : c'est un auxiliaire dans la production de la parole (pour les sons nasaux).

1.5 Phonème et Phonétique

La phonétique est le domaine de la linguistique qui a pour objet l'étude des langues naturelles dans leurs dimensions sonores. Le phonème est la plus petite unité discrète ou que l'on puisse isoler par segmentation dans la chaîne parlée. Un phonème est en

Mots en Arabe	Prononciation	Signification	Représentation syllabique
كَتَبَ	kataba	Il a écrit	CV CV CV
يَكتُبُ	i :aktobo	Il écrit	CVC CV CV
كَاتِبٌ	ka :tibon	écrivain	CV CV CVC
جَمِيلٌ	jami :lon	beau	CV CV CVC
صَبرٍ	ṣabr	patience	CVCC

TABLE 1.1 – exemple d'analyse syllabique de quelques mots arabes

réalité une entité abstraite, qui peut correspondre à plusieurs sons. Il est en effet susceptible d'être prononcé de façon différente selon les locuteurs ou selon sa position et son environnement au sein du mot. Les phones sont d'ailleurs les différentes réalisations d'un phonème. L'arabe classique standard a 34 phonèmes parmi lesquels 6 sont des voyelles et 28 sont des consonnes [2]. Les phonèmes arabes se distinguent par la présence de deux classes qui sont appelées pharyngales et emphatiques. Ces deux classes sont caractéristiques des langues sémitiques comme l'hébreu [2][3]. Les syllabes permises dans la langue arabe sont : CV, CVC et CVCC [4] . Où le V désigne une voyelle courte ou longue et le C représente une consonne [2]. La langue arabe comporte cinq types de syllabes classées selon les trais ouvert/fermé et court/long. Une syllabe est dite ouverte (respectivement fermée) si elle se termine par une voyelle (respectivement une consonne). Toutes les syllabes commencent par une consonne suivie d'une voyelle et elles comportent une seule voyelle. La syllabe CV peut se trouver au début, au milieu ou à la fin du mot [3] [5]. Le tableau (tab. 1.1) représente quelques exemples de mots arabes avec leurs prononciation en Alphabet Phonétique Internationale[6]

1.6 Traitement du signal vocal

L'information contenue dans le signal de parole peut être analysée de bien des façons. Si nous observons la forme que produit la parole selon l'audiogramme présenté par la figure (fig. 1.3) nous remarquons une forme périodique avec des amplitudes variantes

ou pseudopériodiques. Ainsi, aux cotés droit et gauche du signal principal nous distin-

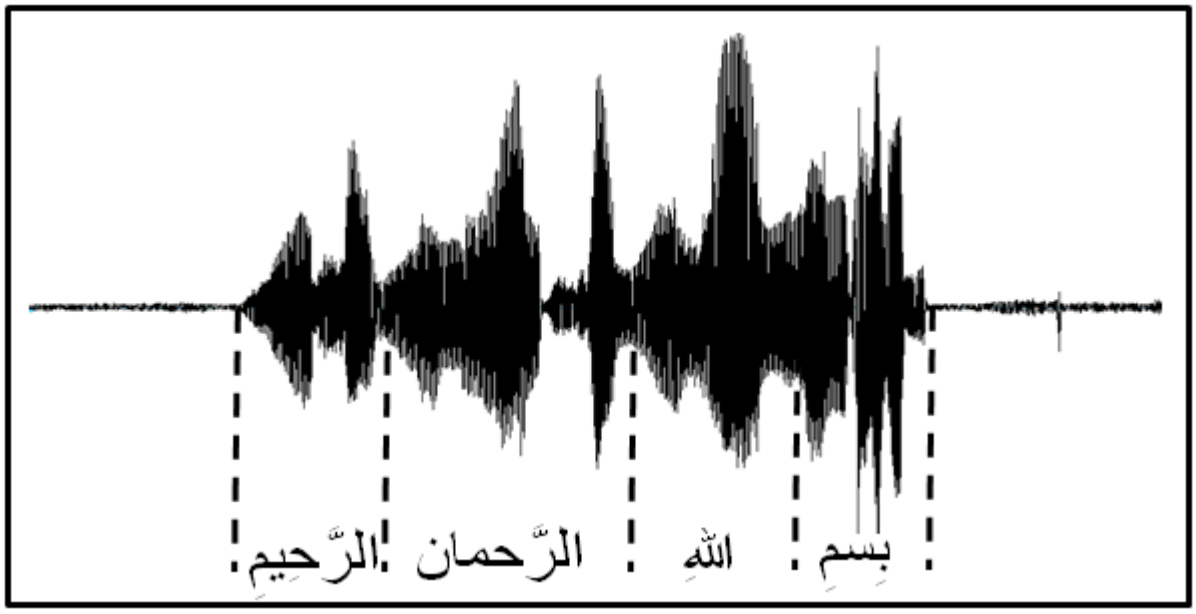

FIGURE 1.3 – Audiogramme d'un signal vocal

guons des petites courbes non identifiées, ce que nous appelons le bruit. Il y a plusieurs travaux sur le sujet de reconnaissance de parole/ non parole basés sur le bruit (Speech/-NONSpeech) [7]. En plus, chaque individu possède sa propre information vocale qui le caractérise. Et cette information peut être extraite à partir des signaux sortant du résonateur. Les traits acoustiques du signal de parole sont directement liés à sa production dans l'appareil phonatoire. Tout d'abord, nous avons l'énergie du son [8] ; celle-ci est liée à la pression de l'air en amont du larynx. Puis nous avons la fréquence fondamentale F0 [9] ; cette fréquence correspond à la fréquence du cycle d'ouverture/fermeture des cordes vocales. Enfin, nous avons le spectre du signal de parole [10] ; celui-ci résulte du filtrage dynamique du signal en provenance du larynx par le conduit vocal qui peut être considéré comme une succession de tubes ou de cavités acoustiques de sections diverses (fig. 1.4). Chacun de ces traits acoustiques est lui-même intimement lié à une autre grandeur perceptuelle, à savoir l'intensité, le rythme, et le timbre. Le spectrogramme est la représentation temps-fréquence qui permet de mettre en évidence les différentes composantes fréquentielles du signal à un instant donné. L'ensemble des spectres consti-

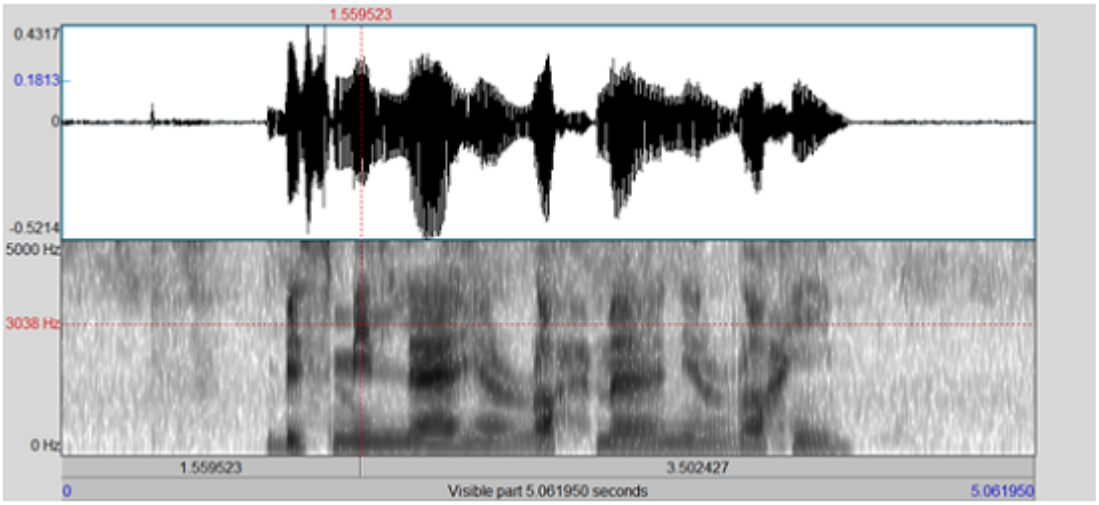

FIGURE 1.4 – un signal vocal et la spectogramme associé

tuant le spectrogramme sont calculé par la transformé de Fourier que nous allons voir plus en détails par la suite.

1.6.1 Intensité d'un signal vocal

L'intensité d'un son, appelée aussi volume, permet de distinguer un son fort d'un son faible. Elle correspond à l'amplitude de l'onde. L'amplitude est donnée par l'écart maximal de la grandeur qui caractérise l'onde. Pour le son, onde de compression, cette grandeur est la pression. L'amplitude sera donc donnée par l'écart entre la pression la plus forte et la plus faible exercée par l'onde acoustique. Lorsque l'amplitude de l'onde est grande, l'intensité est grande et donc le son est plus fort. L'intensité du son se mesure en décibels (dB). On distingue différentes façons de mesurer l'amplitude d'un son :

- La puissance acoustique : La puissance acoustique est associée à une notion physique. Il s'agit de l'énergie transportée par l'onde sonore par unité de temps et de surface. Elle s'exprime en Watt par mètre carré ($W.m^{-2}$).
- Addition de sons : L'échelle des décibels est une échelle dite logarithmique, ce qui signifie qu'un doublement de la pression sonore implique une augmentation de l'indice d'environ 3 : avec 3 dB de plus, l'intensité est en fait doublée [11].

1.6.2 Le rythme

Le rythme est la durée des silences et des phones. Il est difficile de les en extraire car un mot prononcé d'une façon naturelle, sans aucun traitement, donne un mélange de phones chevauchés entre eux et un silence d'intensité non nulle (le bruit).

1.6.3 Le timbre

Le timbre est l'ensemble des caractéristiques qui permettent de différencier une voix. Il provient en particulier de la résonance dans la poitrine, la gorge la cavité buccale et le nez ; ce sont les amplitudes relatives des harmoniques du fondamental qui déterminent le timbre du son. Les éléments physiques du timbre comprennent :

- la répartition des fréquences dans le spectre sonore,
- les relations entre les parties du spectre, harmoniques ou non,
- les bruits existant dans le son (qui n'ont pas de fréquence particulière, mais dont l'énergie est limitée à une ou plusieurs bandes de fréquence),
- l'évolution dynamique globale du son,
- l'évolution dynamique de chacun des éléments les uns par rapport aux autres.

1.7 Automatisation de la Parole

La parole est produite par l'articulation des membres phonatoires de l'homme et prend une forme analogique apériodique ; ce qui est impossible pour que la machine puisse l'interpréter ou le prédire car elle ne comprend que du numérique. Pour cela on doit faire un traitement de numérisation sur ce signal. L'une des méthodes les plus utilisés dans la numérisation est la méthode Delta ou MIC qui consiste en trois étapes : l'échantillonnage, la quantification et le codage.

1.7.1 L'échantillonnage

L'échantillonnage consiste à transformer une fonction $a(t)$ à valeurs continues en une fonction $\hat{a}(t)$ discrète constituée par la suite des valeurs $a(t)$ aux instants d'échan-

tillonnage $t = kT$ avec k un entier naturel (fig. 1.5). Le choix de la fréquence d'échantillonnage n'est pas aléatoire car une petite fréquence nous donne une présentation pauvre du signal. Par contre une très grande fréquence nous donne des mêmes valeurs, redondance, de certains échantillons voisins donc il faut prélever suffisamment de valeurs pour ne pas perdre l'information contenue dans $a(t)$. Le théorème suivant traite cette problématique :

Théorème (de Shannon)**.** *La fréquence d'échantillonnage assurant un non repliement du spectre doit être supérieure à 2 fois la fréquence haute du spectre du signal analogique.*

$$F_{ech} = 2 \times F_{max}$$

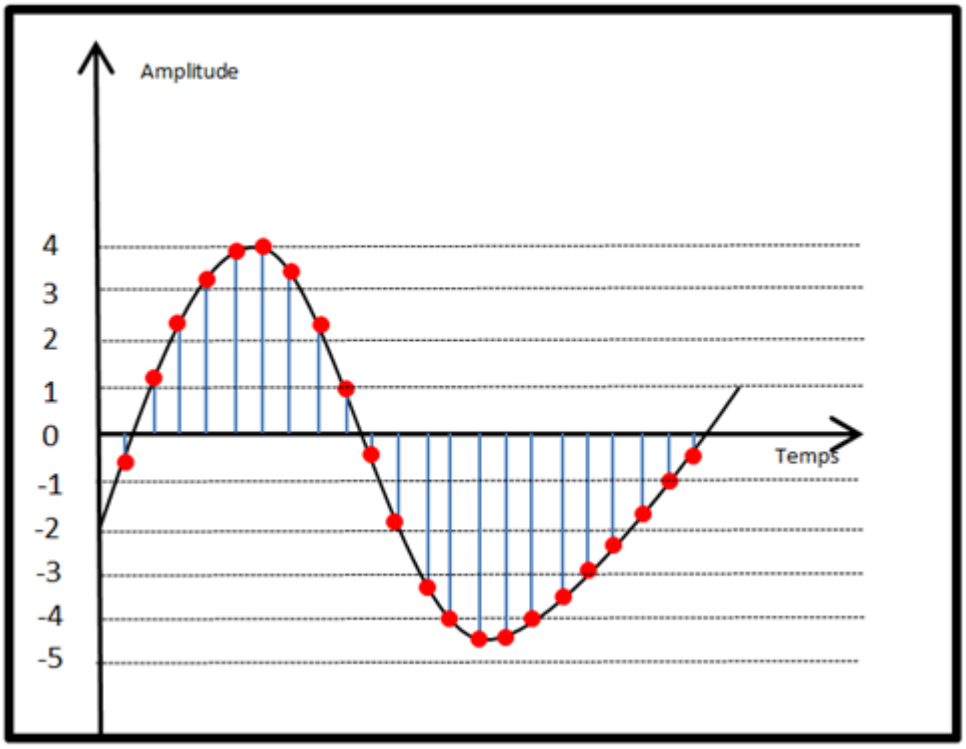

FIGURE 1.5 – un signal échantillonné

Pour la téléphonie, on estime que le signal garde une qualité suffisante lorsque son spectre est limité à 3400 Hz et l'on choisit fe = 8000 Hz. Pour les techniques d'analyse, de synthèse ou de reconnaissance de la parole, la fréquence peut varier de 6000 à 16000 Hz. Par contre pour le signal audio (parole et musique), on exige une bonne représentation

du signal jusque 20 kHz et l'on utilise des fréquences d'échantillonnage de 44.1 ou 48 kHz. Pour les applications multimédia, les fréquences sous-multiples de 44.1 kHz sont de plus en plus utilisées : 22.5 kHz, 11.25 kHz [12].

1.7.2 Quantification

Cette étape consiste à approximer les valeurs réelles des échantillons selon une échelle de n niveaux appelée échelle de quantification. Il y a donc 2^n valeurs possibles comprises entre $-2n-1$ et $2n-1$ pour les échantillons quantifiés (fig. 1.6). L'erreur systématique que l'on commet en assimilant les valeurs réelles de l'écart au niveau du quantifiant le plus proche est appelé bruit de quantification.

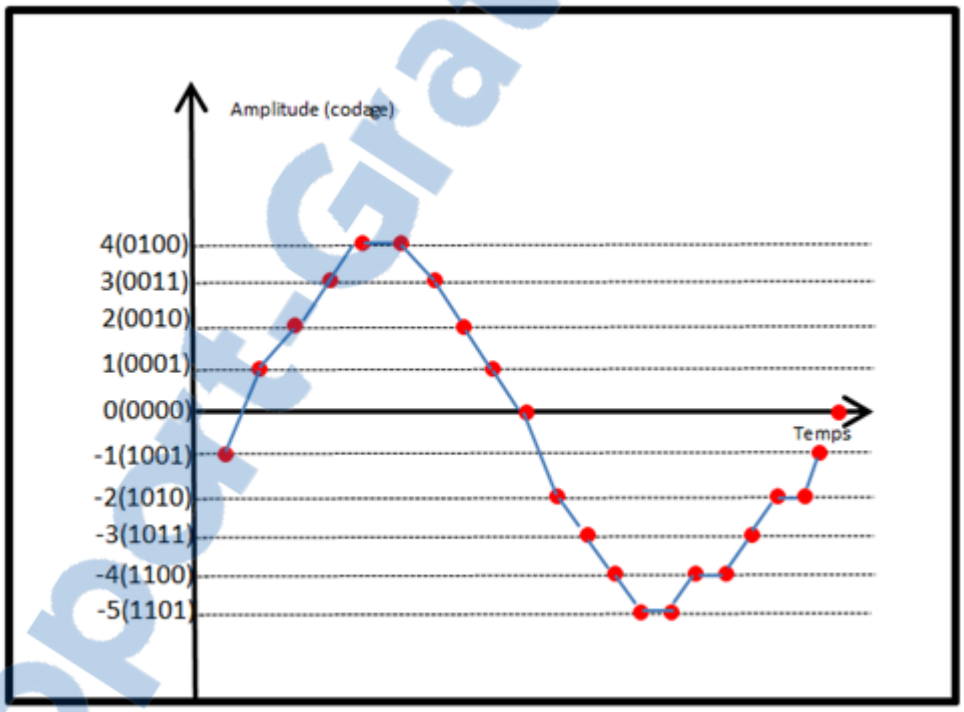

FIGURE 1.6 – un signal quantifié

1.7.3 Codage

C'est la représentation binaire des valeurs quantifiées qui permet le traitement du signal sur machine (fig. 1.6).

1.8 Paramétrisation du signal vocal

L'objectif de cette phase de reconnaissance est d'extraire des coefficients représentatifs du signale de la parole. Ces coefficients sont calculés à intervalles réguliers. En simplifiant les choses, le signal de la parole est transformé en une série de vecteurs de coefficients, ces coefficients doivent représenter au mieux ce qu'ils sont censé modéliser et doivent extraire le maximum d'informations utiles pour la reconnaissance. Parmi les coefficients les plus utilisés et qui représentent au mieux le signal de la parole, nous trouvons les coefficients ceptraux, appelés également ceptres. Les deux méthodes les plus connus pour l'extraction du ceptres sont : l'analyse spectrale et l'analyse paramétrique. Pour l'analyse spectrale (par exemple, Mel-Scale Frequency Ceptral Coefficients (MFCC)) comme pour l'analyse paramétrique (par exemple, le codage prédictif linéaire (LPC)), le signal de parole est transformé en une série de vecteurs calculés pour chaque trame. Il existe d'autres types de coefficients qui sont surtout utilisés dans des milieux bruités, nous citons par exemple les coefficients PLP (Perceptual Linear Predective). Ces coefficients permettent d'estimer les paramètres d'un filtre autorégressif en modélisant au mieux le spectre auditif [13]. Il existe plusieurs techniques permettant l'amélioration de la qualité des coefficients, nous trouvons par exemple ; l'analyse discriminante linéaire (LDA), l'analyse discriminante non linéaire (NLDA), etc.[14] Ces coefficients jouent un rôle capital dans les approches utilisées pour la reconnaissance de la parole. En effet, ces paramètres qui modélisent le signal seront fournis au système de reconnaissance pour l'estimation de la probabilité $P(séquence|message)$. Dans notre travail, nous utilisons les coefficients MFCC pour tester leur rendement dans un environnement bruité. L'utilisation des MFCC est motivée par les deux propriétés suivantes :

- Déconvolution : les MFCC découplent les caractéristiques du conduit vocal (qui véhicule la plus grande partie de l'information disponible sur les traits distinctifs de la parole) des caractéristiques générées par l'excitation (information prosodique et l'information dépendante du locuteur).
- Décorrélation : La transformée en cosinus discrète possède un effet de décorrélation entre les éléments du vecteur de traits. Les MFCC sont une représentation

définie comme étant la transformée cosinus inverse du logarithme du spectre de l'énergie du segment de la parole. L'énergie spectrale est calculée en appliquant un banc de filtres uniformément espacés sur une échelle fréquentielle modifiée, appelée échelle Mel. L'échelle Mel redistribue les fréquences selon une échelle non linéaire qui simule la perception humaine des sons.[15]

Étapes de calcul du vecteur caractéristique de types MFCC :

Dans ce qui suit, nous décrivons chacune des étapes nécessaires pour l'obtention d'un vecteur caractéristique tiré des coefficients MFCC, tel qu'illustré par la Figure (fig. 1.7)

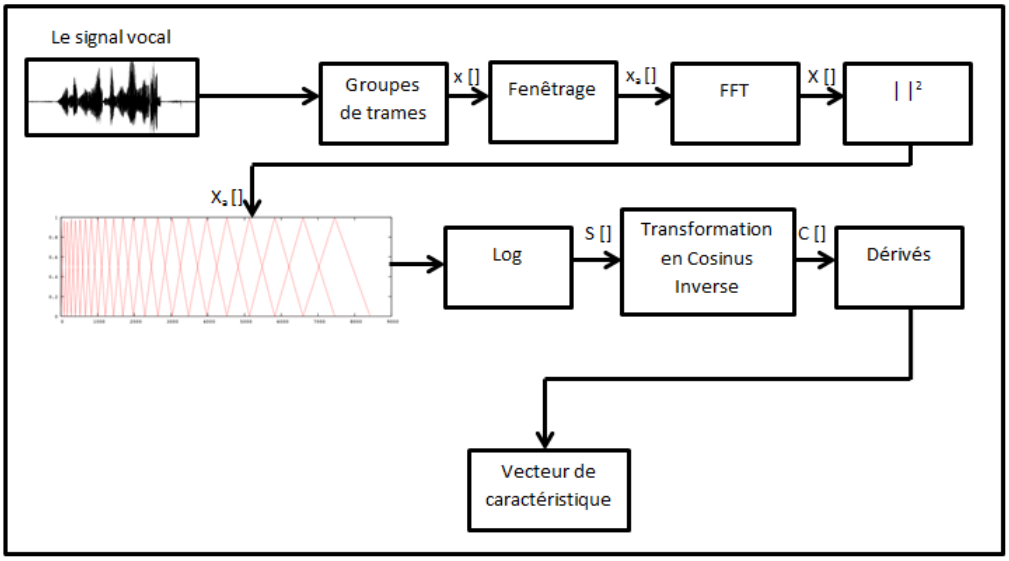

FIGURE 1.7 – Étapes de calcul d'un vecteur caractéristique de type MFCC

1.8.1 Groupement en trames *(Frame blocking)*

Le signal acoustique continu est segmenté en trames de N échantillons, avec un pas d'avancement de M trames ($M < N$), c'est-à-dire que deux trames consécutives se chevauchent sur $N - M$ échantillons. Les valeurs couramment utilisées pour M et N sont respectivement 10 et 20. Comme prétraitement, il est d'usage de procéder à la préaccentuation du signal en appliquant l'équation de différence du premier ordre aux

échantillons $x(n)$, avec l'équation (1.1)

$$x'(n) = x(n) - kx(n-1), \quad 0 < n < N-1 \tag{1.1}$$

k représente un coefficient de préaccentuation qui peut prendre une valeur dans l'étendue $0 < k < 1$.

1.8.2 Fenêtrage

Si nous définissons $w(n)$ comme fenêtre où $0 < n < N-1$ et N représente le nombre d'échantillons dans chacune des trames, alors le résultat du fenêtrage est le signal x_a, donné par la formule (1.2).

$$x_a = x(n)w(n), \quad 0 < n < N-1 \tag{1.2}$$

Les fenêtres les plus utilisées sont :

– Fenêtre de Hamming :(1.3)

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) & 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \tag{1.3}$$

– Fenêtre rectangulaire :(1.4)

$$w(n) = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \tag{1.4}$$

– Fenêtre triangulaire :(1.5)

$$w(n) = \begin{cases} \frac{2n}{N-1} & si 0 \leq n \leq \frac{N-1}{2} \\ \frac{2(N-n-1)}{N-1} & si \frac{N-1}{2} < n \leq N-1 \\ 0 & sinon \end{cases} \tag{1.5}$$

– Fenêtre de Hann :(1.6)

$$w(n) = \begin{cases} 0.5 - 0.5\cos\frac{2\pi n}{N-1} & si 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \tag{1.6}$$

– Fenêtre de Blackman :(1.7)

$$w(n) = \begin{cases} 0.42 - 0.5\cos\frac{2\pi n}{N-1} + 0.08\cos\frac{4\pi n}{N-1} & si 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \tag{1.7}$$

La figure (fig. 1.8) illustre la forme que prennent les fonctions définies ci-dessus

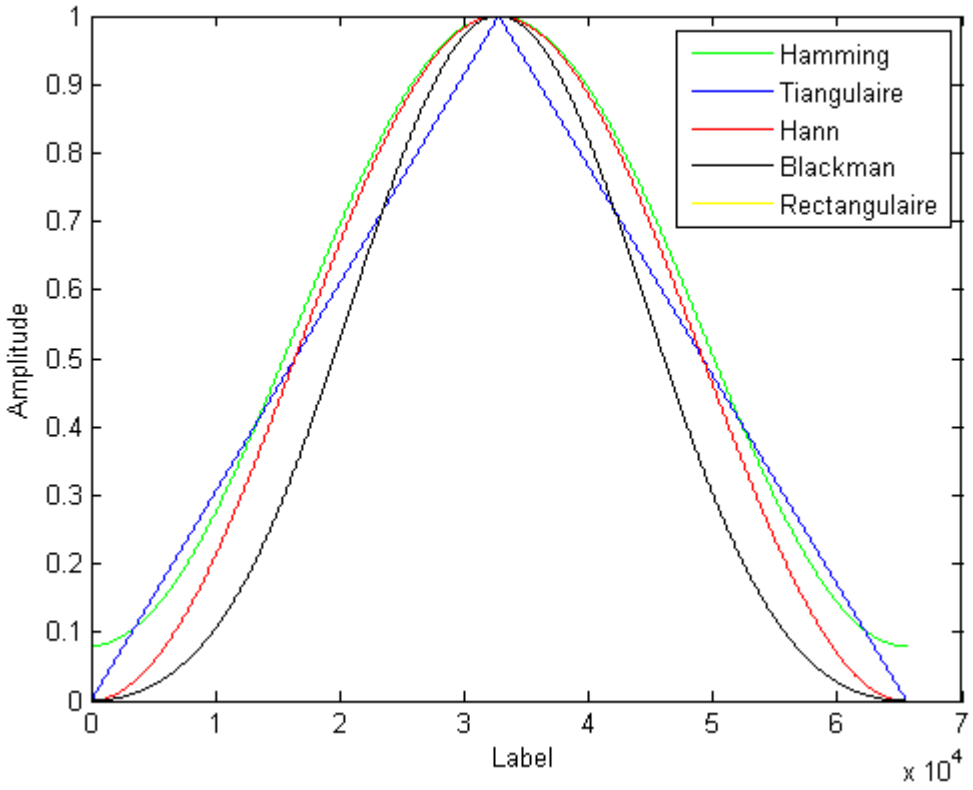

FIGURE 1.8 – Les fonctions de fenêtrage

1.8.3 Calcul de la transformée de Fourier rapide *(Fast Fourier Transform, FFT)*

Au cours de cette étape chacune des trames, de N valeurs, est convertie du domaine temporel au domaine fréquentiel. La FFT est un algorithme rapide pour le calcul de la transformée de Fourier discret (DFT) et est définie par la formule (1.8). Les valeurs obtenues sont appelées le spectre.

$$x[k] = \sum_{n=0}^{N-1} x_a[n] e^{\frac{-2j\pi}{N} kn}, \quad 0 \leq k \leq N-1 \tag{1.8}$$

En général, les valeurs $X[k]$ sont des nombres complexes et nous nous utilisons que leurs valeurs absolues (énergie de la fréquence).

1.8.4 Filtrage sur l'échelle Mel

Le spectre d'amplitude est pondéré par un banc de M filtres triangulaires espacés selon l'échelle Mel. Dans l'échelle de mesure Mel, la correspondance est approximativement linéaire sur les fréquences au-dessous de $1kHz$ et logarithmique sur les fréquences supérieures à celle-ci. Cette relation est donnée par la formule (1.9) [16] :

$$m = 2595 \log_{10}(1 + \frac{f}{700}) \tag{1.9}$$

Le logarithme de l'énergie de chaque filtre est calculé selon l'équation 1.10 :

$$S[m] = \ln[\sum_{k=0}^{N-1} X_a[k] H_m[k]], \qquad 0 < m \leq M \tag{1.10}$$

1.8.5 Calcul du cepstre sur l'échelle Mel

Le cepstre sur l'échelle de fréquence Mel est obtenu par le calcul de la transformée en cosinus discrète (equation (1.11)) du logarithme de la sortie des M filtres (reconversion du log-Mel-spectre vers le domaine temporel).

$$c[n] = \sum S[n] \cos \pi n (m - \frac{1}{2})/M, \qquad 0 \leq n < M \tag{1.11}$$

Le premier coefficient, $c[0]$, représente l'énergie moyenne dans la trame de la parole ; $c[1]$ reflète la balance d'énergie entre les basses et hautes fréquences ; pour $i > 1$, $c[i]$ représente des détails spectraux de plus en plus fins [16].

1.8.6 Calcul des caractéristiques dynamiques des MFCC

Les changements temporels dans le cepstre (c) jouent un rôle important dans la perception humaine et c'est à travers les dérivées des coefficients ($\triangle_c$, coefficients delta ou vélocité) et les dérivées secondes ($\triangle\triangle_c$, coefficients delta du second ordre ou accélération) des MFFC statiques que nous pouvons mesurer ces changements. En résumé, un système de parole typique de l'état de l'art effectue premièrement un échantillonnage à une fréquence de 16 kHz et extrait les traits suivants [17]

$$\begin{pmatrix} c_k \\ \triangle c_k \\ \triangle \triangle c_k \end{pmatrix}$$

Où :

- c_k est le vecteur MFCC de la k$^{\text{ième}}$trame
- $\triangle c_k = c_{k+2} - \triangle c_{k-2}$, dérivée première des MFCCs calculée à partir des vecteurs MFCC de la k$^{\text{ième}}$ + 2 trames et k$^{\text{ième}}$ - 2
- $\triangle \triangle c_k = \triangle c_{k-1} - \triangle c_{k+1}$, seconde dérivée des MFCCs.

1.9 conclusion

Le traitement automatique de la parole repose sur des données analogiques en fonction du temps. L'extraction des meilleurs paramètres aide, sans aucun doute, à ce traitement.

L'intelligence artificielle peut intervenir pour trouver les paramètres pertinents ou utiliser n'importe quels représentants de la parole pour faire la segmentation ou la classification.

Chapitre 2

Techniques de classification

2.1 Introduction

La classification est une partie de l'intelligence artificielle qui rend le comportement de la machine plus intelligent. Tout classifieur assurant la classification nécessite la définition des classes, des attributs, l'algorithme de décision et un moyen pour mesurer ses performances ; à partir d'un ensemble de règles d'état explicites ou bien à travers des exemples d'apprentissage.
Le traitement de la parole est un vaste domaine de recherche auquel plusieurs travaux ont été faits pour trouver les meilleures techniques qui donnent de meilleurs taux de classification sur différents types de traitements que ce soit parole continue ou discrète, basé sur les mots connectés ou mots continus, petit ou grand vocabulaire.

2.2 Chaînes de Markov cachés

2.2.1 Les processus stochastiques

Un processus stochastique est une fonction, ou plus généralement une application $X(\omega, t)$, définie dans l'ensemble fondamental Ω à valeurs dans $F(t)$, ensemble des fonctions d'une variable t. L'évolution d'un processus stochastique est une suite de transitions d'états : $s_0 s_1 \dots s_T$, pour laquelle on note s_0 l'état du processus à l'instant 0. Sa

loi d'évolution est obtenue à l'aide de la probabilité $P(s_0 \dots s_T)$ définie successivement de la manière suivante (eq. (2.1)) :

$$P(s_0...s_T) = P(s_0) \times P(s_1|s_0) \times P(s_2|s_0s_1) \times ... \times P(s_T|s_0...s_{T-1}) \qquad (2.1)$$

La caractérisation du processus se résume donc par l'obtention des probabilités initiales $P(s_0)$ et des probabilités des états conditionnés par leurs évolutions antérieures. La loi de probabilité des états, à un instant t, dépend de l'histoire du processus qui garde la mémoire de son passé. L'espace des états S est l'ensemble dénombrable des valeurs prises par l'ensemble des variables aléatoires du processus stochastique. Ces valeurs, tout comme celles prises dans l'espace du temps T, peuvent être discrètes ou continues, ce qui permet de les classer respectivement par rapport à ω et t [18] :

- T et S sont continus : $X(\omega, t)$ est continu, on parle alors de processus de renouvellement ou de diffusion.
- T est continu, S est discret : $X(\omega, t)$ discontinu en ω, pour l'étude des files d'attente.
- T est discret, S est continu : $X(\omega, t)$ discontinu en t, pour l'étude des séries temporelles.
- T et S sont discrets : $X(\omega, t)$ est discontinu en ω et en t, ce sont les processus markovien ou chaînes de Markov qui nous intéressent particulièrement.[19]

2.2.2 Les modèles de Markov

Les modèles de Markov cachés (Hidden Markov Models ou HMMs) ont été introduits par Baum et al. À la fin des années 60. Un HMM est un processus stochastique défini par le quintuplé $\lambda = (S, \delta, T, G, \pi)$ où :

- S : est un ensemble de N états,
- δ : est un alphabet de M symboles,
- $T = S \times S \rightarrow [0, 1]$ est la matrice de transition, indiquant les probabilités de transition d'un état à l'autre ; on note $P(s \rightarrow s_0)$ la probabilité de transition de l'état s vers l'état s_0,
- $G = S \times \delta \rightarrow [0, 1]$ est la matrice de génération, indiquant les probabilités de

génération associées aux états ; on note $P(o|s)$ la probabilité de générer le symbole o appartenant à δ à partir de l'état $s \in S$.

– $\pi : S \rightarrow [0, 1]$ est un vecteur de probabilités initiales de visite.

Il n'y a pas de règle stricte pour choisir l'architecture du HMM, par conséquent nous trouvons des travaux sur l'apprentissage dynamique du nombre d'états d'un Modèle de Markov Caché à observations continues au traitement de signal et au traitement d'images[20][21].

La procédure de génération d'une séquence $o_1...o_T$ de symboles à l'aide d'un HMM consiste à partir d'un état s en suivant la distribution π, de se déplacer d'état en état suivant les probabilités de transition, et générer un symbole sur chaque état rencontré en utilisant la distribution de probabilité de génération associée à l'état. Lorsqu'un symbole a été généré, on choisit une transition sortante suivant la distribution de probabilité de transition associée à l'état courant, et la procédure est réitérée jusqu'à la $T^{\text{ième}}$ génération de symbole (fig. 2.1).[22][23]

Au traitement de la parole la suite d'états cachés est la suite des paramètres tirés des

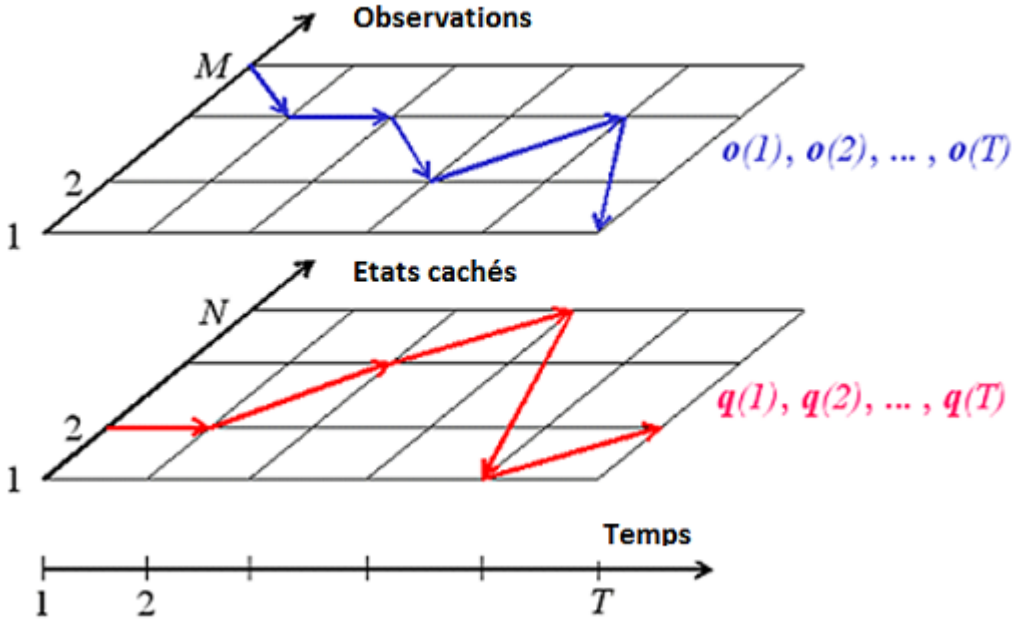

FIGURE 2.1 – Séquences observés et cachées

données audio qui caractérisent le spectre de la parole.[24]

2.2.3 Les problèmes fondamentaux d'un HMM

Pour qu'un HMM puisse être utilisé efficacement dans les applications réelles il faut bien définir sa topologie et les paramètres des quintuplé vus précédemment. A partir de ce point les spécialistes ont tirés trois problèmes : l'évaluation, décodage, et l'apprentissage.

- L'évaluation : c'est le fait de trouver l'évaluation d'une probabilité $P(O|\lambda)$ de la suite d'observations O selon le modèle λ
- Décodage : C'est l'estimation de la suite d'états cachés appartenant à S sachant qu'on a l'ensemble d'observations O et le modèle λ
- L'apprentissage : C'est le problème d'ajustement des paramètres du modèle λ pour maximiser la probabilité $P(O|\lambda)$.

2.2.4 L'algorithme FORWARD

Soit $\alpha_t(i)$ la probabilité de la séquence d'observation partielle $O_t = o(1), o(2), ..., o(t)$ produite par l'ensemble des séquences d'états possibles qui se terminent au $i^{\text{ème}}$ état.

$$\alpha_t(i) = P(o(1), o(2), ..., o(t)|Q(t) = q_i, \lambda).$$

Puis la probabilité inconditionnelle de la séquence partielle d'observation est la somme de $P_t(i)$ sur tous les états N. L'algorithme Forward est un algorithme récursif pour calculer $\alpha_t(i)$ pour la séquence d'observation à l'instant t. Tout d'abord, on calcule la probabilité de générer le premier symbole de la séquence par la formule $\alpha_t(i) = \pi(i).P(o1|i)$, puis à chaque étape de l'induction,$\alpha_t(i) = (\sum_{i' \in S} \alpha_{t-1}(i').P(i' \to i)P(o_t|i)$ on rajoute un symbole et on réitère la procédure jusqu'à ce que l'on ait calcule la probabilité de génération de la séquence entière et par la suite $P(O|\lambda)$ par la formule

$$P(O|\lambda) = \sum_{i \in S} \alpha_T(i)$$

2.2.5 L'algorithme BACKWARD

C'est un algorithme qui peut être utilisé pour faire l'opération inverse de l'algorithme FORWARD. On utilise alors la variable backward définie par

$$\beta_t(i) = P(o_{t+1}, o_{t+2}, ..., o_T | i_t = s, \lambda)$$

qui exprime la probabilité de générer la séquence $O = o_{t+1}...o_T$ en partant de l'état s. L'induction suit alors le schéma :

1. initialisation : $\beta_T(i) = 1$
2. induction : $\beta_t(i) = \sum_{i' \in S} \beta_{t+1}(i') P(i \to i') P(o_{t+1}|i')$

En connaissant la probabilité de générer la séquence O en partant de l'état s, le calcul de P(O | H) peut alors être réalisé suivant la formule.[23]

$$P(O|\lambda) = \sum_{i \in S} \pi(i)\beta_1(i)$$

2.2.6 L'Algorithme de Viterbi

Afin de résoudre le problème de décodage, l'algorithme de Viterbi est employé. Le critère d'optimalité ici est de rechercher un meilleur ordre simple d'état par la technique modifiée de la programmation dynamique. L'algorithme de Viterbi est un algorithme de recherche parallèle, à savoir il recherche le meilleur ordre d'état en traitant tous les états en parallèle. Nous devons maximiser $P(Q|O, \lambda)$ pour détecter le meilleur ordre d'état. Soie la probabilité $\delta_t(i)$ qui représente la probabilité maximale le long du meilleur chemin probable d'ordre d'état d'une séquence d'observation donné après t instants et en étant à l'état i ;

$$\delta_t(i) = \max_{q_1, q_2 \cdots q_{t-1}} P[q_1, q_2...q_{t-1}, q_t = S_i, o_1...o_t|\lambda]$$

La meilleure séquence d'états et retournée par une autre fonction $\psi_t(j)$. Cette fonction tient l'index de l'instant $t-1$, à partir duquel la meilleure transition est faite à l'état actuel. L'algorithme complet est comme suit :

1. Initialisation : $\psi_1(i) = 0; \quad \delta_1(i) = \pi(i)P(o_1|i);$

2. Induction :

$$\delta_t(i) = \max_{i' \in S}(\delta_{t-1}(i')P(i' \to i))P(o_t|i)$$

$$\psi_t(i) = arg\max_{i' \in S}(\delta_{t-1}(i')P(i' \to i))$$

Une fois les variables $\delta_t(i)$ et $\psi_t(j)$ calculées pour chaque étape de l'induction et pour chaque état, il ne reste plus qu'à lancer une procédure inductive de retro-propagation pour "dérouler" le chemin de Viterbi ${s_1}^*...{s_T}^*$:

1. Initialisation : ${s_T}^* = arg\max_{i \in S}(\delta_T(i))$
2. Induction : ${s_t}^* = \psi_{t+1}({s_{t+1}}^*), \quad t \in \{T-1...1\}$

Cet algorithme a eu beaucoup d'extensions [25], parmi lesquels nous allons voir l'algorithme à passage de jeton.

2.2.7 L'algorithme de Baum-Welch

Cet algorithme est lié au problème d'apprentissage qui est le plus difficile. Le but est d'ajuster des paramètres du modèle selon un critère d'optimalité. L'algorithme Baum-Welch est strictement lié à l'algorithme FORWARD-BACKWARD et il essaye d'atteindre le maximum local de la fonction de probabilité $P(O|\lambda)$. Le modèle converge toujours mais la maximisation globale n'est pas garantie. C'est pourquoi le point initial de recherche est très important. Soit

$$\xi_t(i,i') = \frac{P(i_t = i, i_{t+1} = i'|O,\lambda)}{P(O|\lambda)}$$

La probabilité qu'en générant O avec λ on passe par l'état i à l'instant t et par l'état i_0 à l'instant $t+1$. et en utilisant les variables forward et backward :

$$\xi_t(i,i') = \frac{\alpha_t(i)P(i \to i')P(o_{t+1}|i')\beta_{t+1}(i')}{P(O|\lambda)} = \frac{\alpha_t(i)P(i \to i')P(o_{t+1}|i')\beta_{t+1}(i')}{\sum_{q \in S}\sum_{r \in S}\alpha_t(q)P(q \to r)P(o_{t+1}|r)\beta_{t+1}(r)}$$

On définit ainsi la quantité $\gamma_t(i) = P(i_t = i|O,H)$ la probabilité qu'en générant O avec H on se trouve sur l'état s à l'instant t, on a :

$$\gamma_t(i) = \sum_{i' \in S}\xi_t(i,i')$$

Si l'on somme $\gamma_t(i)$ sur l'ensemble des instants t, on obtient une quantité que l'on peut interpréter comme l'espérance du nombre de fois où l'état i est utilisé pour générer la séquence O. De même, si on somme $\xi_t(i, i_0)$ sur l'ensemble des instants t, on obtient une quantité que l'on peut interpréter comme l'espérance du nombre de fois où la transition $s \to s_0$ est utilisée pour générer la séquence O. On a donc un estimateur $\hat{H}$ du HMM défini par les expressions suivantes :

$$\hat{\pi}(i) = \gamma_1(i)$$

$$\hat{P}(i \to i') = \frac{\sum_{t=1}^{T-1} \xi_t(i, i')}{\sum_{t=1}^{T-1} \gamma_t(i')}$$

$$\hat{P}(o|i) = \frac{\sum_{t=1, o_t=o}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)}$$

Après la re-estimation des paramètres du modèle, nous allons avoir un nouveau modèle plus adapté à générer la séquence d'observation O. Le procédé itératif de re-estimation continue jusqu'à ce qu'aucune amélioration de $P(O|\lambda)$ ne soit réalisée.[19]

2.2.8 Algorithme à passage de Jeton (Token passing algorithm)

Introduit par Young en 1989 [26], l'algorithme à passage de jeton est une amélioration du décodage de Viterbi qui se base sur la DTW, or cette dernière fait que des calculs et des comparaisons et en conséquence, par exemple au traitement de la parole continue, une fausse décision à un instant t induit un faut résultat finale. L'avantage de l'algorithme à passage de jeton est qu'il fait une recherche parallèle en profondeur avec des retours en arrière des jetons. L'algorithme est présenté comme suit (fig. 2.2)[27] :

1. Initialisation

Tous les jetons des états initiaux des modèles prennent la valeur 0 ;

Tous les autres jetons des autres états sont marqués à ∞

2. Récursion

pour t=1 à T faire

pour chaque état i faire

Passer une copie du jeton dans l'état i à tous les états voisins j, en incrémentant sa valeur $\delta_t(j)$ par $P(i \rightarrow j)P(o_t|j)$;

fin

pour chaque marque propagée par l'intermédiaire d'un arc externe au temps t faire créer un nouvel lien de mot contenant {contenu symbolique, t, identité de mot précédant}

fin

Oublier les jetons originaux ;

pour chaque état i faire

trouver le jeton dans l'état i avec la plus petite valeur et jeter le reste ;

fin

FIGURE 2.2 – Token Passing Algorithm

2.2.9 Les limites des HMMs

Il devrait noter ici que les HMMs ont quelques limitations :

1. La probabilité de transition dépend seulement de l'origine et de la destination.
2. Le choix à priori de la topologie des modèles (nombre d'états, transitions autorisées et règles de transition) limite la souplesse des modèles
3. Ignorance complète de la durée relative des événements du signal.
4. Dégradation des performances s'il y a problème à l'apprentissage.

Certains recherches [28][29] ont trouvés que l'hybridation des HMMs avec les réseaux de neurones artificiels a donnée de meilleurs résultats avec un taux de reconnaissance

supérieur à 85.8 % par rapport à 83.4% d'un HMM simple. Nous trouvons aussi des extensions des HMMs par la notion de Hidden semi-Markov model [30]avec la redéfinition de ses propre algorithmes d'estimation, d'apprentissage et de paramétrisation[31] ,etc.

2.3 Support Vector Machines (SVM)

Introduite au début des années 90 par Vladimir Vapnik et qui connaît jusqu'à nos jours un très grand succès dans la reconnaissance des formes. Elle repose sur une théorie solide d'apprentissage statistique qui vise à trouver des hyperplans séparant les données dans un espace approprié des caractéristiques[32]. Et en conséquence elle donne une solution aux limites des classifieurs à séparation linéaire par des séparations basée sur les hyperplans (fig. 2.3). Selon Jaume Padrell-Sendra et son équipe[33], l'utilisation du

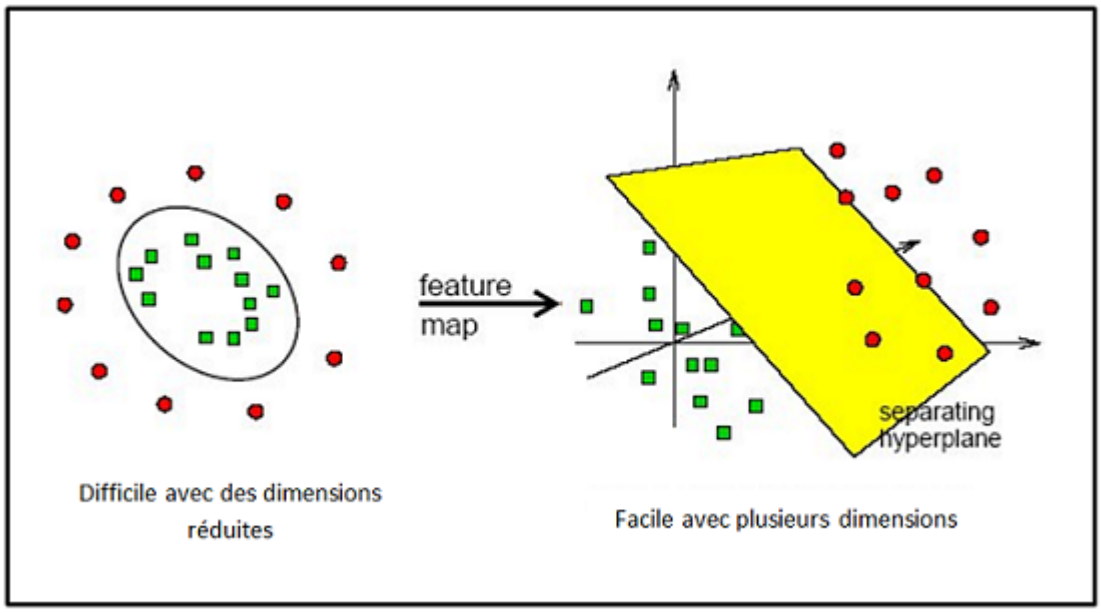

FIGURE 2.3 – Le principe du SVM

svm pour prendre les décisions et l'utilisation de l'algorithme à passage de jetons pour déterminer la suite de mots dans la reconnaissance de chiffres composés a donnée un résultat meilleur que celui d'un HMM classique avec un taux 96,96% pour les svm

et 96,47% pour les HMMs. Par contre ils démontrés que les performances des SVMs dépendent sur le nombre de support utilisé.

2.4 Dynamic Time Warpping (DTW)

Appelée aussi Alignement de Viterbi, introduite par H.Sakoe et S.Chiba[34], offre de meilleures performances car elle tient compte des compressions et extensions temporelles qui sont observées lors de la prononciation plus ou moins rapide d'un mot. Le principe de base est d'essayer de trouver le chemin optimal à parcourir parmi l'ensemble des distances entre les vecteurs. Au traitement de la parole un mot n'est jamais prononcé deux fois de la même manière, c'est pourquoi il est difficile de le repérer. La reconnaissance basée sur la DTW est plus fiable dans la reconnaissance de parole continue car elle tient compte des compressions et extensions temporelles. Le principe étant de créer une matrice de dimension $N \times J(k)$ (N et $J(k)$ sont respectivement le nombre de vecteurs dans la séquence de test et de référence) Une fois cette matrice obtenue, le but est de partir du point (1.1) et d'arriver au point final (N.J(k)) en minimisant le chemin à parcourir.

$$D(n,j) = d(n,j) + \min p(n,j) D(p(n,j))$$

Avec :

1. $p(n,j)$: ensemble des prédécesseurs possibles de l'élément (n,j)
2. $D(n,j)$: distance globale
3. $d(n,j)$: distance locale

La figure 2.4 résume le fonctionnement de la DTW

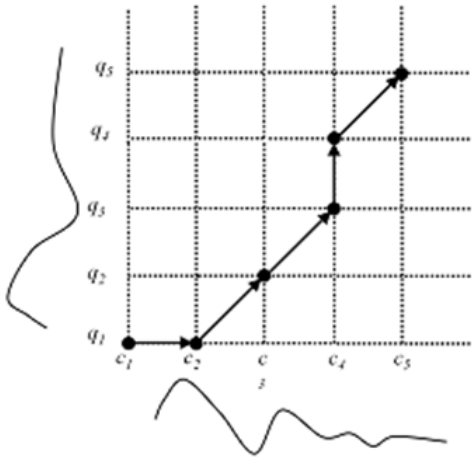

FIGURE 2.4 – Processus DTW

Où les c_i représentent les paramètres de la forme à reconnaitre, les coefficients MFCC par exemple , et les q_j représentent les paramètres d'une référence d'une forme connue. Après le calcul du taux de dissemblance de la donnée prononcée à reconnaitre par rapport à toutes les références, nous choisissons celle avec la plus grande valeur(fig. 2.5).

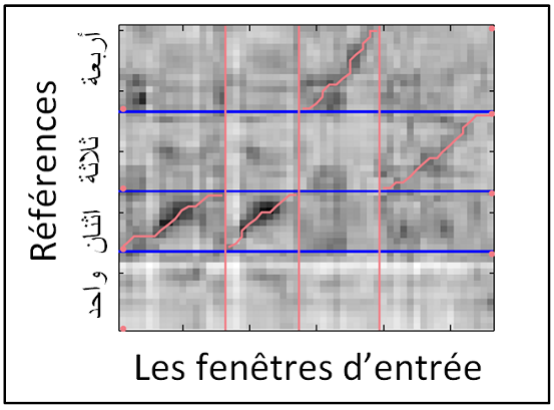

FIGURE 2.5 – Reconnaissance à base de la DTW

Il est claire que les HMMs sont plus adaptés au traitement de la parole mais ça ne laisse pas la DTW hors compétition car ils ont trouvés dans une recherche au traitement de la parole de la langue Punjabi [35] que la DTW est bien meilleure que les HMMs classiques avec un taux de reconnaissance à 92,3% par rapport à 87,5%.

2.5 Réseaux de neurones à délai temporel (TDNN)

Proposé par A.Waibel en 1989 pour la reconnaissance de la parole, il est constitué de sous réseau agissant comme des extracteurs de formes sur une période définie de la fenêtre d'entrée, chaque sous réseaux ayant pour tâche de reconnaitre des séquences. Le réseau se base sur la détection de groupe d'événements, dont la position absolue est moins importante que la disposition relative de leurs composantes. Les TDNN sont constitués comme les Perceptrons Multicouches d'une couche d'entrée, de couches cachées et d'une couche de sortie. Il se singularise d'un perceptron multicouche classique par le fait qu'il prend en compte une certaine notion de temps. C'est à dire qu'au lieu de prendre en compte tous les neurones de la couche d'entrée en même temps, il va effectuer un balayage temporel. La couche d'entrée du TDNN prend une fenêtre du spectre et balaie le signal ; cette fenêtre s'appelle fenêtre de spécialisation. Le TDNN permet ainsi de reconnaître le signal tout en étant moins strict que le PMC classique (c'est à dire qu'il pourra y avoir des petits décalages). Aussi, Les neurones de la couche $i+1$ sont reliés aux neurones de la couche i par des connexions à retard. Ce nombre de retard définit la largeur de la fenêtre de spécialisation. Le TDNN se caractérise par :

- Le nombre de couches (Chaque couche a deux directions : direction temporelle et direction caractéristique).
- Le nombre de neurones de chaque couche selon la direction temporelle, fenêtre d'observation.
- Le nombre de neurones de chaque couche selon la direction caractéristique.
- La taille de la fenêtre temporelle qui se traduit par le nombre de neurones de la couche i suivant la caractéristique temporelle vue par un neurone de la couche $i+1$.

– Le délai temporel (nombre de neurones) entre deux fenêtres successives dans une couche donnée.

La détermination du nombre de neurones de la couche $i+1$ selon la direction temporelle ($Nbt_i + 1$) se déduit du nombre de neurones de la couche i selon la direction temporelle (Nbt_i) et de la largeur de la fenêtre de spécialisation (D) de la manière suivante (fig. 2.6) : Les TDNNs introduisent des contraintes qui leurs permettent d'avoir

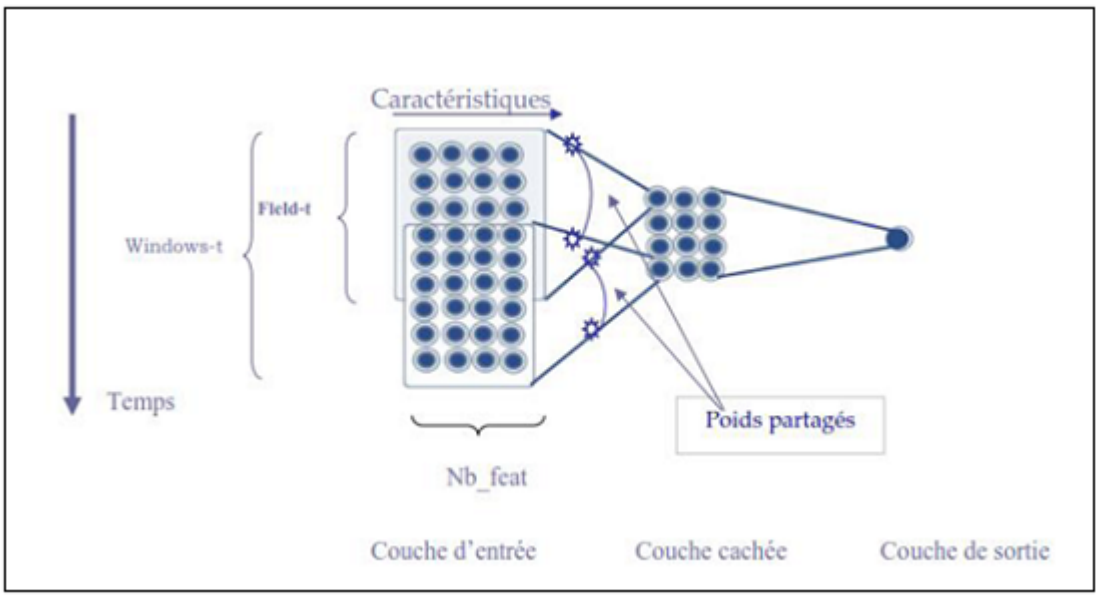

FIGURE 2.6 – Time Delay Neural Network (TDNN)

un certain degré d'invariance par décalage temporel et déformation. Celles-ci utilisent trois idées : poids partagés, fenêtre temporelle et délai.

– Les unités à délais : Les unités à retard sont des unités de base de ce modèle (TDNN) qui comportent des liaisons avec des retards, une sommation spatiotemporelle est donc effectuée au niveau de chaque neurone.

– Fenêtre de spécialisation : Le concept de fenêtre temporelle implique que chaque neurone de la couche $i+1$ n'est connecté qu'à un sous ensemble de la couche i. La longueur de cette fenêtre est la même entre deux couches données selon la caractéristique temporelle. Cette fenêtre temporelle permet que chaque neurone n'ait qu'une vision locale du signal, cette zone de vision s'appelle champs récepteurs du neurone ; ce dernier peut être vu comme une unité de détection d'une

caractéristique locale du signal.

– Les poids partagés : Les poids partagés permettent de réduire le nombre de paramètres du réseau neuronal et induisent ainsi une capacité de généralisation plus importante. Les poids sont partagés suivant la direction temporelle, c'est à dire que pour une caractéristique donnée, la fenêtre associée à celle-ci aura les mêmes poids selon la direction temporelle ceci est appelé l"invariance en translation.[36]

Pour résoudre les problèmes de prédiction et classification phonétique liés au réseaux de neurones à délai temporel, nous pouvons utilisé les algorithmes génétique[37]

2.6 Conclusion

Les algorithmes de classification ont généralement donnés des résultats convaincants mais quelques critiques liées aux SVM, DTW et le TDNN nous ont permis de travailler avec les HMMs. Tout d'abord, l' inconvénient des SVMs est le choix empirique de la fonction noyau adaptée au problème, et la DTW ne fait pas l'apprentissage et n'est pas basée sur une base mathématique solide, et enfin pour les TDNNs, ils nécessitent un long temps d'apprentissage avec une architecture difficile à déterminer. Par contre pour les HMMs, ce sont les plus performants pour le traitement de la parole car ils prennent en considération l'alignement temporel et l'ordre des séquences des données, et grâce à leur architecture nous pouvons introduire les propriétés linguistiques de la langue étudiée.

Chapitre 3

Expériences sur les mots connectés et continus

3.1 Introduction

Le traitement de la parole offre deux possibilités d'utiliser les mots d'un vocabulaire. La première ne dépend pas de la langue et prend la forme de chaque mot tel qu'il est ;on dit que ce sont des mots connectés. La deuxième utilise les caractéristiques linguistiques et découpe chaque mot en syllabes ou en phonèmes ; on dit que ce sont des mots continus. Dans ce chapitre nous allons faire une comparaison entre ces deux modes de traitement dans domaine de la reconnaissance de parole isolée et de parole continue sur des bases d'apprentissage de tailles variables pour tester l'influence de ses dernières sur le taux de réussite de chacun. L'approche utilisée comme technique de classification est les modèles de Markov Cachés pour lesquels nous allons utiliser l'outil HTK (annexe A).

3.2 Construction de la base de données

Tout travail s'appuyant sur l'apprentissage nécessite une base de données pour en apprendre le système et ensuite de l'évaluer. Ils existent plusieurs base de données internationales dans domaine de la parole tels que TIMIT qui a été développée par la commission DARPA pour l'anglais américain. Et nous trouvons aussi d'autres base

de données de différentes langues connus, comme le français et l'allemand, et inconnus, comme le vietnamiens et le turque. Pour la langue arabe, nous n'avons pas découvert une base de données standard, mais nous avons quand même repéré quelques références. La base KACST développée par l'institut du roi Abdul-Aziz en Arabie Saoudite, construite à base d'instruments médicaux [38]. Et la base ALGERIAN ARABIC SPEECH DATABASE (ALGASD)[39] développée en Algérie pour le traitement de la parole arabe en prenant en compte les différents accents de différentes régions du pays. La non disponibilité et le manque de moyens pour avoir une base de données audio nous a poussé à construire notre propre base de données destinée à faire la reconnaissance des chiffres et les opérations d'une calculatrice standard en arabe pour un seul utilisateur. Nous avons fait 27 enregistrements de 28 mots de vocabulaire.

3.3 Introduction des fichiers sons

Nous avons pu utiliser n'importe quel outil d'acquisition des fichiers audio mais nous avons choisi un outil qui est développé pour le traitement de la parole. Cet outil s'appelle *Praat* téléchargeable librement à partir du site `http://www.fon.hum.uva.nl/praat/download_win.html`.*Praat* fait en plus de l'acquisition des données audio, des analyses du pitch, l'analyse spectrale du signal et d'autres fonctionnalités dont la plus intéressante et la reconnaissance vocale basée sur l'analyse phonétique et syntaxique (fig. 3.1). Mais cette dernière n'est disponible que pour quelques langues et, malheureusement, l'arabe n'en fait pas partie.

3.4 Etiquetage manuel des données

Notre système fait un apprentissage supervisé pour lequel les données doivent être représentées par leurs caractéristiques et leurs classes associées. Ces données sont de nature audio et leurs caractéristiques sont les coefficients MFCC avec leurs dérivés primaires et secondaires. Et la classe de sortie contient des étiquettes. L'outil HTK met à disposition une fonction intitulée HSLAB qui permet de visualiser un fichier

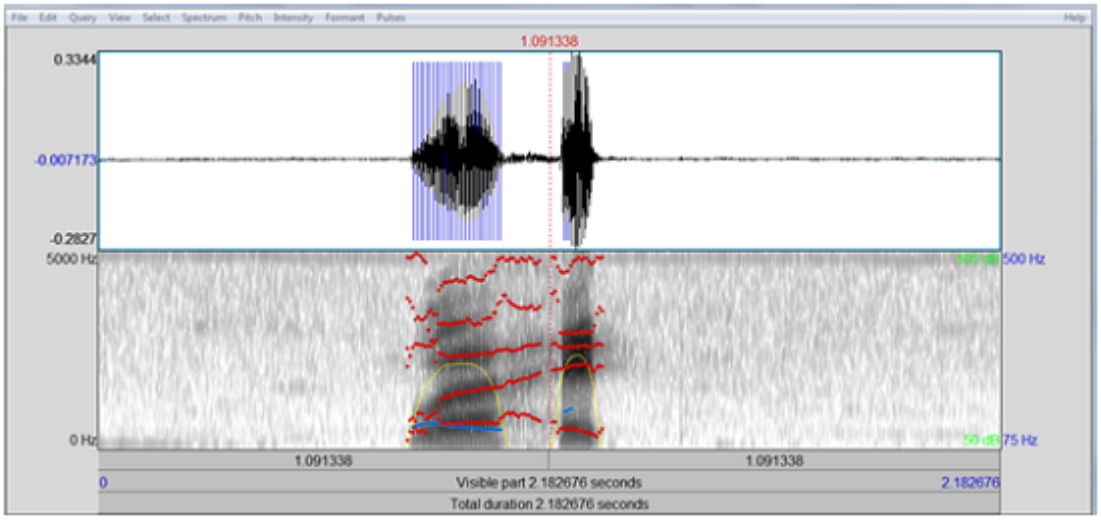

FIGURE 3.1 – Quelques fonctionnalités de Praat

audio dans une interface graphique pour ensuite étiqueter les zones significatives en sélectionnant leurs parties associés. Il y a deux manières d'étiquetage de la parole :

3.4.1 Etiquetage pour la reconnaissance de mots connectés

C'est la méthode pour laquelle chaque mot est représenté par sa forme lexicale sans prendre en compte la phonation (fig. 3.2)

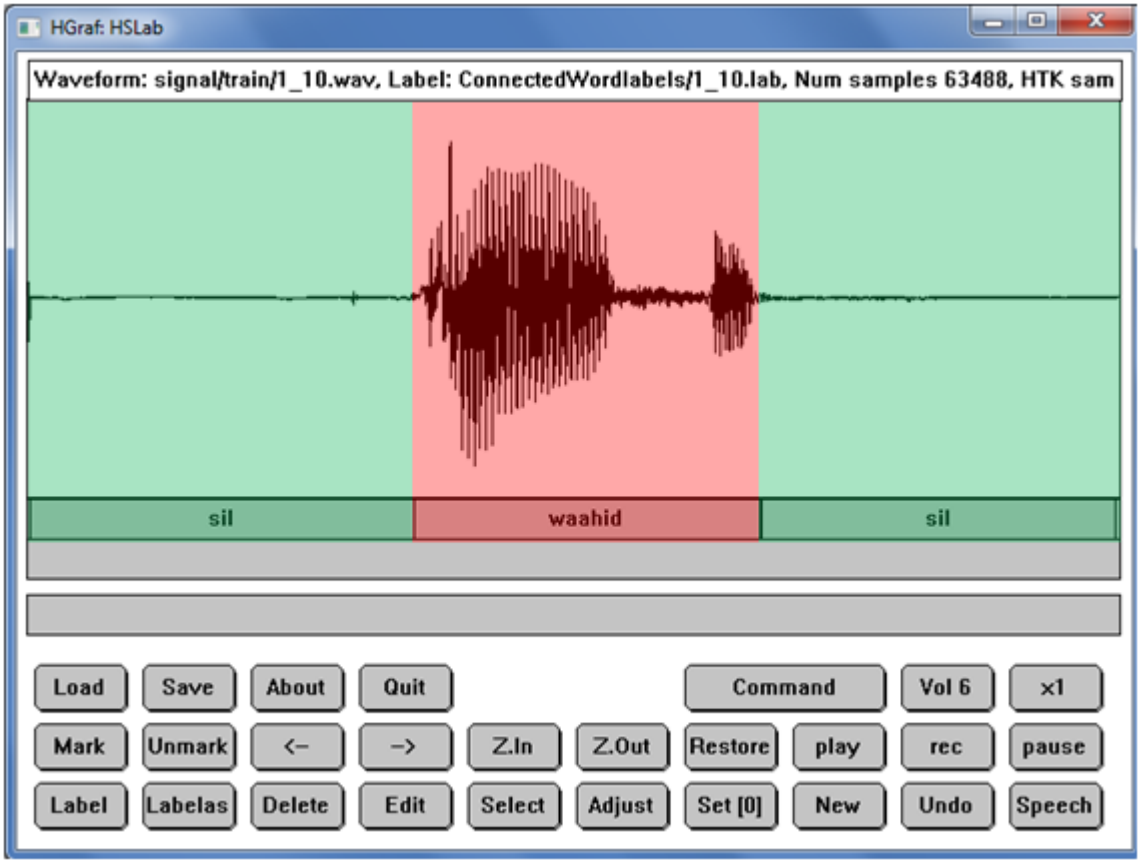

FIGURE 3.2 – Étiquetage de mots connectés

3.4.2 Étiquetage pour la reconnaissance de mots continue

Pour ce type d'étiquetage chaque mot est découpé en syllabes ou en phonèmes, et les caractéristiques linguistique de ce mot sont introduits par la suite (fig. 3.3).

Le tableau (tab. 3.1) représente le vocabulaire et les étiquettes selon les deux formes d'étiquetage

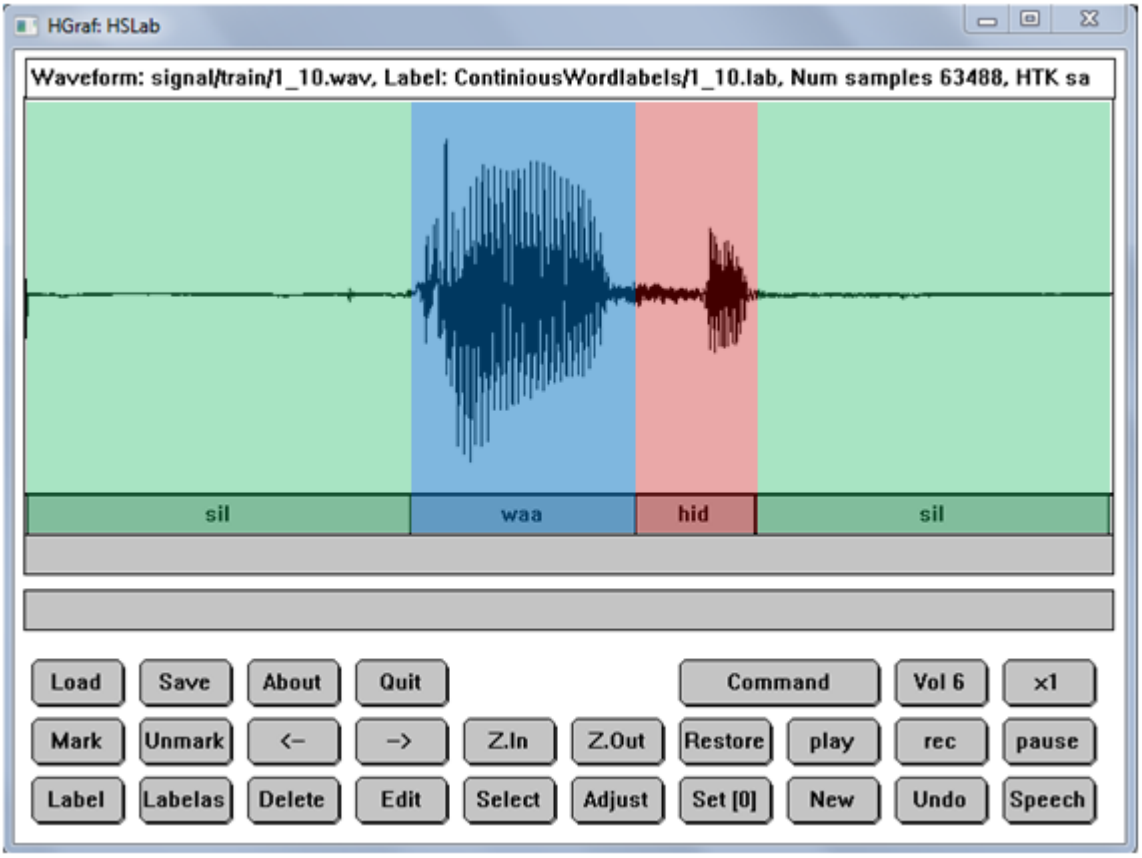

FIGURE 3.3 – Étiquetage de mots continus

Le mot	**Prononciation en Arabe**	**Étiquetage connectés**	**Étiquetage continus**
0	صِفر	cifr	cifr
1	وَاحِد	waahid	waa hid
2	اِثنَان	ithnaan	ith naan
3	ثَلَاثَة	thalaatha	tha laa tha

Le mot	**Prononciation en Arabe**	**Étiquetage connectés**	**Étiquetage continus**
4	أَربَعَة	arbaaa	ar ba aa
5	خَمسَة	khamsa	kham sa
6	سِتَة	sitsa	si tsa
7	سَبعَة	sabaa	sa b aa
8	ثَمَانيَة	thamaania	tha maa nia
9	تِسعَة	tisaa	tis aa
10	عَشَرة	aachar	aa char
11	أَحَدَاعَشَر	a7adaaaachar	a 7a daa aa char
12	اِثنَاعَشَر	ithnaaaachar	ith naa aa char
20	عِشرُون	iichroun	iich roun
30	ثَلَاثُون	thalaathoun	tha laa thoun
40	أَربَعُون	arba3oun	ar ba 3oun
50	خَمسُون	khamsoun	kham soun
60	سِتُّون	sitsoun	si tsoun
70	سَبعُون	sab3oun	sa b 3oun
80	ثَمَانُون	thamaavoun	tha maa noun
90	تِسعُون	tis3oun	tis 3oun
+	زَائِد	zaaid	zaa id
-	نَاقِص	naakis	naa kis
×	ضَرب	dharb	dharb
/	قِسمَة	kismats	kis mats
=	يُسَاوِي	yosaawii	yo saa wii
<-	رُجُوع	rojou3	ro jou3
wa	وَ	wa	wa

TABLE 3.1 – Étiquetage connecté et continu des mots de vocabulaire

3.5 Paramètrisation

Selon certaines recherches[40], la méthode MFCC est la meilleure pour la reconnaissance de la parole, et les dérivés primaires et secondaires fournissent des informations supplémentaires. Ces paramètres sont calculables par le biais d'une fonction dont dispose l'outil HTK. Cette fonction est HCOPY qui prend en entrée un fichier audio et calcule ses coefficients suivant une configuration de la taille des fenêtres, nombre de ceptres, le type de fenêtrage, et d'autres paramètres introduits par l'utilisateur. Certains travaux utilisent même des algorithme de Boosting comme AdaBoost pour pallier les carences des données d'apprentissage[41].

Dans notre cas nous avons calculé les paramètres des 27×28 fichiers audio. Ces paramètres sont : Le nombre de coefficients MFCC utilisé est 8 + l'energie + les dérivés (donc 18) choisi a partir des travaux similaires sur la parole arabe[42]

3.6 Définition du HMM

La fonction de principe de HTK est de manœuvrer des ensembles de modèles de Markov cachés (HMMs). La définition d'un HMM doit spécifier la topologie du modèle, les paramètres de transition et les paramètres de distribution de rendement. Les vecteurs d'observation du HMM peuvent être divisés en multiples trames de données indépendantes et chaque trame peut avoir son propre poids.[43] Pour l'outil HTK, les chaines de Markov cachées sont d'abord estimées par des prototypes (fig. 3.4). La fonction d'une définition de prototype est de décrire la forme et la topologie du HMM, les nombres réels utilisés dans la définition ne sont pas importants. Par conséquent, la taille du vecteur (VecSize) et le type de paramètre (MFCC) devraient être spécifiés et le nombre d'états doit être choisi (NumStates). Les transitions permises entre les états devraient être indiquées en mettant des valeurs différentes de zéro dans les éléments correspondants à la matrice de transition (TransP) et zéros ailleurs. La somme de chaque ligne de la matrice de transition doit être égale à 1, sauf la dernière qui devrait être 0. Toutes les valeurs moyennes peuvent être zéro mais les variances diagonales devraient être positifs et les matrices de covariance devraient avoir les éléments diagonaux posi-

tifs. Toutes les définitions d'état peuvent être identiques. Rappelons que notre but est

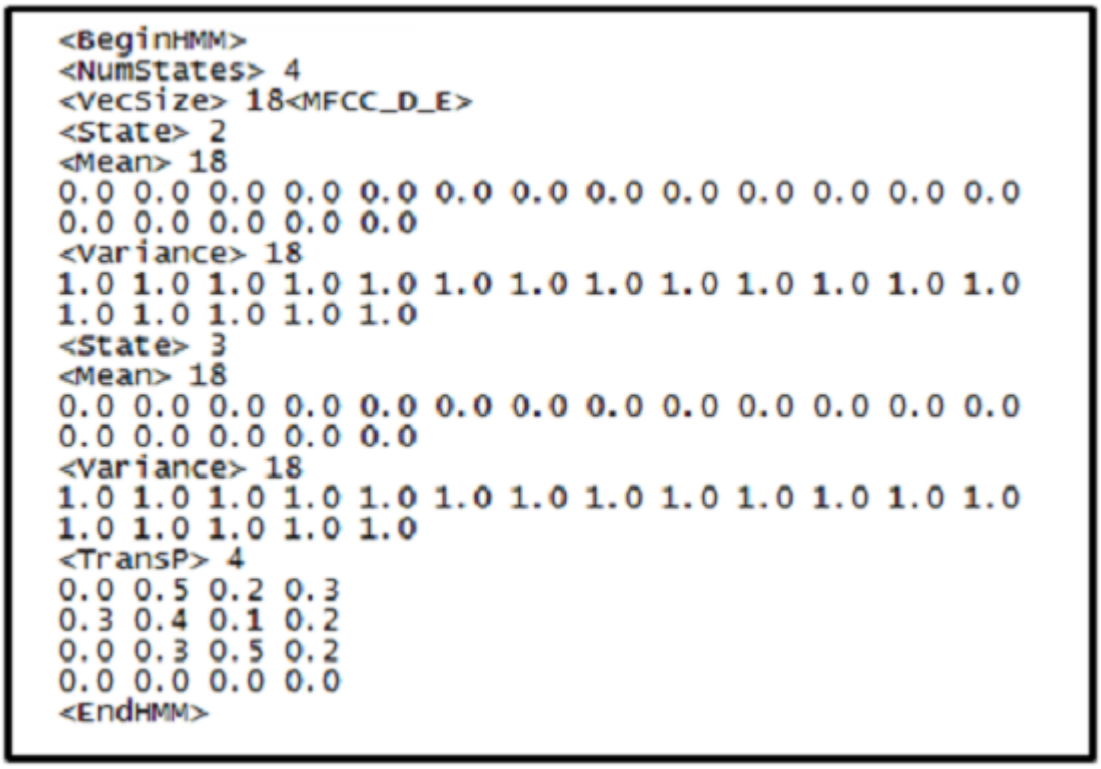

FIGURE 3.4 – Prototype d'un HMM

de faire une comparaison entre la reconnaissance de mots connectés et les mots continus dont chacun nécessite une modélisation de son HMM

3.6.1 HMM de reconnaissance de mots connectés

Nous avons modélisés un mot connecté par le nombre de syllabes qu'il contient, c'est-à-dire que chaque syllabe représente un état du HMM associé au mot en plus les deux états d'entrée et de sortie. La figure (fig. 3.5)représente un prototype du mot « sabaa » (sept).

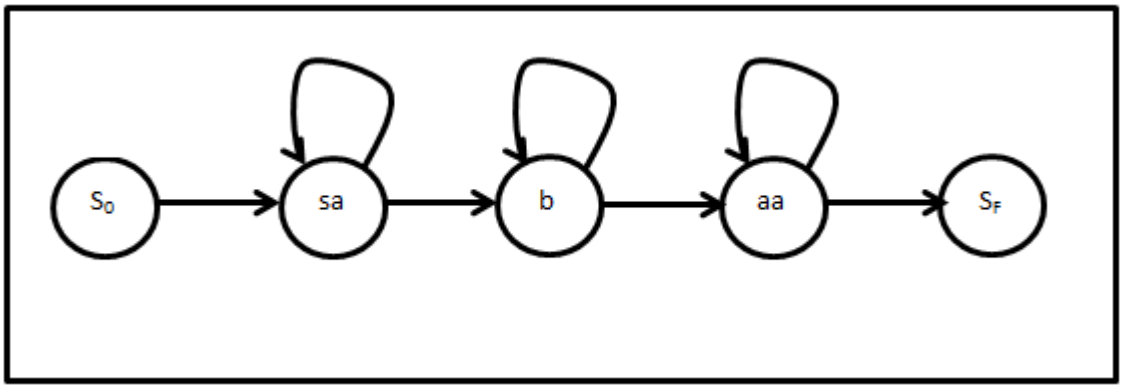

FIGURE 3.5 – Prototype d'un HMM de mot connecté

3.6.2 HMM de reconnaissance de mots continus

Pour les mots continus, chaque syllabe est spécifiée par un HMM. Notre système contient 42 syllabes et pour leur modélisés nous nous somme basé sur les classes de syllabes de la langue arabe vues au premier chapitre. Nous avons mentionnés que la langue arabe comporte 3 types de syllabes qui sont : CV, CVC et CVCC. Pour cette raison nous ne définissons que 3 types de HMM pour la reconnaissance de mots continus (fig. 3.6). Le tableau (fig. 3.2) représente les HMM utilisés des 42 syllabes.

Syllabe	**Représentation syllabique**	**Numéro du HMM**	**Syllabe**	**Représentation syllabique**	**Numéro du HMM**
cifr	CVCC	3	iich	CV	1
waa	CV	1	roun	CVC	2
hid	CVC	2	thoun	CVC	2
ith	CV	1	3oun	CVC	2
naan	CVC	2	soun	CVC	2
tha	CV	1	tsoun	CVC	2
laa	CV	1	noun	CVC	2
ar	CV	1	zaa	CV	1
ba	CV	1	id	CV	1
aa	CV	1	naa	CV	1
kham	CVC	2	kis	CVC	2
sa	CV	1	dharb	CVCC	3
b	CV	1	mats	CVCC	3
si	CV	1	yo	CV	1
tsa	CV	1	saa	CV	1
maa	CV	1	wii	CV	1
nia	CV	1	ro	CV	1
tis	CVC	2	jou3	CVC	2
char	CVC	2	wa	CV	1
a	CV	1	daa	CV	1
7a	CV	1			

TABLE 3.2 – Les HMMs des syllabes du vocabulaire

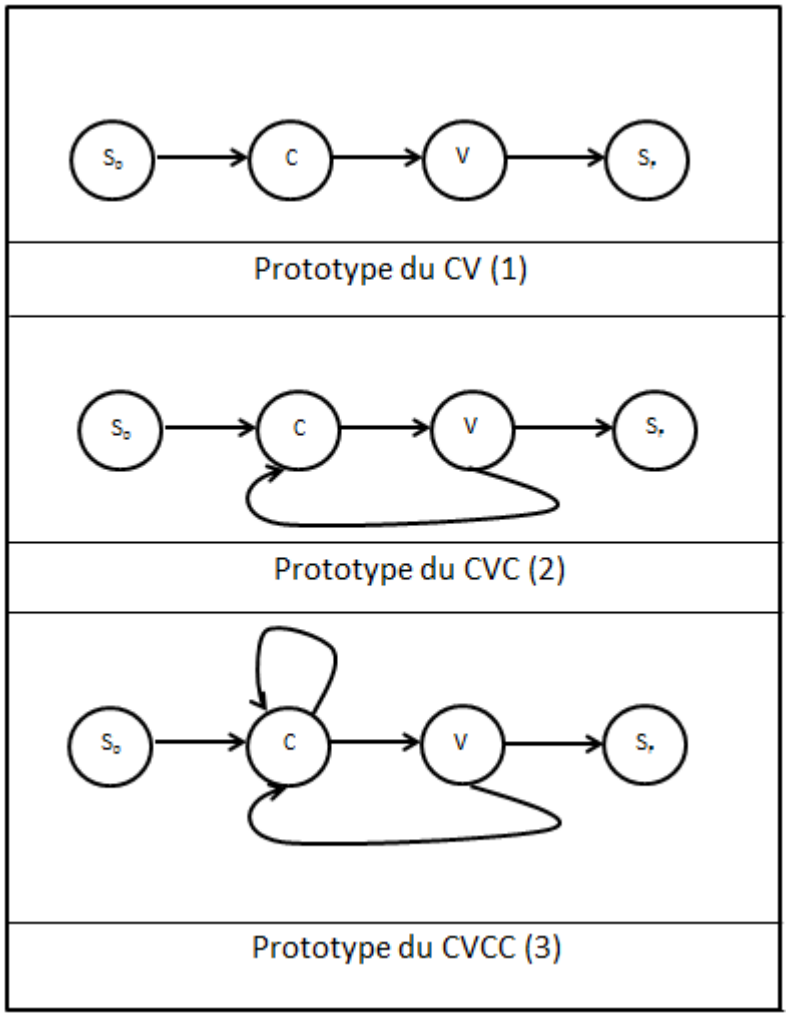

FIGURE 3.6 – Prototypes des mots continus

Dans le HMM associé au silence on ajoute un lien du $2^{\text{ème}}$ état au $4^{\text{ème}}$ état et un autre du $4^{\text{ème}}$ au $2^{\text{ème}}$ pour rendre le modèle plus robuste en absorbant les variations des impulsions nasales de l'ensemble d'apprentissage [43]

3.7 Initialisation

Avant de démarrer le processus d'apprentissage, les paramètres des HMMs doivent être correctement initialisés en utilisant la base d'apprentissage afin de permettre une convergence rapide et précise de l'algorithme d'apprentissage.[44] La commande HInit

de l'outil HTK permet d'initialiser les HMMs par alignement temporel en utilisant l'algorithme de Viterbi à partir des prototypes, et les données d'apprentissage dans leur forme MFCC et leur fichier étiqueté associé. L'organigramme suivant résume le processus (fig. 3.7) : Premièrement, HTK charge le prototype du HMM à définir, ensuite il

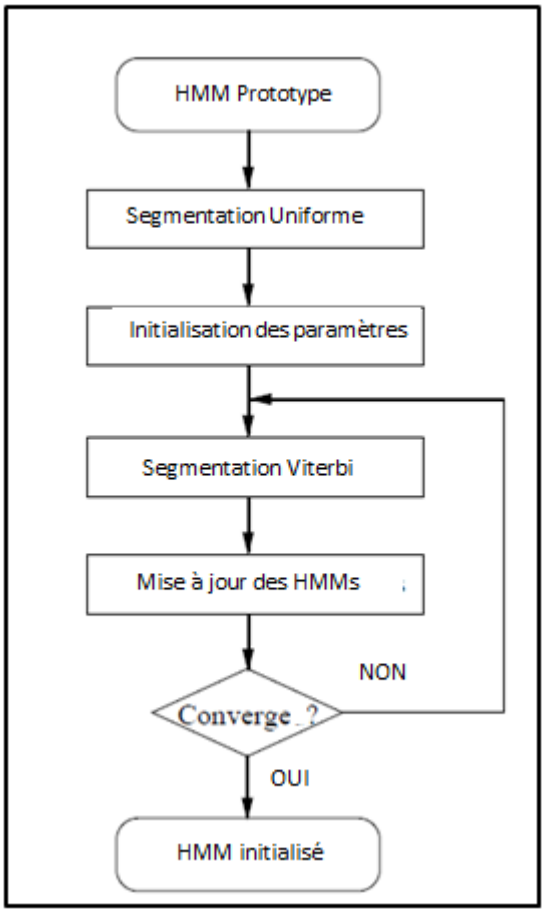

FIGURE 3.7 – L'opération HInit

cherche dans la base des étiquettes le label portant le nom de ce HMM ; à noter qu'un fichier label contient le temps de début et de fin d'une étiquette dans un enregistrement. Et par le biais du fichier de configuration il trouve le lien avec les coefficients MFCC calculés précédemment et en prend ensuite ce dont il a besoin (fig. 3.8). Quand le système charge tout ce dont 'il a besoin l'algorithme de Viterbi est employé pour trouver l'ordre le plus susceptible d'état correspondant à chaque exemple d'apprentissage, puis les paramètres de HMM sont estimés. Nous pouvons calculer le logarithme

de vraisemblance de l'ensemble d'apprentissage pour éviter l'effet de bord de trouver l'alignement Viterbi des états. Par conséquent, le procédé entier d'évaluation peut être répété jusqu'à ce qu'aucun accroissement plus ultérieur de probabilité ne soit obtenu.

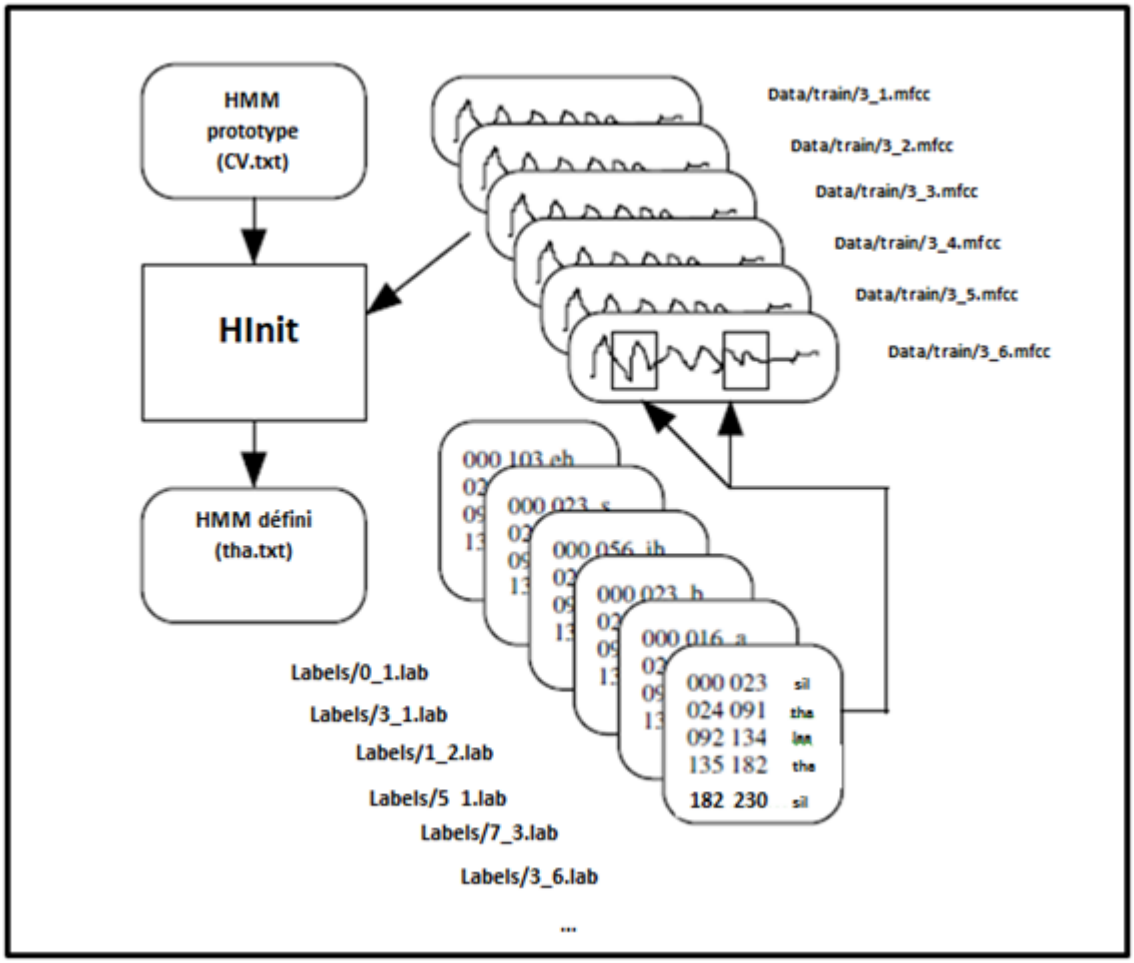

FIGURE 3.8 – Processus de chargement de données pour la commande HInit

3.8 Apprentissage

Nous avons vu que l'initialisation n'est qu'un calcul de distance ; car l'algorithme de Viterbi se base essentiellement sur la DTW. Et pour l'apprentissage nous allons appliquer l'algorithme de Baum-Welch vu en deuxième chapitre. Cette étape est assurée par la commande HRest de l'outil HTK, qui est désigné à la manipulation des HMMs

isolés. Son fonctionnement est très semblable à HInit sauf que, suivant les indications de la figue(fig. 3.9), en partant d'un HMM initialisé elle emploie la réévaluation Baum-Welch au lieu de l'apprentissage de Viterbi. Ceci implique de trouver que la probabilité d'être dans un état donné en une fenêtre de temps donnée en utilisant l'algorithme Baum-Welch (forward-backward). Cette probabilité est alors employée pour former des moyennes pondérées pour les paramètres du HMM.

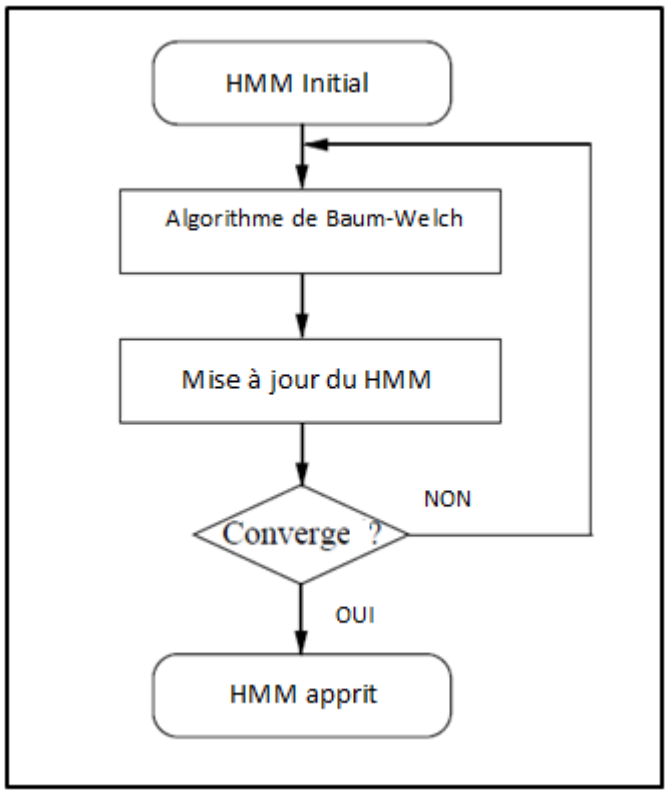

FIGURE 3.9 – Le processus d'apprentissage

3.9 Définition de la grammaire

Il est nécessaire de donner au système des indications pour qu'il puisse déterminer une solution satisfiable. A noter que la grammaire ne dépend pas du type d'étiquetage des mots (connectés ou continus). Nous avons construit nos grammaire en suivant le format du HTK. Où les variables sont précédées par un $, les {} aux extrémités des

mots dénotent zéro ou plusieurs répétitions permises, les [] pour au plus une répétition, le trait verticale signifie une alternative, et la parenthèse ouverte représente le début de l'arbre de dérivation et la parenthèse fermante pour l'état finale.[44]

3.9.1 Grammaire pour la reconnaissance de parole isolée

(fig. 3.10) Ici tous les mots du langage ont la même probabilité.

```
(WORD-START {sil}

[cifr|waahid|ithnaan|thalaatha|arbaaa|khamsa|sitsa|sabaa|thamaania|tisaa|aachar|a7ad
aaaachar|ithnaaaachar|iichroun|thalaathoun|arba3oun|khamsoun|sitsoun|sab3oun|
thamaanoun|tis3oun|zaaid|naakis|dharb|kismats|yosawi|rojou3|wa]

{sil}

WORD-END)
```

FIGURE 3.10 – grammaire de parole isolée

3.9.2 Grammaire pour la reconnaissance de parole continue

Il est clair que plus la grammaire est complexe, plus le système à plus d'alternatives de reconnaissance et par la suite le taux d'erreur augmente. Nous avons choisi de travailler avec une grammaire simple qui permet de générer les mots de type « A op B » (fig. 3.11) avec A et B deux opérandes et op une opération. La liste suivante donne plus de détail.

```
(SENT-START {sil}
[cifr|waahid|ithnaan|thalaatha|arbaaa|khamsa|sitsa|sabaa|thamaania|tisaa|aachar
|a7adaaaachar|ithnaaaachar|iichroun|thalaathoun|arba3oun|khamsoun|sitsoun|sab3oun|
thamaanoun|tis3oun|zaaid|naakis|dharb|kismats|yosawi|rojou3|wa]
        {sil}
SENT-END)
```

FIGURE 3.11 – grammaire de parole continue

A noter que cette grammaire, au contraire de la grammaire des mots isolés, débute avec le mot « SENT-START » et se termine par « SENT-END ».

3.10 Construction du dictionnaire

Le système doit naturellement savoir quel HMM correspond chacune des variables de grammaire « cifr, waahid, . . . , rojou3, wa ». Cette information est stockée dans un fichier texte appelé le dictionnaire de tâche. Dans une tâche si simple, la correspondance est franche, et le dictionnaire de tâche joint simplement les mots ou les syllabes(Tab. 3.3).

Dictionnaire des mots continus	**Dictionnaire des mots connectés**
SENT-START []	SENT-START []
SENT-END []	SENT-END []
cifr [0] cifr	cifr [0] cifr
waahid [1] waa hid	waahid [1] waahid
ithnaan [2] ith naan	ithnaan [2] ithnaan
thalaatha [3] tha laa tha	thalaatha [3] thalaatha
arbaaa [4] ar ba aa	arbaaa [4] arbaaa
khamsa [5] kham sa	khamsa [5] khamsa
sitsa [6] si tsa	sitsa [6] sitsa

Dictionnaire des mots continus	**Dictionnaire des mots connectés**
sabaa [7] sa b aa	sabaa [7] sabaa
thamaania [8] tha maa nia	thamaania [8] thamaania
tisaa [9] tis aa	tisaa [9] tisaa
aachar [10] aa char	aachar [10] aachar
a7adaaaachar [11] a 7a daa aa char	a7adaaaachar [11] a7adaaaachar
ithnaaaachar [12] ith naa aa char	ithnaaaachar [12] ithnaaaachar
iichroun [20] iich roun	iichroun [20] iichroun
thalaathoun [30] tha laa thoun	thalaathoun [30] thalaathoun
arba3oun [40] ar ba 3oun	arba3oun [40] arba3oun
khamsoun [50] kham soun	khamsoun [50] khamsoun
sitsoun [60] si tsoun	sitsoun [60] sitsoun
sab3oun [70] sa b 3oun	sab3oun [70] sab3oun
thamaanoun [80] tha maa noun	thamaanoun [80] thamaanoun
tis3oun [90] tis 3oun	tis3oun [90] tis3oun
zaaid [+] zaa id	zaaid [+] zaaid
naakis [-] naa kis	naakis [-] naakis
dharb [x] dharb	dharb [x] dharb
kismats [/] kis mats	kismats [/] kismats
yosawi [=] yo saa wii	yosawi [=] yosaawii
rojou3 [<-] ro jou3	rojou3 [<-] rojou3
wa wa	wa wa
sil sil	sil sil

TABLE 3.3 – Dictionnaires du système

Les éléments de gauches se rapportent aux noms des variables de grammaire. Les éléments de droite se rapportent aux noms du HMM (présenté par le h dans la définition du HMM). Les éléments encadrés au milieu sont facultatifs, ils indiquent les symboles qui seront affichés par le système de reconnaissance : les noms des étiquettes sont employés ici (par défaut, les noms des variables de la grammaire sont affichés.)

3.11 Génération du réseau de mots (Word Network)

À ce stade, notre tâche de reconnaissance de la parole, complètement définie par son réseau, son dictionnaire, et son ensemble de HMMs, est opérationnelle. La figure (fig. 3.12) est le réseau complet utilisé par le système. Chaque cercle représente le HMM de l'étiquette qu'il contient.

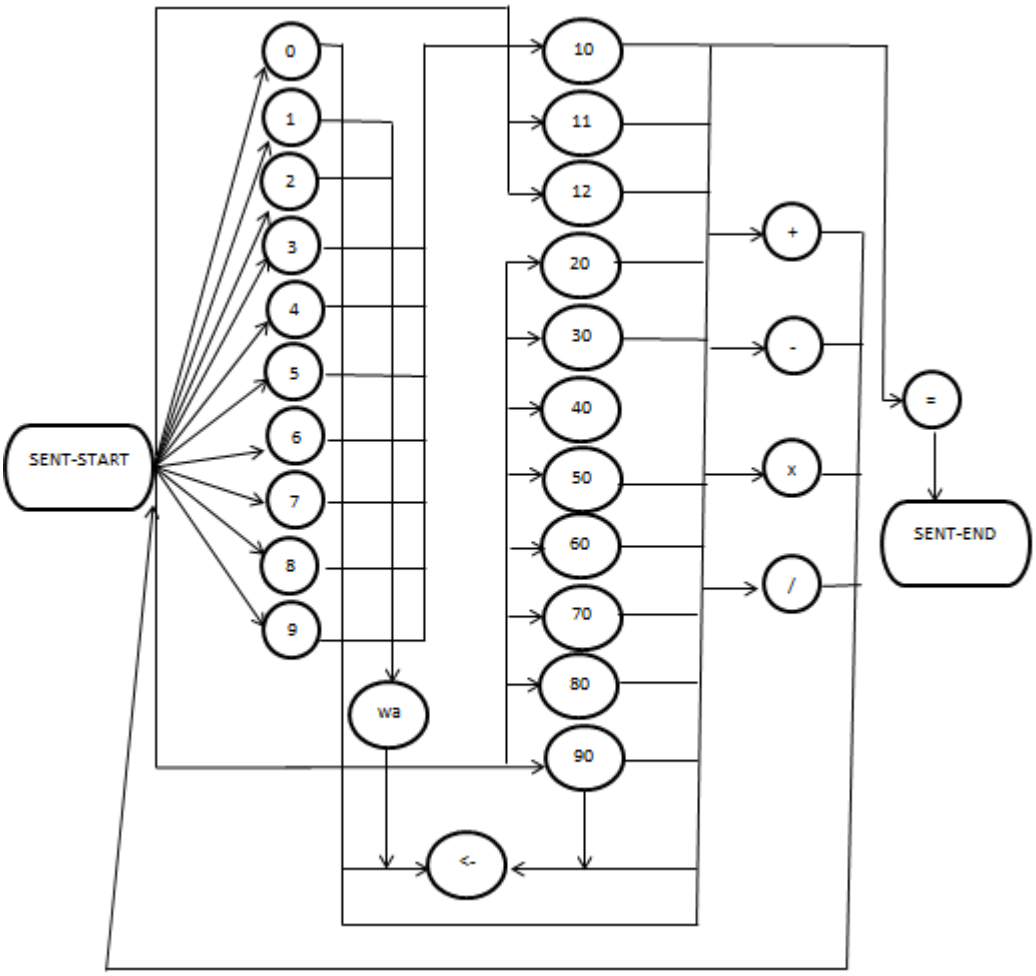

FIGURE 3.12 – le réseau de mots associé à la grammaire de la parole continue

3.12 La reconnaissance

Après avoir entré le fichier de la parole à reconnaitre via le microphone, il est transformé en un fichier de caractéristiques (MFCC pour notre cas) par la commande HCopy de la même manière que ce qui a été fait avec les données d'apprentissage (étape acoustique d'analyse). Pour une expression donnée avec T fenêtres possibles, chaque nœud du chemin de début à la fin du réseau qui traverse exactement T émettant des états du HMM est une hypothèse potentielle d'identification. Chacun de ces chemins a une probabilité logarithmique qui est calculée en additionnant la probabilité de notation de chaque transition individuelle dans le chemin et la probabilité logarithmique de chaque état d'émission produisant l'observation correspondante. Dans un HMM, les transitions qui sont déterminées par les paramètres du HMM, et les transitions entre les modèles sont constantes et les transitions des extrémités des mots sont déterminées par les probabilités de vraisemblance avec le modèle de langage. A Chaque étape en appliquant l'algorithme à passage du jeton vu au chapitre précédent, les jetons sont propagés le long des transitions permises et s'arrêtent lors d'un état d'émission du HMM. Quand il y a les sorties multiples d'un nœud, le jeton est copiée de sorte que tous les chemins possibles soient explorés en parallèle. Pendant que le jeton passe à travers des transitions et par des nœuds, sa probabilité logarithmique est incrémentée par les probabilités correspondantes de transition et d'émission. Lorsque chaque jeton traverse le réseau il doit maintenir un historique enregistrant son itinéraire. La quantité de détail dans cet historique dépend du rendement voulu d'identification défini par la grammaire [43]. Ce travail est assuré par la commande HVite de l'outil HTK. Cette commande permet à partir d'un fichier de paramètres de produire un fichier contenant les étiquettes affectées par le système aux différentes parties du fichier audio, en plus d'un affichage sur la fenêtre console.

3.13 L'évaluation

Nous avons optés pour une comparaison entre la reconnaissance de mots connectés et la reconnaissance de mots continus dans les domaines de la parole isolée et la parole

continue avec une taille de corpus d'apprentissage variante, et mono-locuteur. Nous avons fait l'évaluation pour un corpus de 5, 10, 15 et 20 enregistrements et pour le corpus de test nous avons utilisé 7 enregistrements, pour tous les tests, pour la parole isolée et pour la parole continue nous avons choisi des combinaisons de mots de type A opération B avec A et B des opérandes, qui ont eu le plus grand taux de reconnaissance à la parole isolée. Le tableau suivant donne les résultats que nous avons obtenus sur ces corpus pour la parole isolée.

	Mots connectés				**Mots continus**			
	base de 5	**base de 10**	**base de 15**	**base de 20**	**base de 5**	**base de 10**	**base de 15**	**base de 20**
0	0	0	0	0	3	4	2	2
1	6	6	6	6	7	7	7	7
2	4	5	7	7	4	7	7	7
3	0	0	0	0	5	6	6	6
4	2	1	2	0	6	7	6	7
5	0	7	7	7	2	2	0	0
6	4	7	7	7	2	0	0	0
7	1	1	0	0	5	5	5	5
8	5	5	5	7	7	7	7	7
9	7	7	7	7	7	7	7	7
10	7	7	7	7	6	5	5	6
11	6	6	7	7	7	7	7	7
12	1	4	4	4	7	7	7	7
20	4	6	6	7	6	7	7	7
30	0	1	1	1	3	3	3	4
40	3	4	1	2	7	7	6	7
50	7	3	4	5	0	0	0	0

	Mots connectés				**Mots continus**			
	base de 5	**base de 10**	**base de 15**	**base de 20**	**base de 5**	**base de 10**	**base de 15**	**base de 20**
60	4	6	6	6	4	1	0	2
70	3	0	0	1	6	5	6	7
80	4	1	2	3	6	5	6	7
90	6	7	7	7	7	7	7	7
+	6	7	7	7	6	4	5	4
-	7	7	7	7	6	6	7	7
×	0	0	0	0	0	0	0	0
/	7	7	7	7	7	6	7	7
=	1	1	1	3	7	7	7	7
rojou3	7	5	6	7	6	7	7	7
wa	6	6	6	6	6	6	6	7

TABLE 3.4 – Résultats avec différents corpus de la parole isolée

Au-dessous chaque tableau donne les résultats pour la parole continue de chaque base de chaque corpus

La base	**Mots connectés à parole continue**	**Mots continus à parole continue**
1+9	0.33	0.67
28/91	1	0.85
10+18	0.75	0.5
99-48	0.85	1
22+11	1	1
98-12	0.8	1
14-49	0.67	0.5
Taux	0.771428571	0.788571429

TABLE 3.5 – Résultat du corpus de 5

La base	Mots connectés à parole continue	Mots continus à parole continue
1+2	0.33	0.33
22/91	0.85	0.85
10+18	0.75	0.5
99-42	1	0.85
22+11	0.8	0.8
92-12	0.8	0.8
15-29	0.83	0.83
Taux	0.765714286	0.708571429

TABLE 3.6 – Résultat du corpus de 10

La base	Mots connectés à parole continue	Mots continus à parole continue
1+2	0.66	0.33
22/91	0.85	0.85
10+18	0.75	0.5
99-82	0.85	0.85
28+11	0.2	0.8
92-12	0.8	0.8
18-29	0.67	0.83
Taux	0.682857143	0.708571429

TABLE 3.7 – Résultat du corpus de 15

La base	Mots connectés à parole continue	Mots continus à parole continue
1+2	0.33	0.33
22/91	0.85	0.85
10+18	0.75	0.5
99-82	0.85	0.85
28+11	0.8	0.8
92-12	0.8	0.8
18-29	0.67	0.83
Taux	0.72	0.708571429

TABLE 3.8 – Résultat du corpus de 20

3.14 Analyse des résultats

Il est clair que nous avons obtenus un taux de reconnaissance trop élevé avec les mots continus par rapport aux mots connectés. Ceci revient à dire que les mots sont traités avec plus de précision en prenant en compte leurs caractéristiques linguistiques. Il y a des mots avec un taux de reconnaissance trop petit ou nul comme pour le cas de ‘cifr’ et ‘dharb’, et ceci s’explique par le besoin de plus d’apprentissage.

La figure (fig. 3.13)montre des améliorations en fonction de la taille du corpus pour les mots connectés mais pour arriver à un taux concurrent aux mots continus nous avons besoin de plus de données d'apprentissage. Aussi, nous remarquons une dégradation aux mots continus et ceci est dû aux ambiguïtés phonatoires. Par exemple 'arbaaa' et 'sabaa' se terminent par la même syllabe 'aa'.

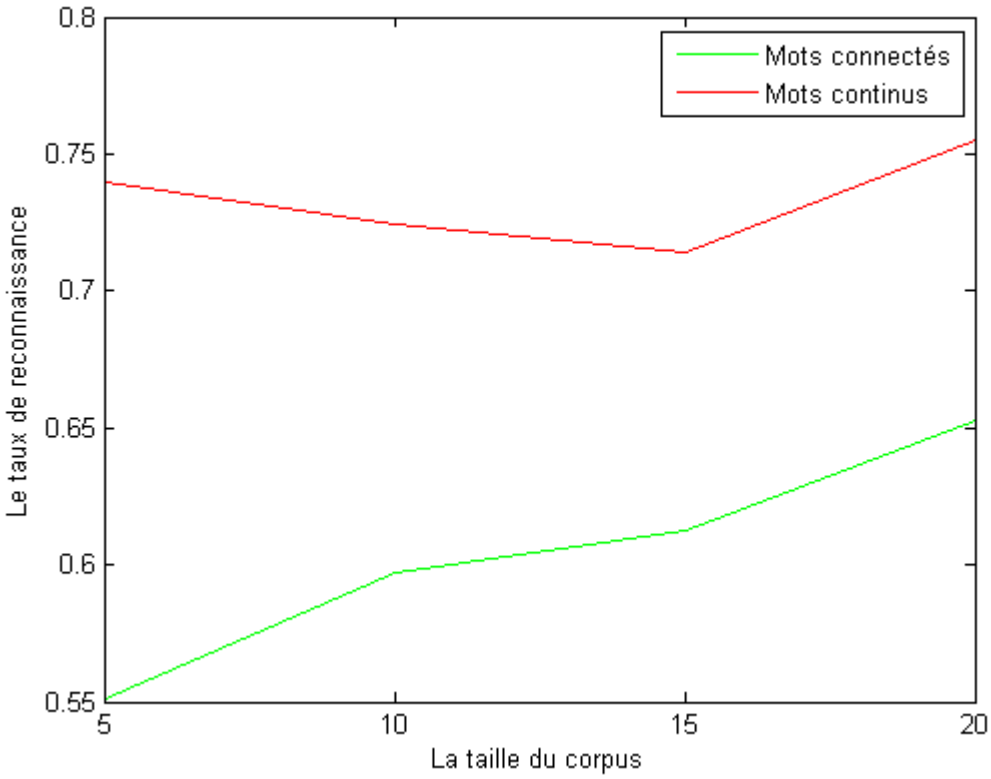

FIGURE 3.13 – Variation du taux de reconnaissance de parole isolée en fonction de la taille du corpus

La figure(fig. 3.14) montre des alternatives des meilleurs taux de reconnaissance entre les mots connectés et les mots continus. Et plus la base s'élargit le taux diminue pour le cas des mots connectés pour ensuite s'améliorer après le corpus 15. Ceci s'explique par le mauvais choix de la base de test c'est à dire que ce qui a donné de bon résultats en mode isolé ne donne pas forcement de meilleurs résultats au mode continu.

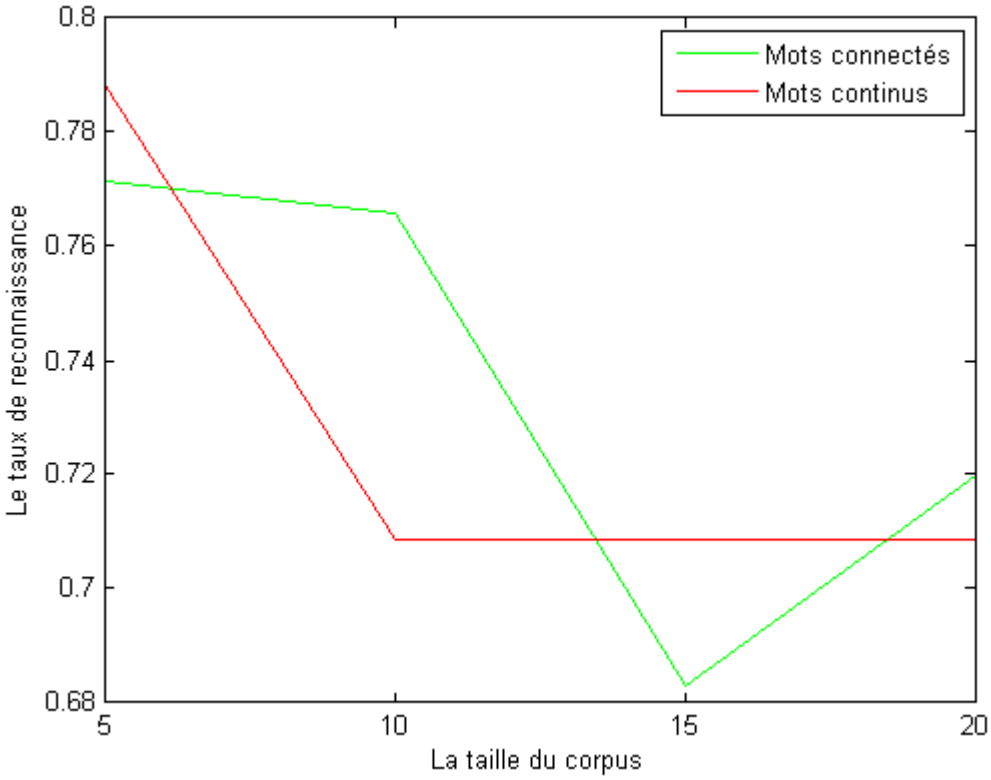

FIGURE 3.14 – Variation du taux de reconnaissance de parole continue en fonction de la taille du corpus

3.15 Implémentation d'une calculatrice vocale

Nous avons réalisés une calculatrice vocale du vocabulaire précédant en prenant les meilleurs modèles de Markov qui ont donnés les meilleurs taux de reconnaissance dans différents corpus. Notre calculatrice (fig. 3.15) fonctionne en mode mono locuteur avec le choix de parole isolée ou parole continue. Elle contient les boutons des chiffres de 0 à 9, les boutons des opérations, de recule (<-) et le bouton clear pour qu'on puisse

intervenir. Pour commencer le traitement il faut appuyer sur إبدء الإملَاءَ, et pour terminer on presse أَوقِف الإملَاءَ.

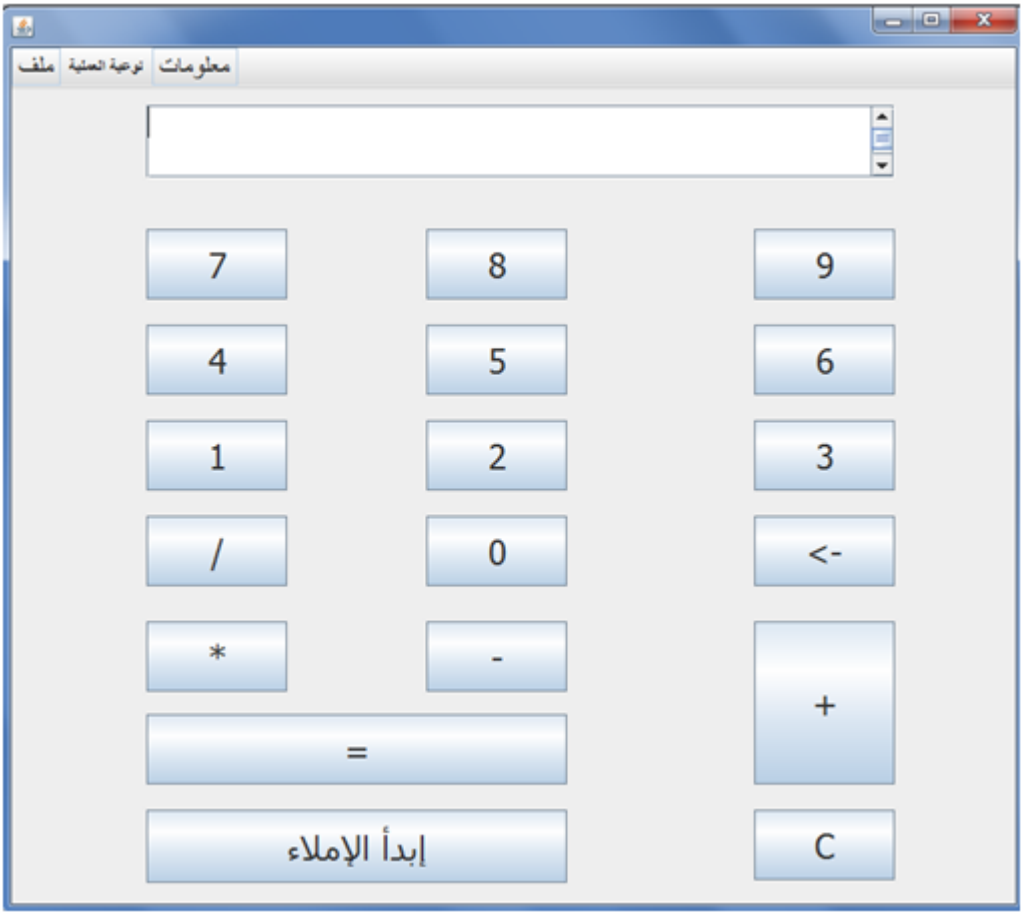

FIGURE 3.15 – Calculatrice vocale

Par défaut la calculatrice fait la reconnaissance de mots isolés, et pour la rendre de mots continus il faut aller au menu نَوعِيَة العَمَلِيَة et sélectionner 'parole continue' et ensuite ne faire entrer que les mots respectant la grammaire de A op B vue précédemment.

3.16 Conclusion

Dans ce chapitre nous avons réalisé un ensemble d'experiences pour examiner la différence entre la reconnaissance des mots connectés et des mots continus, en tenant compte de deux critères : la taille du corpus d'apprentissage et le type de traitement. Pour la classification, nous avons utilisé les chaines de Markov cachées (HMM), les plus adaptées au traitement de la parole ; les traitements associés aux HMMs ont été fait par l'outil HTK.

D'après les résultats de nos expériences, nous pouvons dire que la reconnaissance basée sur les mots continues est bien meilleure que celle des mots connectés en mode parole isolée car pour la première, avec une petite base d'apprentissage nous avons obtenus un taux de réussite acceptable par contre à la deuxième nous avons besoin de beaucoup plus de données d'apprentissage. Dans le mode de la parole continue les mots connectés se sont bien comportés et montre une concurrence avec les mots continus. La diminution de la reconnaissance de la parole continue peut se traduire par une faiblesse de la base de test.

Conclusion générale

Dans ce travail nous avons abordé un domaine en cours d'expansion cette dernière décennie : c'est la reconnaissance automatique de la parole et particulièrement en arabe. Après avoir lu différents documents sur ce domaine, nous avons pris le choix de travailler avec les chaines de Markov cachées (HMM) qui représentent un outil très robuste en s'appuyant sur des fondements mathématiques très solides et qui se caractérise par la notion d'états/transitions ; qui permet de traiter les phénomènes temporels dont la parole fait partie. Nous avons opté pour l'outil HTK afin de manipuler les HMMs.

Nous avons construit un système de reconnaissance d'un vocabulaire d'une calculatrice vocale en arabe. Ce système se compose d'une base d'apprentissage et d'une base de test. Les données sont représentées par des vecteurs caractéristiques de type MFCC. Nous avons mis en place un classifieur HMM pour lequel nous avons construit plusieurs modèles ainsi que leur apprentissage. Les testes réalisés nous ont donné les résultats vus au troisième chapitre.

Ce projet nous a permis d'apprendre et surtout de toucher à plusieurs domaines tels que le traitement de signal, la programmation, le traitement de la langue, etc.

Perspectives

Ce travail peut être perfectionné en passant du monolocuteurs aux multilocuteurs et en élargissant le vocabulaire. Ainsi une intervention des experts linguistes peut mieux modéliser les classes de mots, syllabes, ou phonèmes.

Nous pouvons aussi utiliser les améliorations en traitement de signale sur la méthode MFCC. Et refaire le travail sur d'autres techniques de classification et en introduisant les hybridations.

Annexe A

L'outil HTK

HTK ou Hidden Markov Model ToolKit est un outil puissant, développé par Cambridge University Engineering Department (CUED), de construction et de manipulation des modèles de Markov cachés. HTK est principalement employé pour la recherche de reconnaissance de la parole bien qu'il ait été employé pour nombreux d'autres applications comprenant la recherche dans la synthèse de la parole, la reconnaissance de caractères et l'ordonnancement d'ADN.

HTK se compose d'un ensemble de bibliothèque et les outils disponibles développées en langage C. Les outils fournissent les équipements sophistiqués pour l'analyse de la parole, apprentissage des HMMs, expériences et analyse des résultats. La figure (fig. A.1) résume les processus d'apprentissage et de décodage utilisé dans ce manuscrit.

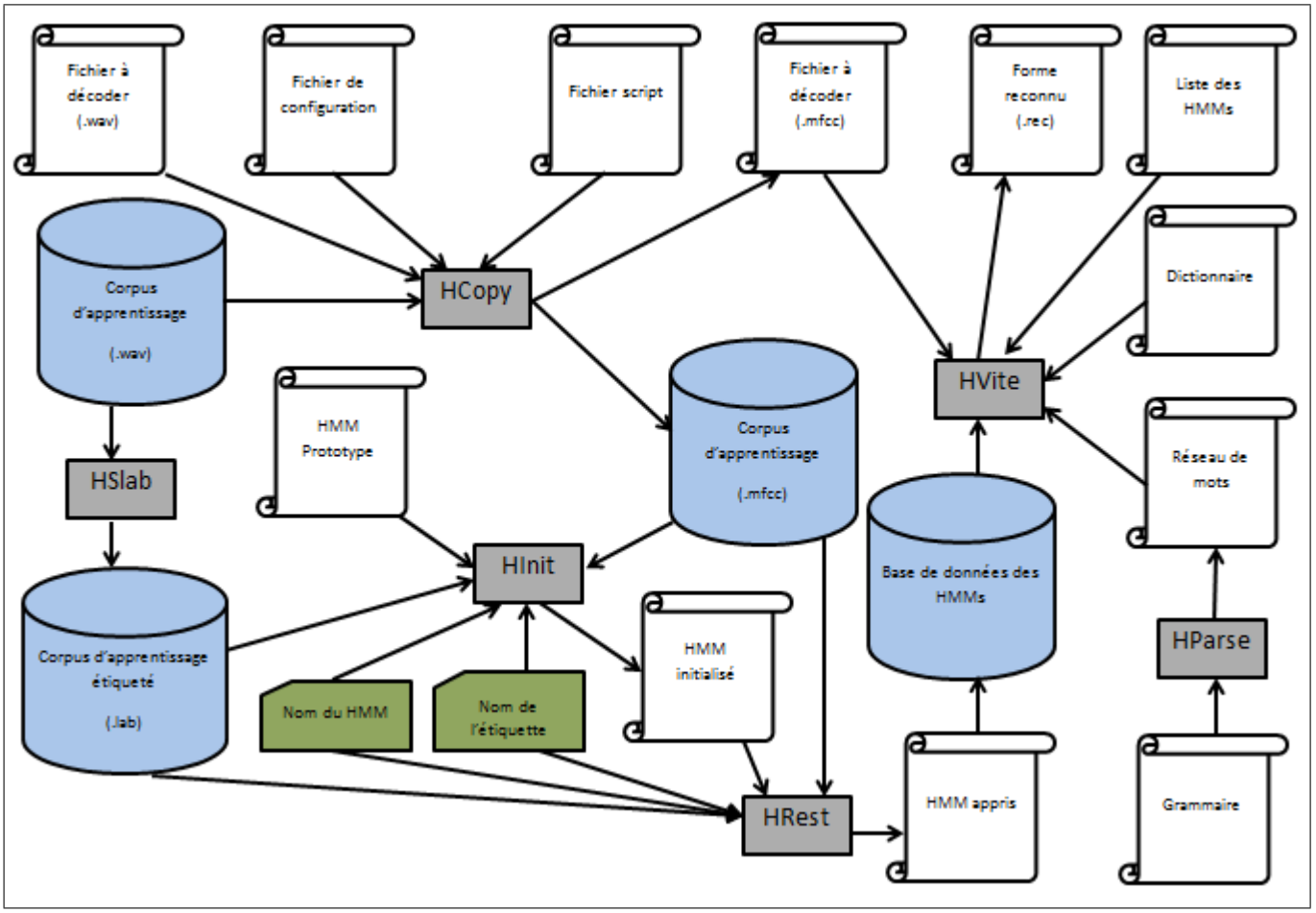

FIGURE A.1 – Fonctionnement du HTK

Références bibliographiques

[1] C.E Shannon and W.Weaver. The mathematical theory of communication. Urbana : University of Illinois Press, 1949.

[2] A. Muhammad. Alaswaat Alaghawaiyah. Daar Alfalah, 1990. Jordan.

[3] M. Elshafei. Toward an arabic text-to-speech system. 1991. vol. 4B no. 16,pp. 565–583.

[4] D. E. Kouloughli. Sur la structure interne des syllabes «lourdes» en arabe classique. 1986.

[5] S. Baloul. Développement d'un système automatique de synthèse de la parole à partir du texte arabe standard voyellé. PhD thesis, 2003. Thèse de Doctorat.

[6] http ://veloschola.e-monsite.com/pages/etudiants-de-langue/alphabet-phonetique-international-la-langue-arabe.html. université de Biskra.

[7] J.Ramírez and al. Speech/non-speech discrimination based on contextual information integrated bispectrum lrt. august 2006. VOL. 13, NO. 8.

[8] Van Den Heuvel H., Rietveld T., and Cranen B. Methodological aspects of segment and speaker-related variability. a study of segmentai durations in dutch. 1994. no 22, pp 389-406.

[9] Atal B. S. Text-independent speaker recognition. April 1972. Paper presented at the Program of the 83rd meeting of the Acoustical Society of America , Buffalo, NY, USA.

[10] T.Matsui and S.Furui. Text-independent speaker recognition using vocal tract and pitch information. 1990. pp 13 7-140.

[11] A.Malraux. `http://andremalrauxtpeson.e-monsite.com/pages/la-physique-du-son/l-intensite-du-son.html`.

[12] Bernard Gosselin. Représentation de l'information et quantification des signaux. Faculté Polytechniques de Mons, 2000. Belgique.

[13] Furui. Ceptral analysis technique for automatic speaker verification. 1981. volume 29, pages 254-272.

[14] Y.Ben Ayed. Détection de mots clés dans un flux de parole. PhD thesis, décembre 2003.

[15] Attabi yazid. Reconnaissance automatique des émotions à partir du signal acoustique. PhD thesis, Février 2008.

[16] O'Shaughnessy. Speech communication human and machine. 2000. second edition, New York, USA.

[17] Huang and al. Spoken language processing : a guide to theory, algorithm, and system development. 2001. United states of America.

[18] julien michot. `http://www.webfractales.org/RapportMMC/node4.html`, aout 2006.

[19] Florian AGEN and Julien MICHOT. Projet de Mathématiques :Chaînes de Markov cachées Algorithme de Baum-Welch. Université François Rabelois TOURS, Jan 2005.

[20] S. Ramdane, B. Taconet, and A. Zahour. Apprentissage dynamique du nombre d'états d'un modèle de markov caché à observations continues. 2003. 19ième Colloque sur le traitement du signal et des images.

[21] T. BROUARD, M. Slimane, G. Venturini, and J. P. ASSELIN DE BEAUVILLE. Apprentissage du nombre d'états d'une chaîne de markov cachée pour la reconnaissance d'images. 1997. 16ième Colloque sur le traitement du signal et des images.

[22] Nikolai Shokhirev. Hidden markov models, Février 2010. `http://www.shokhirev.com/nikolai.html`.

[23] Vincent BARRA. Apprentissage. Institut Supérieur d'Informatique, de Modélisation et de leurs Applications, 2006.

[24] LAWRENCE R. RABINER and FELLOW. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, février 1989. USA.

[25] M.J. Castro-Bleda and al. Efficient viterbi algorithms for lexical tree based models. 2007. Valencia, Spain.

[26] S.J. Young, N.H.Russell, and J.H.S Thornton. Token passing : a simple conceptual model for connected speech ecognition systems. July 1989.

[27] MURAT ALİ ÇÖMEZ. LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION FOR TURKISH USING HTK. PhD thesis, June 2003.

[28] Lilia Lazli and Mohamed Tayeb Laskri. Nouvelle méthode de fusion de données pour l'apprentissage des systèmes hybrides mmc/rna. novembre 2005.

[29] Emilie POISSON. Architecture et Apprentissage d'un Système Hybride Neuro-Markovien pour la Reconnaissance de l'Écriture Manuscrite En-Ligne. PhD thesis, ECOLE DOCTORALE STIM, décembre 2005. Université de Nante, France.

[30] Shun-Zheng Yu. Hidden semi-markov models. 2010. PR China.

[31] Y. Xie, S. Tang, C. Tang, and X. Huang. An efficient algorithm for parameterizing hsmm with gaussian and gamma distributions. 2012. China, USA.

[32] Djeffar Abdelhamid and al. Un système de tri automatique des dattes par svm. International Conference On Industrial Engineering and Manufacturing ICIEM'10, May 2010. Batna, Algeria.

[33] Jaume Padrell-Sendra, Dario Martin-Iglesias, and Fernando Diaz de Maria. Support vector machines for continuous speech recognition. Septembre 2006. Florence, Italy.

[34] H.Sakoe and S.Chiba. A dynamic programming approach to continious recognition. 1971. Budapest,Hungary.

[35] Ravinder Kumar. Comparaison of hmm and dtw for isolated word recognition system of punjabi language. 2010.

[36] mc-chapitre-7-tdnn.pdf.

[37] Asmaa OURDIGHI and Abeldelkader BENYETTOU. L'intégration des algorithmes génétiques dans l'apprentissage des réseaux de neurones à délais temporels adaptatifs. 2007. Algérie.

[38] Mansour M.Alghmadi. Kacst arabic phonetics database. 2004. Riyadh, Kingdom of Saudi Arabia.

[39] G.Droua-Hamdani and al. Algerian arabic speech database (algasd). decembre 2010.

[40] Zaïz Fouzi and al. Calculatrice vocale basée sur les svm.

[41] L. Wang, K. Chen, and Y.S. Ong. Boosting input/output hidden markov models for sequence classification. 2005. United Kingdom.

[42] Mounir Gragy. Rapport de Projet sous HTK : Reconnaissance de mots isolés Et Reconnaissance de mots connectés. Université Mohammed 1er Oujda, 2006.

[43] S.Young and al. The HTK Book (for HTK version 3.4). Cambridge University Engineering Department, december 2006.

[44] Nicolas Moreau. HTK (v.3.1) : Basic Tutorial, February 2002.