REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

Faculté de **T**echnologie

Département du **G**énie **E**lectrique

Mémoire de fin d'étude

Filière : Télécommunications

Spécialité : Systèmes de télécommunications

HMMs à matrice de covariance pleine Vs HMMs à matrice de covariance diagonale, Application à la reconnaissance de la parole

Présenté par :

Chikhi Farida

Guettache Nora

Proposé et Dirigé par :

Mr Raggai Djahid

Promotion 2022

Dédicaces

Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie.

Je dédie ce modeste travail :

A mes très chers parents, source de vie, d'amour et d'affection Mon père ***M'hamed,*** *et ma mère* ***Meriem****, que je n'arriverai jamais à exprimer ma gratitude et ma reconnaissance, pour ses amours ses soutiens tout au long de mes études et leurs prières, leur encouragement et qu'aucune dédicace ne pourra compenser leur sacrifice pour mon bien être et mon bonheur.*

A la famille Chikhi et Sahraoui surtout Mes sœurs et Mes frères : ***Khadîdja, Fatiha, lamia, Missoum*** *et* ***Karim*** *source de joie et de bonheur*

A mes amis et collègue : ***Thanae, lameya, Amina,Moussa***

En particulier à ma chère amie avant d'être binôme ***Guettache Nora*** *qui n'a jamais manqué de travailler.*

A notre encadreur ***Mr.Raggai Djahid***

A vous cher lecteur

Et à tous ceux qui nous ont soutenus pour l'achèvement de ce projet avec réussite.

Farida

Dédicaces

Avant tout, je tiens à remercier le bon dieu, et l'unique qui m'offre le courage et la volonté nécessaire pour affronter les différentes difficultés de la vie.

Je dédie ce modeste travail :

A mes très chers parents, source de vie, d'amour et d'affection Mon père ***Ramdan,*** *ma mère* ***Houria*** *que Dieu lui fasse miséricorde, et ma belle-mère* ***Meriem*** *que je n'arriverai jamais à exprimer ma gratitude et ma reconnaissance, pour ses amours ses soutiens tout au long de mes études et leurs prières, leur encouragement et qu'aucune dédicace ne pourra compenser leur sacrifice pour mon bien être et mon bonheur.*

A la famille ***Guettache*** *surtout Mes sœurs et Mes frères :* ***Bilel, Ahmed, Nadia et Safaa*** *source de joie et de bonheur*

A mes amis et collègue : ***Feriel, Thanae, lameya, Imen***

En particulier à ma chère amie avant d'être binôme ***Chikhi Farida*** *qui n'a jamais manqué de travailler.*

A notre encadreur ***Mr.Raggai Djahid***

A vous cher lecteur

Et à tous ceux qui nous ont soutenus pour l'achèvement de ce projet avec réussite.

Nora

Remerciements

Tout d'abord je Remercie le bon Dieu le tout puissant pour la bonne santé, la volonté et de la patience qu'il nous a donné pour accomplir ce travail.

*Je remercie très sincèrement '' **Mr. Raggai Djahid** '' L'encadreur de ce travail, pour ses conseils pertinents, Ses orientations judicieuses, et sa patience.*

Nous tenons à remercier également les membres de jury

De Bien vouloir accepter d'examiner et de juger ce modeste travail.

Nous remercions aussi tous ceux qui ont contribué de près ou de loin à la réalisation de ce mémoire de fin d'étude, sans oublier d'adresser nos remerciements à tous les enseignants du département Génie électrique notamment la spécialité télécommunication .

Farida et Nora

ملخص

هذا العمل هو دراسة مقارنة بين سلاسل ماركوف المخفية التي تستخدم غوسية ذات مصفوفة تغاير قطرية وتلك التي تستخدم مصفوفات تغاير مملوءة. تم اجراء تجارب التعرف على الكلام في الحالتين للتحقق من مدى مشروعية الفرضية التبسيطية التي تقضي بتبني مصفوفة قطرية. تظهر النتائج بان هده الفرضية مبررة عند استخدام معاملات مستقلة في مرحلة تشفير إشارة الكلام مثل معاملات MFCC و LPCEPSTRA بينما تكون هده الفرضية مبررة عند استخدام معاملات غير مستقلة مثل LPC

الكلمات المفتاحية :التعرف الآلي على الكلام، نماذج ماركوف المخفية، أداةHTK .

Résumé

Ce travail est une étude comparative entre les modèles de Markov cachés utilisant des gaussiennes à matrice de covariance diagonale et ceux utilisant des matrices de covariance pleine. Des tests de reconnaissance ont été réalisés dans les deux cas de figure pour voir si l'hypothèse simplificatrice qui consiste à adopter une matrice diagonale était justifiée. Les résultats montrent que cette hypothèse est justifiée lors de l'utilisation de coefficients décorrélés dans la phase de codage de la parole, c'est notamment le cas des coefficients MFCC et LPCEPSTRA. Cependant dans le cas de 0.

coefficients corrélés comme LPC, cette hypothèse n'est plus justifiée.

Mots-clés : reconnaissance automatique de la parole, modèles de Markov cachés, l'outil HTK

Abstract

This work is a comparative study between Hidden Markov Models using gaussians that have diagonal covariance matrices and those using full covariance matrices. Recognition tests have been performed in the two cases to see whether the simplifying hypothesis consisting of adopting a diagonal matrice is justified. The obtained results show that this hypothesis is justified when using decorrelated coefficients during the speech coding process. This is notably the case when using MFCC or LPCEPSTRA coefficients. However, in the case of LPC coefficients this hypothesis is not justified.

Keywords: automatic speech recognition, hidden Markov models, HTK tool

Sommaire

Liste des figures……..……I

Liste des tableaux…….…...II

Abréviations.....…..III

Liste de références bibliographiques……..…… IV

Introduction Générale.....…...1

CHAPITRE I: Généralités sur le traitement de parole……...……4

1. Introduction : 4
2. Le son : 4
4. La perception de la parole : 6
5. Caractéristiques phonétiques : 7
6. Le niveau acoustique : 8
   - 6.1 Echantillonnage : 8
   - 6.2 La quantification : 9
   - 6.3 Le codage : 11
     - 6.3.1 Coefficients MFCC : 11
     - 6.3.2 Coefficients de prédiction linéaire LPC : 12
     - 6.3.3 Coefficients LPCEPSTRA : 12
7. La reconnaissance : 12
   - 7.2 Les systèmes de reconnaissance de la parole : 13
8. Techniques de reconnaissance de la parole : 13
   - 8.1 Dynamic Time Warping : 13
   - 8.2 Hidden Markov Models (les modèles de Markov cachés) : 14
   - 8.3 Les réseaux de neurones : 14
   - 8.4 Mélange de gaussiennes GMM : 14
9. Conclusion : 14

CHAPITRE II: Reconnaissance de la parole par HMMs……..…….....16

1. Introduction : 15
2. Modèles de Markov cachés (HMM) : 15
   - 2.1 Principes d'usage des HMMs en RAP : 15
   - 2.2 Topologies des HMM : 16
3. Reconnaissance des mots isolés : 17

3.1 Les paramètres du HMM continu : 20
4. Les trois Problèmes en HMMs : 21
5. Calcul de la vraisemblance d'une séquence : 22
6. L'apprentissage : 22
7. La reconnaissance : 24
8. Conclusion : 24
CHAPITRE III :Résultats et Interprétations……..……..…..……27
1. Introduction : 25
2.1 Vocabulaire : 25
2.2 Système d'étiquetage des fichiers de la base : 26
3. Description de l'outil de reconnaissance HTK : 26
4. Réalisation du système de reconnaissance par HTK : 26
4.1 Le codage 26
4.2 la topologie du HMM choisie : 29
4.3 Initialisationdes modèles : 30
4.4 Apprentissage : 31
4.5 Définition de la grammaire et du dictionnaire : 32
4.6 La reconnaissance : 35
5. Expérimentation 36
5.1 Performances de reconnaissance obtenues pour les MFCC : 37
5.2 Performances de reconnaissance obtenues pour les LPC : 37
5.3 Performances de reconnaissance obtenues pour les LPCEPSTRA : 38
6. Conclusion : 39
Conclusion générale……..……..……..43

Liste des figures

Figure (I. 1) : Les étapes de production du son 6
Figure (I. 2] : l'appareil phonatoire [2] 6
Figure (I. 3) : l'appareil auditif humain 7
Figure (I. 4) : Prétraitement du signal de parole [2] 8
Figure (I. 5) : l'opération de quantificati on [4] 9
Figure (I. 6) : exemple de représentation temporelle de signaux de parole [3] 10
Figure (I. 7) : Exemple de son voisé (haut) et non-voisé (bas).[3] 10
Figure (I. 8) : banc de filtres 11
Figure (I. 9) : analyse cepstrale 12

Figure (II. 1) : le codage et le décodage d'un message 16
Figure (II. 2) : la topologie du HMM utilisée 17
Figure (II. 3) : Le modèle de génération de Markov 18
Figure (II. 4) : utilisation des HMMs dans la reconnaissance des mots isolés. 20
Figure (II. 5) : simple gauche- droite HMM 20

Figure (III. 1) : procédure de codage 27
Figure (III. 2) : utilisation de HCopy pour un groupe de fichiers 27
Figure (III. 3) : fichier de configuration 27
Figure (III. 4) : représente un exemple d'un script 28
Figure (III. 5) : la topologie choisie pour les HMMS 29
Figure (III. 6) : Fichier de description HMM 30
Figure (III. 7) : fonctionnement de HCompv 31
Figure (III. 8) : fonctionnement de HRest 32
Figure (III. 9) : le fichier de la grammaire 33
Figure (III. 10) : le fichier du dictionnaire 33
Figure (III. 11) : le réseau de connexion entre mot 34
Figure (III. 12) : processus de codage et reconnaissance 35
Figure (III. 13) : Exemple d'un fichier reco 36

Liste des tableaux

Tableau(III.1): Taux de reconnaissance dans le cas MFCC……....37

Tableau(III.2): Taux de reconnaissance dans le cas LPC……38

Tableau(III.3): Taux de reconnaissance dans le cas LPCEPSTRA…….38

Abréviations

ADN : Acide Désoxyribonucléique

API : Alphabet Phonétique International

ARPA : Advanced Research Project Agency

DTW: Dynamic time Warping

DRPA: Defense Advanced Research Projects Agency

GMM: Gaussian Mixture Models

HMM: Hidden Markov Models

HTK: Hidden Markov Model Toolkit

IDCT : Invers Discret Cosinus Transform

LPC : Analyse Predictive Linéaire

MFCC : Mel Frequency Cepstral coefficient

MMC : Modèle De Markov Caché

RAP : Reconnaissance Automatique De La Parole

RAP : Reconnaissance Automatique De La Parole

RAL : Reconnaissance Automatique De Locuteur

SAPI : Speech Application Programming Interface

SRAP : système de reconnaissance automatique de la parole

VLSI : Very Large Scale Integration

Introduction Générale

Le signal de la parole est l'un des signaux les plus complexes, il n'est pas facile de le caractériser par un modèle simple. L'une des principales sources de difficulté dans l'analyse du signal vocal est sa grande variabilité. Les précurseurs dans le domaine du traitement de la parole, pensaient que la parole était une simple juxtaposition d'éléments appelés "caractéristiques distinctives" ou invariants (phonèmes), et qu'à partir d'échantillons représentant les invariants d'une langue on pouvait reconstituer ou reconnaître n'importe quelle phrase. Le traitement de la parole est un domaine vaste faisant intervenir plusieurs disciplines telles que l'acoustique, le traitement du signal, la linguistique et bien d'autres disciplines. On distingue plusieurs faisceaux de recherche qui dérivent du domaine de la parole à savoir : l'analyse, la reconnaissance et la synthèse de parole. Les applications qui en découlent sont nombreuses et variées parmi lesquelles on peut citer : les systèmes nécessitant une interaction homme-machine (commande vocale), criminologie et domaine médical.

Voici quelques dates importantes qui ont marqué l'évolution dans ce domaine :

- 1952 : Reconnaissance de 10 chiffres pour un mono-locuteur par un dispositif électronique câblé.
- 1960 : Utilisation des méthodes numériques.
- 1965 : Reconnaissance de phénomènes en parole continue.
- 1968 : Reconnaissance de mots isolés par des systèmes implantés sur gros ordinateur (500 mots).
- 1969 : Utilisation d'information linguistique.
- 1970 : Méthodes basées sur la programmation dynamique.
- 1971 : Lancement du projet ARPA aux USA visant à tester la faisabilité de la compréhension automatique de la parole continue avec des contraintes raisonnables.
- 1972 : 1er appareil commercialisé de reconnaissance des mots isolés.
- 1976 : Fin de projet ARPA, les systèmes opérationnels sont HARPY, HEARSAY1et2 et HWIM.
- 1978 : Commercialisation d'un système de reconnaissance à micro-processeur sur une carte circuits imprimés.
- 1981 : Utilisation de circuits intégrés VLSI (Very Large Scale Integration) spécifique le traitement de la parole. / Système de reconnaissance de mots sur un circuit VLSI.
- 1983 : première mondiale de commande vocale à bord d'un avion de chasse en France.
- 1985 : Commercialisation des 1 er système de reconnaissance de plus milliers de mots.

- 1986 : Lancement du projet japonais ATR utilisant la traduction automatique en temps réel par le téléphone.
- 1988 : Apparition des premières machines de dictée vocale par mots isolés.
- 1989 : Recrudescence des modèles connexionniste neuro-mimétiques.
- 1990 : Première véritables applications de dialogue orale homme-machine.
- 1993 : Premier SRAP de parole continue (langue allemande). fonctionnant en quasi temps réel présenté par Phillips à la conférence Eurospeech .
- 1994 : IBM lance une machine à dictée vocale en parole continue : IBM Voice Type – Dictée personnelle.
- 2000 : Microsoft s'associe à Lernout & Hauspie pour intégrer les solutions de reconnaissance vocale et de synthèse vocale de la société belge à son interface de programmation vocale nommée SAPI (Speech Application Programming Interface).
- 2006: Le DRPA (Defense Advanced Research Projects Agency). Prépare une super machine dédiée à la traduction ; Cet organisme ambitionne de développer un logiciel capable de traduire de façon quasi simultané l'arabe et le mandarine avec une précision accrue, l'appareil devrait être utilisé par les renseignements américains pour écouter et parcourir
- 2008 : Lance une application de recherche sur internet mettant en œuvre une fonctionnalité de reconnaissance vocale.
    - 2011 : Apple propose l'application Siri sur ses téléphones.
    - 2017 : Microsoft annonce égaler les performances de reconnaissance vocale des êtres humains.

Les approches prédominantes actuellement sont les réseaux à apprentissage profond ainsi que les modèles HMM (Hidden Markov Models) qui représentent l'état de l'art en RAP.

Les modèles de Markov cachés tirent principalement leur force du formalisme mathématique solide sur lequel ils reposent. Il faut cependant noter que lors de la mise au point d'un système de reconnaissance basé sur les HMMs, un certain nombre d'hypothèses est adopté, la plus populaire étant le choix de gaussiennes à matrice de covariance diagonale dont l'utilisation permet de diminuer amplement la complexité de calcul mais dont l'impact sur les performances de reconnaissance peut être significatif dans certains cas. L'objectif de ce travail est justement d'étudier la validité d'une telle hypothèse en utilisant trois schémas de codages (MFCC, LPC et LPCEPSTRA).

Ce mémoire suit la chronologie suivante :

Après une introduction générale, on expose dans le premier chapitre notions d'ordre général concernant le traitement de la parole, l'accent sera mis essentiellement sur les méthodes de codage ainsi que sur les méthodes de reconnaissance de la parole.

Le deuxième chapitre expose en détail le formalisme mathématique des HMMs ainsi que leur mise en œuvre dans un système de reconnaissance de la parole.

Le troisième chapitre est dédié aux tests de reconnaissance réalisés en utilisant plusieurs méthodes de codage et ce dans les deux cas de figure : adoption de l'hypothèse simplificatrice (matrice de covariance diagonale) et non adoption de l'hypothèse simplificatrice.

Enfin le mémoire sera couronné par une conclusion générale.



Généralités sur le traitement de la parole

1. Introduction :

Le présent chapitre expose des généralités sur la parole et son traitement. Loin de prétendre à l'exhaustivité, seules les notions de base ayant trait de près ou de loin avec notre travail seront exposées. La chronologie adoptée est la suivante :

Seront exposées en premier lieu les notions fondamentales telle que : le son, la production de la parole, la perception de la parole, analyse au niveau phonétique, analyse au niveau acoustique. Par suite il sera question de développer les méthodes de reconnaissance de la parole avec un accent mis sur la méthode adoptée dans ce travail à savoir les modèles de Markov cachés. Le chapitre sera clos par une conclusion.

2. Le son :

Le son est ce que l'oreille perçoit de la vibration d'un corps, il s'agit d'un ensemble d'ondes sonores. Une onde sonore est une suite de variations de la pression, dans un axe longitudinal, qui se propage dans un milieu élastique (gaz, liquide, solide) dans toutes les directions à partir du point d'émission. Selon la densité du milieu dans lequel le son se propage, celui-ci aura des caractéristiques physiques (vitesse, fréquence...) différentes. Si on devait donner une comparaison, l'air c'est un peu comme un ressort, ou un accordéon. Lorsqu'on provoque une vibration, il s'allonge et se contracte, mais chaque point finit toujours par revenir à sa place [1].

3. La production de la parole :

Le signal de parole est produit par notre appareil phonatoire. Il s'agit d'un système dynamique, dont le comportement à un moment donné dépend de ses états antérieurs. Le système est donc dépendant d'une variable paramétrable. La parole se distingue des autres sons par des caractéristiques acoustiques ayant leurs origines dans le mécanisme de production. L'appareil phonatoire est un organe d'une grande complexité mécanique. Il se compose de deux parties anatomiquement distinctes. Les poumons et le larynx, partie supérieure de la trachée artère, constituent l'essentiel du générateur sonore [2].
Les trois étapes de la production des sons de parole sont illustrées dans la figure (I.1) :

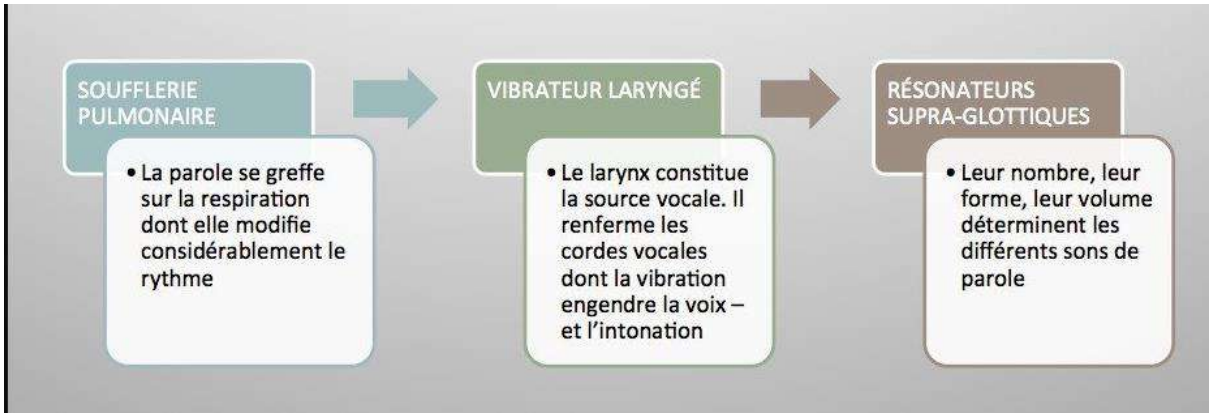


Figure (I.1) : Les étapes de production du son

En général, le traitement automatique de la parole utilise des sources de connaissance Phonétiques, Phonologiques, Prosodiques, Lexicales, Syntaxiques, Sémantiques.

La phonétique : Science qui étudie les caractéristiques physiques des sons sur trois plans complémentaires (articulatoire, acoustique, perceptif) [2].

La phonologie : Étudie l'aptitude linguistique en relation avec le son, en faisant abstraction de ses propriétés physiques. Elle définit un inventaire des unités de base (phonèmes) avec des contraintes de combinaison [2].

La prosodie : La prosodie peut être considérée comme une sorte de "ponctuation acoustique" de la parole. Elle recouvre les aspects liés à la hauteur de la voix, à l'intensité et à la durée des segments syllabiques. Son rôle dans la langue est multiple [2].

Le lexique : Les performances d'un système de reconnaissance sont affectées par la taille du vocabulaire et aussi par le degré de confusion entre les mots. Le dictionnaire doit être étudié de telle sorte qu'il autorise économiquement la représentation de toutes les prononciations envisageables des mots, mais aussi pour qu'il permettre d'accéder directement à tous les mots contenant la même syllabe ou le même trait acoustique, de telle sorte qu'il soit possible de générer les hypothèses des mots à partir des caractéristiques du signal dans l'analyse ascendante [2].

La syntaxe : Du point de vue de la langue, la syntaxe est l'ensemble des règles contraignant l'ordre des mots dans la phrase. Dans un système de compréhension, le but de la syntaxe est de réduire le nombre de phrases autorisées à partir du vocabulaire choisi [2].

La sémantique : La sémantique est définie d'un point de vue linguistique, comme la relation entre la forme des signes linguistiques, ou "signifiants", et ce qui est signifié, ou "signifiés". En reconnaissance de la parole la sémantique restreint la combinatoire syntaxique.

La figure (I.2) montre l'appareil phonatoire de l'être humain [2].

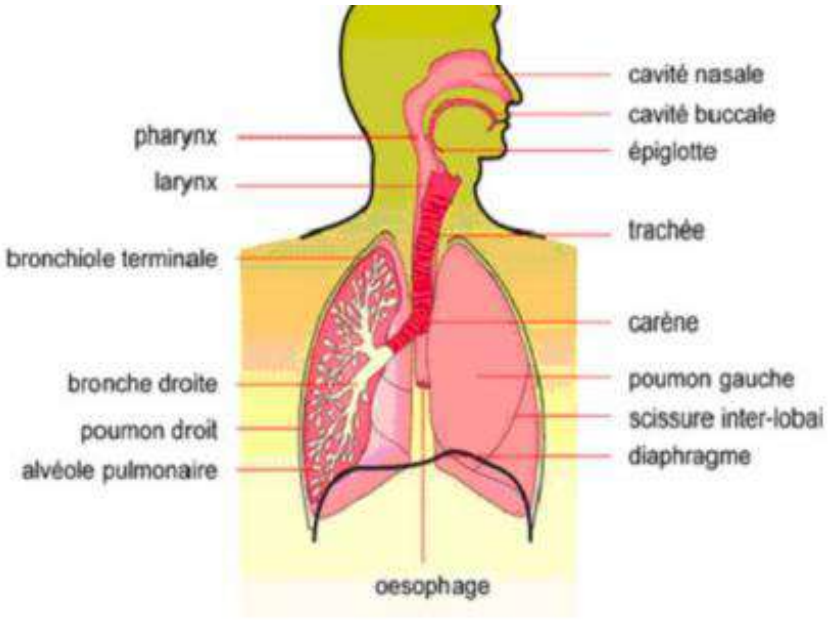


Figure (I. 2) : l'appareil phonatoire [2]

4. La perception de la parole :

Le système auditif reçoie les ondes sonores et les transforment en signaux électriques avant d'être acheminés vers notre cerveaux. L'appareil auditif humain est un ensemble de structures anatomiques d'origine et de nature très différentes. La figure (I.3) montre une coupe verticale de la tête humaine au niveau de l'os temporal, montrant les trois parties consécutives de l'appareil auditif : oreille externe, oreille moyenne et oreille interne.

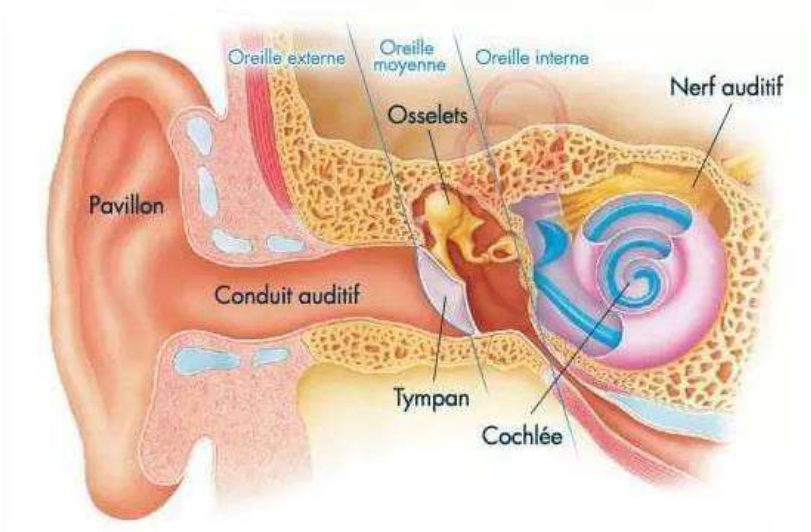


Figure (I.3) : l'appareil auditif humain

L'oreille externe : est composée du pavillon et du conduit auditif externe. C'est le pavillon de l'oreille qui **capte les vibrations sonores** pour les acheminer vers le tympan, via le conduit auditif.

L'oreille moyenne : est composée du tympan et de la chaîne des osselets, elle-même constituée du marteau, de l'enclume et de l'étrier, qui sont les plus petits os du corps humain. **L'oreille moyenne est un amplificateur qui propage à son tour les ondes sonores de la membrane tympanique jusqu'à l'oreille interne**, en passant par la chaîne des osselets

L'oreille interne : est composée du vestibule, organe de l'équilibre, et de la cochlée et ses 15 000 cellules sensorielles. Ce sont ces dernières qui **traduisent le message vibratoire en influx nerveux.** C'est alors que l'influx nerveux quitte l'oreille par le nerf auditif pour être transmis aux centres auditifs cérébraux où il va être décodé et interprété sous forme de son.

5. Caractéristiques phonétiques :

En phonologie, domaine de la linguistique, un phonème est la plus petite unité discrète ou distinctive que l'on puisse isoler par segmentation dans la chaîne parlée. Un phonème est en 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 (voir allophone). Les phones sont d'ailleurs les différentes réalisations d'un phonème.

On distingue :

- **Les voyelles :**[i.o.e.u.a…] différent de tous les autres sons par le degré d'ouverture du conduit vocale (et non, comme on l'entend souvent dire, par le degré d'activité des cordes vocales, déjà mentionné sous le terme de **voisement**). Si le conduit vocal est suffisamment ouvert pour que l'air poussé par les poumons le traverse sans obstacle, il y'a la production d'une voyelle, [3] les voyelles peuvent être rangées selon : La nasalité, l'ouverture du conduit vocal, la position de la constriction du conduit vocal et l'arrondissement des lèvres.
- **Les consonnes :** Les consonnes se prononcent avec un rétrécissement du passage de 1' air et sont classées selon: le voisement (selon que les cordes vocales vibrent ou non à leur passage), le mode d'articulation (occlusif, nasal, fricatif) et le lieu d'articulation (labiale, dentale, palatale).

L'alphabet phonétique international (API) associe des symboles phonétiques aux sons, de façon à permettre l'écriture compacte et universelle des prononciations. [3]

6. Le niveau acoustique :

Le niveau acoustique s'intéresse aux caractéristiques acoustiques du signal parole. Usuellement le traitement primaire (prétraitement) du signal de parole passe par les étapes illustrées dans la figure (I.4)

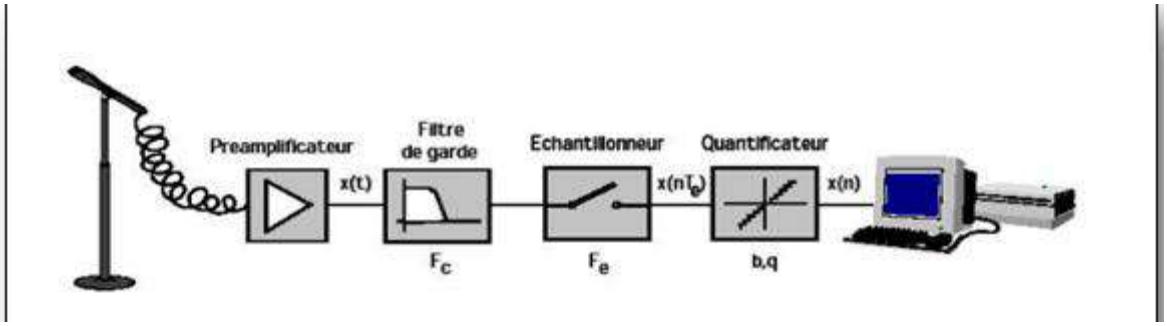


Figure (I.4) : Prétraitement du signal de parole [2]

6.1 Echantillonnage :

L'échantillonnage d'un signal continu est l'opération qui consiste à prélever des échantillons du signal pour obtenir un signal discret, c'est-à-dire une suite de nombres représentant le signal, dans le but de mémoriser, transmettre, ou traiter le signal [2].

Pour que le signal puisse être entièrement reconstruit à partir des échantillons, il faut et il suffit que : La fréquence d'échantillonnage soit strictement supérieure à deux fois la plus grande fréquence présente dans le spectre du signal continu (condition de Nyquist-Shannon).

6.2 La quantification :

La quantification est l'approximation de chaque valeur du signal s(t) par un multiple entier d'une quantité élémentaire q, appelée échelon de quantification. Si q est constant quelle que soit l'amplitude du signal, la quantification est dite uniforme. Cette opération revient à faire passer le signal dans un organe qui possède une caractéristique en marche d'escalier, comme le montre la figure 1.11 pour q = 1, et fournit le signal sq(t).[4]

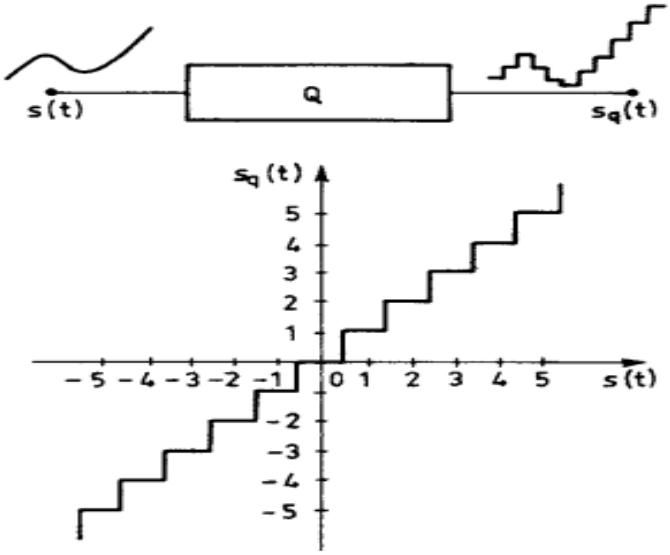


Figure (I.5) : l'opération de quantification [4]

La figure (I.6) représente l'évolution temporelle du signal vocal pour les mots 'parenthèse', et 'effacer'. On y constate une alternance de zones assez périodiques et de zones bruitées, appelées respectivement zones voisées et non-voisées. L'évolution temporelle ne fournit cependant pas directement les traits acoustiques du signal. Il est nécessaire, pour les obtenir, de mener à bien un ensemble de calculs.

Les sons voisés sont produits par l'excitation du conduit vocal par des impulsions périodiques dues aux vibrations des cordes vocales. Ce sont des sons ayant une forme quasi périodique dont la représentation temporelle est illustrée sur Figure (I.7)**.** Cette catégorie comprend, en plus des voyelles, d'autres groupes de phonèmes comme par exemple les consonnes nasales, les fricatives et les plosives.

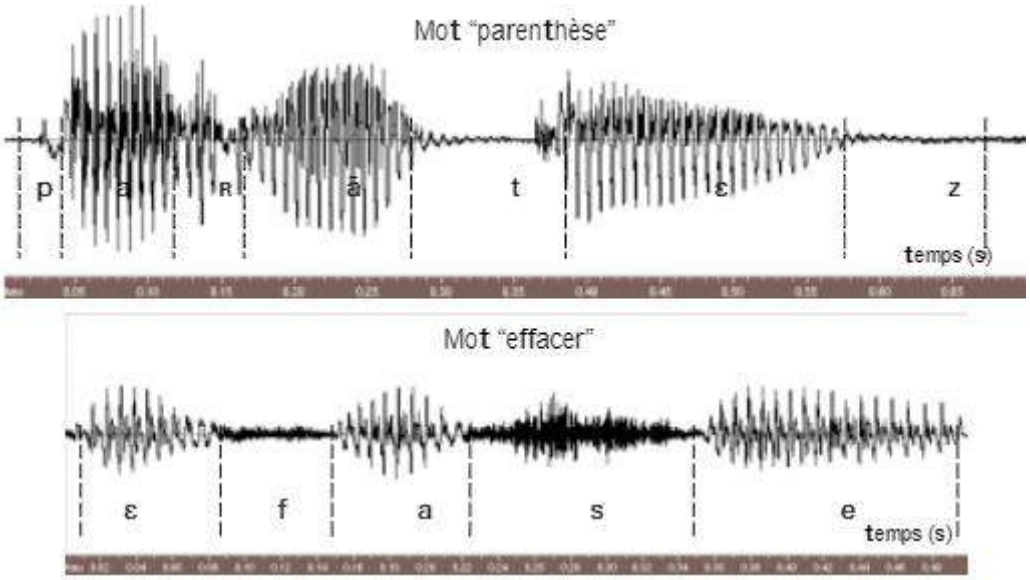


Figure (I.6) : exemple de représentation temporelle de signaux de parole [3]

Les sons non-voisés sont des signaux transitoires (comme les plosives) ou des signaux qui n'ont pas de composante périodique. Ils sont produits dans le premier cas par les lèvres et dans le deuxième cas par les turbulences de l'air passant par le conduit vocal. Dans ce dernier cas, la position des parties du conduit vocal modifie le spectre du signal non périodique.

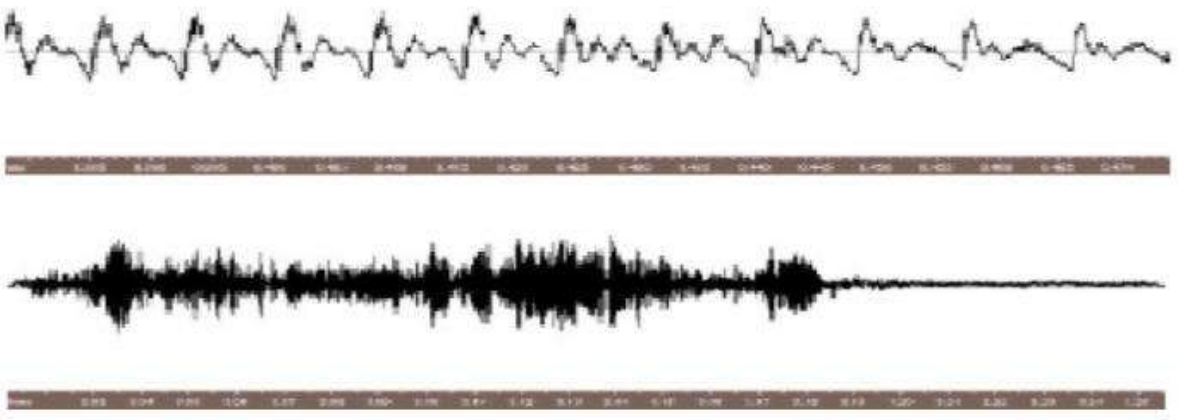


Figure (I.7) : Exemple de son voisé (haut) et non-voisé (bas).[3]

6.3 Le codage :

Par codage on entend l'ensemble des traitements visant à extraire les paramètres pertinents du signal de parole. Plus précisément il s'agit de l'extraction des coefficients MFCC, LPC ou LPCEPSTRA à partir de trames d'une vingtaine de millisecondes du signal temporel.

6.3.1 Coefficients MFCC :

Puisque la résolution en fréquence de l'appareil auditif humain n'est pas linéaire et en partant de l'idée que le système de reconnaissance aurait de meilleures performances lorsqu'il imite cet appareil humain, on effectue une analyse du signal en bancs de filtres. Ces bancs de filtres auront des largeurs non uniformes comme illustré sur la figure (I.8)

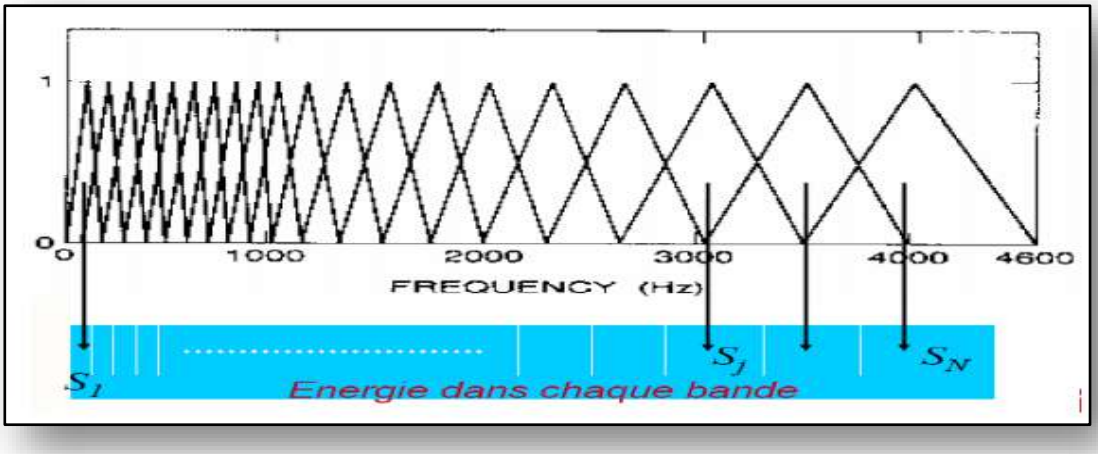


Figure (I.8) : banc de filtres

On calcule ensuite le logarithme de l'énergie dans chaque filtre et on effectue enfin la transformée en cosinus inverse IDCT. La figure (I.9) illustre le processus entier de calcul

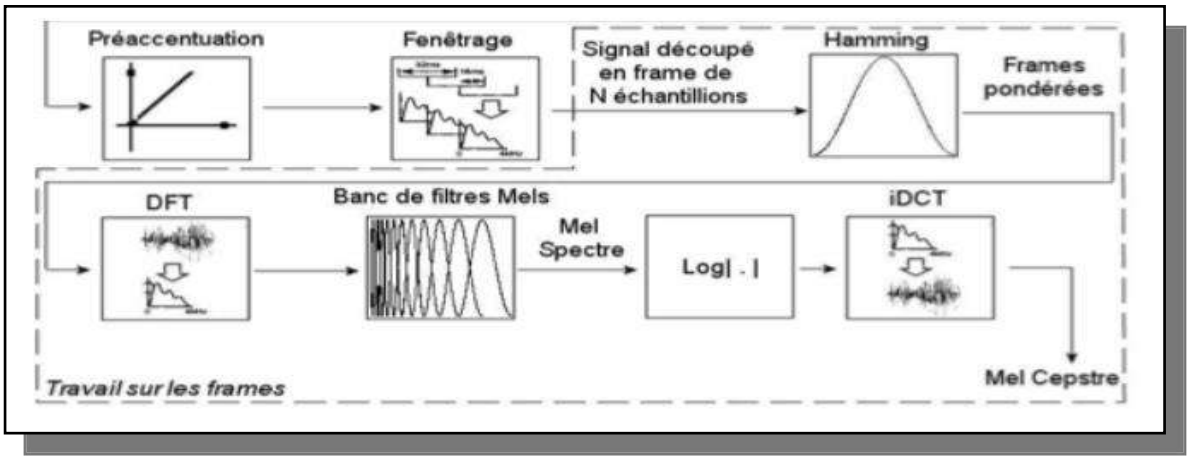


Figure (I.9) : analyse cepstrale

6.3.2 Coefficients de prédiction linéaire LPC :

Dans cette approche le conduit vocal est modélisé par un filtre tout pôles dont la fonction de transfert [5] est donnée par :

$$\boldsymbol{H(z) = \frac{1}{\sum_{i=0}^{p} a_i z^{-i}}} \qquad \ldots (\boldsymbol{I.1})$$

Avec : p nombre de pôles et $a_0 = 1$

Les coefficients $\{a_i\}$ du filtre sont choisis pour minimiser l'erreur quadratique moyenne sommée sur toute la fenêtre d'analyse.

6.3.3 Coefficients LPCEPSTRA :

Il s'agit de faire la même démarche entreprise dans le calcul des MFCC mais en remplaçant le spectre qui a été obtenu par le banc de filtres par le spectre de prédiction linéaire.

7. La reconnaissance :

C'est la transcription du signal vocal en un ensemble d'unités phonétiques représentant le message. L'objectif étant de faciliter l'interaction entre l'homme et la machine. Comme souligné dans l'introduction générale, la reconnaissance de la parole fait appel aux connaissances de plusieurs disciplines : l'anatomie (les fonctions de l'appareil phonatoire et de l'oreille), la phonétique, le traitement du signal, la linguistique,

l'informatique, l'intelligence artificielle et les statistiques. Comme première classification, on parle de la reconnaissance du locuteur ainsi que de la reconnaissance de la parole.

7.1 Les systèmes de reconnaissance du locuteur :

La reconnaissance automatique du locuteur (RAL) est généralement décrite comme l'identification d'une personne par sa voix. Il est possible de classifier les systèmes de reconnaissance automatique du locuteur selon leur dépendance au texte prononcé, la tâche ciblée ou leur domaine d'application. On présente dans les sous-sections suivantes, des classifications possibles selon ces critères et on finit par décrire la structure d'un système de vérification du locuteur.[6]

7.2 Les systèmes de reconnaissance de la parole :

Les systèmes de RAP sont en général conçus pour des applications spécifiques dans un environnement contrôlé (vocabulaire limité et faible bruit ambiant). On se dirige toutefois vers des systèmes à vocabulaire plus important et pouvant fonctionner dans des environnements bruités. Lorsque le vocabulaire du système est de petite taille, l'approche globale est adoptée et la plus petite unité à reconnaitre est le 'mot'. Lorsque la taille du vocabulaire est importante, on cherche à modéliser de plus petites unités phonétiques (phonèmes, syllabes) dont la concaténation produit n'importe quel mot.

Dans le domaine de la reconnaissance automatique de la parole, on distingue trois grandes catégories de systèmes :

- Les systèmes de commandes vocales
- Les systèmes de dictée automatique
- Les systèmes de compréhension.

8. Techniques de reconnaissance de la parole :

8.1 Dynamic Time Warping :

Dynamic Time Warping, est une méthode fondée sur un principe de comparaison d'un signal à analyser avec un ensemble de signaux stockés dans une base de référence. Le signal à analyser est comparé avec chacune des références et est classé en fonction de sa proximité avec une des références stockées. Le DTW est en fait une application au domaine de la

reconnaissance de la parole de la méthode plus générale de la programmation dynamique. Elle peut ainsi être vue comme un problème de cheminement dans un graphe [7].

8.2 Hidden Markov Models (les modèles de Markov cachés) :

Les modèles de Markov cachés (HMM pour Hidden Markov Model) [Rabiner, 1989] sont des modèles statistiques dédiés à la modélisation des processus évoluant au cours du temps [Scharf, 1991 ; Rabiner et Juang, 1993]. Dans un modèle de Markov caché, nous avons connaissance d'une certaine séquence d'observations mais on ne sait pas exactement de quelle façon les observations ont été produites (ce sont les symboles émis qui sont observables). En fait, la séquence d'observation dépend d'une séquence d'états non observables (cachées), ce qui explique le nom donné à ce modèle.[8]

8.3 Les réseaux de neurones :

Les réseaux neuronaux sont inspirés des neurones biologiques. Le système est organisé en couches comprenant des éléments de calcul interconnectés. Il existe théoriquement une proportionnalité entre le nombre de couches et la capacité du réseau à apprendre des taches plus complexes. L'augmentation de la puissance de calcul et de stockage apportés par la technologie actuelle a donné un regain d'intérêt croissant à cette approche.

8.4 Mélange de gaussiennes GMM :

Le modèle GMM permet une modélisation probabiliste continue et efficace de l'espace acoustique d'un locuteur. Les discontinuités spectrales présentent dans tous les autres algorithmes de conversion de voix disparaissent et le naturel de la voix convertie est amélioré.

9. Conclusion :

Ce chapitre avait pour objectif d'exposer brièvement les notions générales de base en traitement de la parole. Un léger accent a été mis sur les concepts ayant un lien avec notre travail. Ainsi donc nous avons abordé : la parole et sa production, sa perception, son codage et sa reconnaissance. La méthode de reconnaissance adoptée dans notre travail étant basée sur les HMMs, nous allons consacrer le deuxième chapitre entièrement à l'utilisation de cette méthode en reconnaissance de la parole.



Reconnaissance de la parole par HMMs

1. Introduction :

Les modèles de Markov cachés HMMs permettent essentiellement de modéliser des phénomènes stochastiques, ils possèdent des capacités remarquables à capter les statistiques d'un phénomène stochastique. Le succès de leur utilisation ne se résume pas seulement à la reconnaissance de la parole qui fait l'objet de notre travail mais s'étend à de nombreux domaines tels que la reconnaissance de formes, la synthèse de la parole ...etc. Nous nous proposons dans ce chapitre d'exposer le formalisme mathématique de cette puissante approche et d'illustrer également son utilisation dans la reconnaissance de la parole.

2. Modèles de Markov cachés (HMM) :

Ces modèles ont été introduits par Baum-Welch à la fin des années 60 et utilisés à partir des années 80 en reconnaissance de la parole, le travail exposé dans [rabiner] est devenu une célèbre référence en la matière. Les HMMs peuvent-être décrits comme étant des automates probabilistes à états finis comportant deux processus : un processus caché et un processus d'émission. Le premier processus est dit caché car il est non observable, il concerne la suite (chaine) d'états visités alors que le second est observable car il concerne la suite d'observations émises par chaque état de la chaine cachée [9]. A chaque transition est associé un symbole (observation) d'un alphabet fini ou continu. Un HMM est définit donc par une structure composée d'états et de transitions et par un ensemble de distributions de probabilité sur les transitions.

L'objectif est de modéliser des séquences d'observations. Ces observations peuvent être de nature discrète (par exemples les caractères d'un alphabet fini) ou continue (température). Pour ce qui nous concerne les symboles émis seront des vecteurs spectraux (MFCC ou LPC), chaque vecteur étant composé de 39 valeurs réelles continues. Cette succession de vecteurs représente en réalité une succession d'unités acoustiques représentant la prononciation à reconnaitre.

2.1 Principes d'usage des HMMs en RAP :

Le système de reconnaissance considère le signal vocal comme étant une succession de symboles phonétiques représentant le message véhiculé par le dit signal vocal.

Il s'agit de reconnaître la séquence de symboles sous-jacente à partir d'un énoncé oral, le signal de parole est d'abord subdivisé en petites trames de 25 à 30 ms pendant lesquelles le signal est considéré comme stationnaire, chaque trame subit ensuite une opération de codage à travers laquelle on extrait les paramètres informatifs pertinents sous la forme d'un vecteur de coefficients. Notons que dans notre travail on utilise trois types de coefficients : MFCC, LPC et LPCEPSTRA. Le but recherché est d'établir une correspondance entre les séquences de vecteurs de parole et les séquences de symboles sous-jacentes souhaitées. En bref c'est le passage du signal au message, on va modéliser les mots ou les unités phonétiques par des modèles de Markov cachés selon qu'il s'agisse de reconnaitre des mots isolés ou de la parole continue. La figure (II.1) illustre le principe de la RAP.

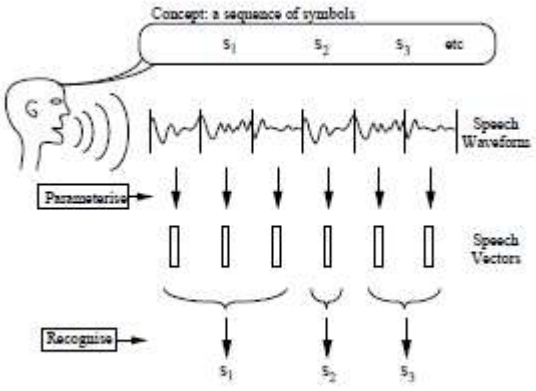


Figure (II.1) : le codage et le décodage d'un message

2.2 Topologies des HMM :

La topologie d'un HMM décrit le nombre d'états ainsi que les transitions permises entre ces états. La topologie choisie dans notre travail est celle illustrée sur la figure (II.2) et qui a donné les meilleurs résultats dans un travail antérieur sur la même base de données [pfe 2019-2020] . Une fois adoptée, la topologie d'un HMM n'est pas modifiée lors de l'apprentissage. Seuls les paramètres du HMM sont réajustés à l'aide des algorithmes classiques d'apprentissage.

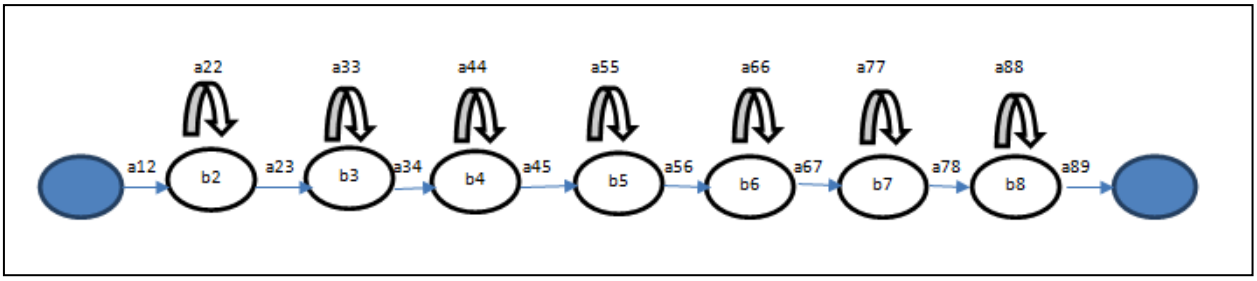


Figure (II. 2) : la topologie du HMM utilisée

Ce sont des modèles dits Gauche-Droite. Chaque état se connecte à lui-même (auto-transitions) et à l'état suivant. Les auto-transitions expliquent les variations de la durée des motifs dans un état [10], dans notre cas il exprime la durée d'une unité phonétique.

3. Reconnaissance des mots isolés :

Chaque mot est modélisé par un HMM, en d'autres termes on dispose d'un ensemble de HMMs faisant office de représentants des mots à reconnaitre. La reconnaissance revient alors à calculer la vraisemblance de la suite d'observations acoustiques constituant le mot à reconnaître par rapport à chacun des modèles. Le modèle présentant la plus grande vraisemblance d'avoir émis cette suite d'observations fournit le mot reconnu. L'algorithme permettant d'optimiser ce calcul est fondé sur la programmation dynamique, mais dans un cadre stochastique, l'algorithme de Viterbi [11]

Chaque mot prononcé doit être représenté par une séquence de vecteurs de parole ou d'observations O, définie comme suit :

$$\boldsymbol{O} = \boldsymbol{O_1}, \boldsymbol{O_2}, \ldots, \boldsymbol{O_T} \qquad \ldots (\boldsymbol{II.1})$$

Où " $\mathbf{O_t}$ "est le vecteur de parole observé au temps t. Le problème de reconnaissance des mots isolés peut alors être considéré comme étant le calcul de :

$$\boldsymbol{arg\,\max_i}\{\boldsymbol{P}(\boldsymbol{W_i}/\boldsymbol{O})\} \qquad \ldots \textit{(II.2)}$$

Où " $\boldsymbol{W_i}$ "est le i^{ème} mot du vocabulaire. Cette probabilité n'est pas directement calculable mais l'utilisation de la règle de Bayes donne :

$$\boldsymbol{P}(\boldsymbol{W_i}/\boldsymbol{O}) = \boldsymbol{P}(\boldsymbol{O}/\boldsymbol{W_i})\,\boldsymbol{P}(\boldsymbol{W_i})/\boldsymbol{P}(\boldsymbol{O}) \qquad \ldots \textit{(II.3)}$$

Ainsi, pour un ensemble donné de probabilités antérieures $\mathbf{P(W_i)}$, le mot prononcé le plus probable ne dépend que de la vraisemblance $\mathbf{P(W_i/O)}$.

Dans la reconnaissance vocale basée sur HMM, il est supposé que la séquence de vecteurs spectraux observés correspondant à chaque mot soit générée par un modèle de Markov comme indiqué sur la figure (II.3). Dans notre contexte, un modèle de Markov sera donc une machine à états finis qui change d'état une fois à chaque unité de temps et à chaque fois qu'un état j est visité, un vecteur de parole Ot est généré à partir de la densité de probabilité bj **(O_t).** De plus, la transition de l'état i à l'état j est également probabiliste et est régie par la probabilité discrète a_{ij}. La figure (II.3) montre un exemple de ce processus dans lequel la séquence d'états ayant généré notre mot est X = 1, 2, 2, 3, 4, 4, 5, 6. Le mot en question est**:** **$O=O_1O_2O_3O_4O_5O_6$**

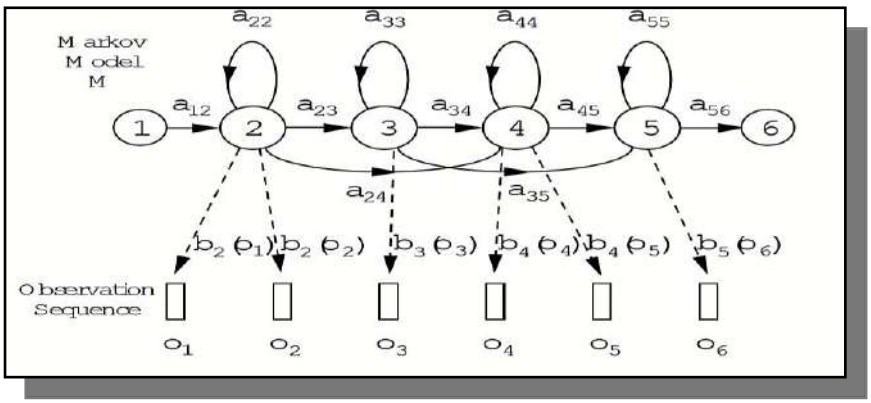


Figure (II.3) : Le modèle de génération de Markov

La probabilité conjointe que "O" soit généré par le modèle M se déplaçant dans la séquence d'états X est calculée simplement comme le produit des probabilités de transition et des probabilités de sortie. Donc, pour la séquence d'états X de la figure (II.3).

$$\boldsymbol{P(O,X/M) =}$$

$$\boldsymbol{b_2(O_1)a_{22}b_2(O_2)a_{23}b_3(O_3)a_{34}b_4(O_4)a_{44}b_4(O_5)a_{45}b_5(O_6)a_{56}} \quad \ldots (\boldsymbol{II.4})$$

Cependant, en pratique, seule la séquence d'observation O est connue et la séquence d'états sous-jacente X est masquée, c'est pourquoi on l'appelle un modèle de Markov caché.
Étant donné que X est inconnu, la vraisemblance requise est calculée en sommant toutes les séquences d'états possibles X = x (1), x (2), x (3), ..., x (T), c'est-à-dire

$$\boldsymbol{P(O/M) = \sum_{X} a_{X(0)X(1)} \prod_{t=1}^{T} b_{X(t)}(O_t) a_{X(t)X(t+1)}} \quad \ldots (\boldsymbol{II.5})$$

Où x(0) est contraint à être l'état d'entrée du modèle et x (T +1) est contraint à être l'état de sortie du modèle.

Comme alternative à l'équation II.14, la probabilité peut être approximée en considérant uniquement la séquence d'états la plus probable, c'est-à-dire

$$\boldsymbol{P(O/M) = \max_{x}\{a_{X(0)X(1)} \prod_{t=1}^{T} b_{X(t)}(O_t) a_{X(t)X(t+1)}\}} \quad \ldots (\boldsymbol{II.6})$$

Bien que le calcul direct des équations II.14 et II.15 ne soit pas traitable, il existe des procédures récursives simples qui permettent de calculer les deux quantités très efficacement. Avant d'aller plus loin, cependant, notez que si l'équation II.11 est calculable, le problème de reconnaissance est résolu. Étant donné un ensemble de modèles M_i correspondant aux mots W_i, l'équation II.11 est résolue en utilisant II.12 et en supposant que

$$\boldsymbol{P(W_i/O) = P(O/M_i)} \quad \ldots (\boldsymbol{II.7})$$

La figure (II.4) résume l'utilisation des HMM pour la reconnaissance de mots isolés. Tout d'abord, un HMM est formé pour chaque mot de vocabulaire en utilisant un certain nombre d'exemples de ce mot. Dans cet exemple, le vocabulaire est celui utilisé dans notre travail. Deuxièmement, pour reconnaître un mot inconnu, la probabilité que chaque modèle génère ce mot est calculée et le modèle le plus probable identifie le mot.

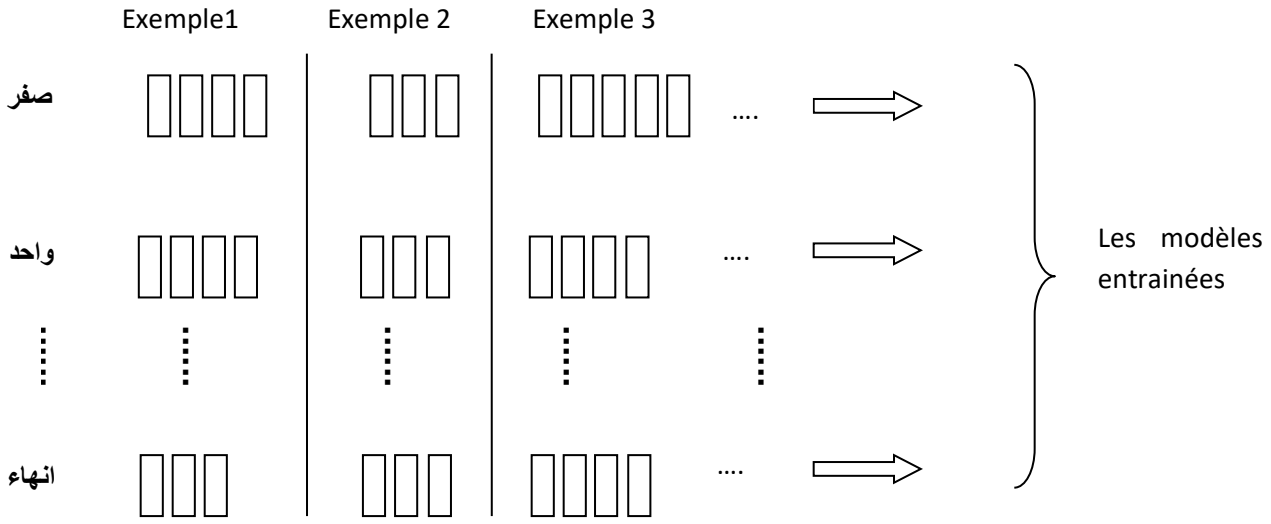


Figure (II.4) : utilisation des HMMs dans la reconnaissance des mots isolés.

3.1 Les paramètres du HMM continu :

Un HMM est constitué d'un certain nombre d'états. Chaque état j est associé à une distribution de probabilité d'observation **bj(ot)** qui détermine la probabilité de générer l'observation Ot à l'instant t et chaque paire d'états i et j a une probabilité de transition associée aij

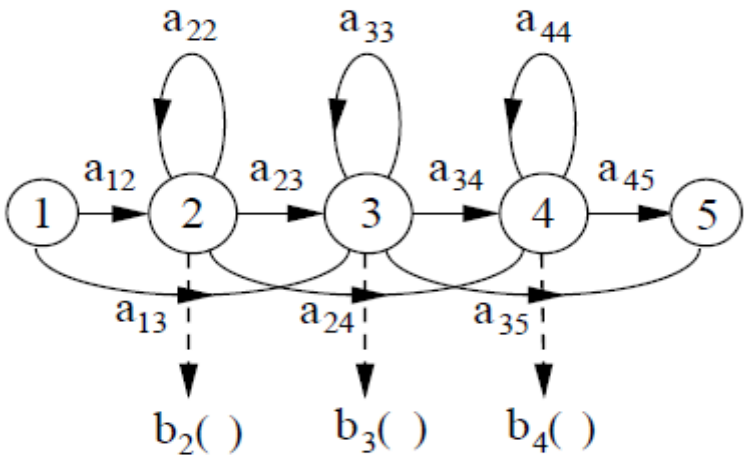


Figure (II.5) : simple gauche- droite HMM

La figure (II.5) montre un simple HMM gauche-droite avec cinq états au total. Trois d'entre eux sont des états émetteurs et ont des distributions de probabilité de sortie qui leur sont associées. La matrice de transition de ce modèle comporte 5 lignes et 5 colonnes. La

somme de chaque ligne est égale à un, sauf pour la dernière ligne qui est toujours égale à zéro puisqu'aucune transition n'est autorisée à partir de l'état final.12

HTK s'intéresse principalement aux modèles à densité continue dans lesquels la distribution de probabilité de chaque observation est représentée par une densité gaussienne de mélange. Dans ce cas, pour l'état j, la probabilité **bj(ot)** de générer l'observation Ot est donnée par :

$$\boldsymbol{b_j(o_t)} = \prod_{s=1}^{S} \left[\sum_{m=1}^{M_{js}} \boldsymbol{C_{jsm}} \mathcal{N}\left(\boldsymbol{o_{st}}; \boldsymbol{\mu_{jsm}}, \sum_{jsm}\right) \right]^{\gamma s} \quad \ldots (\boldsymbol{II.8})$$

Où **Mjs** est le nombre de composants du mélange dans l'état j pour le flux s, cjsm est le poids de la m^{ème} composante et N est une gaussienne multi variée avec un vecteur moyen μ et une matrice de covariance Σ , c'est-à-dire que :

$$\mathcal{N}(\boldsymbol{o}; \boldsymbol{\mu}, \sum) = \frac{1}{\sqrt{(2\pi)^n |\sum|}} e^{-\left(\frac{1}{2}\right)(o-\mu)' \sum^{-1} (o-\mu)} \quad \ldots (\boldsymbol{II.9})$$

Où n est la dimensionnalité de l'observation (39 dans notre cas) [13]

4. Les trois Problèmes en HMMs :

La manipulation des HMMs soulève trois problèmes principaux :

- **Calcul de la vraisemblance** : C'est l'évaluation de la probabilité de l'observation d'une séquence. Etant donnée une suite d'observations **O = {O1, O2, O3,..., Ot}** et un modèle λ = (A, B, π), on cherche à évaluer la probabilité d'apparition de cette séquence connaissant le modèle. Pour résoudre ce problème on utilise l'algorithme Forward-Backward.
- **Le problème d'apprentissage :** qui cherche à ajuster les paramètres du modèle **λ= (A,B,π),** pour maximiser P(O/λ), à partir d'un ensemble de séquences d'apprentissages **O = {O1,O2, O3,..., Ot}** qui ont été émises par ce modèle. L'algorithme de Baum Welch permet de résoudre ce problème.

- **Le problème de la reconnaissance :** C'est la recherche du chemin le plus probable, ou l'estimation de la partie cachée. Etant donnée une suite d'observation

O= {O_1, O_2, O_3... O_t} et un modèle λ = (A, B, π), on cherche à trouver la suite d'états, appartenant à X = {x_1, x_2, x_3,..., x_T}, qui explique le mieux l'observation. Pour ce faire, on fait appel à l'algorithme de Viterbi qui permet de montrer la séquence d'états la plus probable produisant cette séquence d'évènements.

5. Calcul de la vraisemblance d'une séquence :

Une séquence donnée peut en général être engendrée de plusieurs façons distinctes par un HMM. La vraisemblance d'une séquence O peut s'exprimer sous la forme suivante :

$$\boldsymbol{P\,(O(t)/\,\lambda)\;=\;\Sigma\,P\,(O(t),q(t)/\,\lambda)} \qquad \textbf{...(II.10)}$$

La somme est faite pour toutes les séquences possibles d'états q(t). Il s'agit là d'un calcul direct de la probabilité P (O(t)/ λ). Mais cette façon de procéder nous mène à des calculs d'ordre exponentiel (T x N^T), qui sont très coûteux en temps de calcul.

Par la suite, nous décrivons un procédé récursif plus astucieux et plus rapide qui permet réduire le temps de calcul de la probabilité II.17. Il s'agit de l'algorithme Forward-Backward ou progressif-rétrogressif **[6].** La vraisemblance de l'observation est évaluée par :

$$\boldsymbol{P\,(O/\lambda)\;=\;\sum_{i=1}^{n}\alpha_t(i)\beta_t\,(i)} \qquad \textbf{...(II.11)}$$

Où $\boldsymbol{\alpha_t(i)}$ est la probabilité d'émettre la suite $O_{(1\,:\,t)}$ et d'être à q_i à l'instant t, et $\beta_t(i)$ est la probabilité d'émettre la suite $O_{(t+1\,:\,T)}$ sachant que l'on part de q_i à l'instant t, le calcul de α se fait avec t croissant tandis que le calcule de β se fais avec t décroissant, d'où l'appellation Frward-Backard. [14]. La probabilité d'observation d'une séquence est obtenue en prenant les valeurs de α et β à un instant t quelconque par l'équation II.18. Cependant, on utilise le plus souvent les valeurs obtenues pour deux cas particulier (t=0) ou (t=T).

6. L'apprentissage :

Il s'agit de modifier les paramètres d'un HMM à partir d'un certain nombre d'exemples pour que le modèle représente de manière fiable cette classe d'exemples. Les formules de réestimation sont :

$$\widehat{a}_{ij} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r-1} \alpha_i^r(t) a_{ij} b_j(O_{t+1}^r) \beta_j^r(t+1)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T_r} \alpha_i^r(t) \beta_j^r(t)} \quad \ldots (II-12)$$

$$\widehat{a}_{1j} = \frac{1}{R} \sum_{r=1}^{R} \frac{1}{P} \alpha_j^r(1) \beta_j^r(1) \quad \ldots (II-13)$$

$$\widehat{a}_{iN} = \frac{\sum_{r=1}^{R} \frac{1}{P_r} \alpha_i^r(T) \beta_i^r(T)}{\sum_{r=1}^{R} \frac{1}{P_r} \sum_{t=1}^{T} \alpha_i^r(t) \beta_i^r(t)} \quad \ldots (II-14)$$

Pour les gaussiennes simples, il est plus efficace dans ce cas d'utiliser **:**

$l_j^r(t) = \frac{1}{p_r} \alpha_j(t) \beta_j(t)$ Où $Pr = P(O/M)$.

Les formules de ré-estimation peuvent maintenant être exprimées en termes de $l_j^r(t)$ **:**

$$\widehat{\mu}_J = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} l_j^r(t) o_t^r}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} l_j^r(t)} \quad \ldots (II-15)$$

$$\widehat{\sum}_j = \frac{\sum_{r=1}^{R} \sum_{t=1}^{T_r} l_j^r \left(o_t^r - \widehat{\mu}_j\right)\left(o_s^t - \widehat{\mu}_j\right)'}{\sum_{r=1}^{R} \sum_{t=1}^{T_r} l_j^r(t)} \quad \ldots (II-16)$$

Avec :

R : nombre d'exemples utilisés pour chaque mot.

Tr : longueur du mot en termes de nombre de vecteurs spectraux.

$\hat{\mu}_j$:Moyenne estimée de la gaussienne utilisée pour modéliser la distribution d'émission de l'état j.

$\widehat{\sum}_j$:Matrice de covariance de la gaussienne utilisée pour modéliser la distribution d'émission de l'état j.

7. La reconnaissance :

A partir d'une séquence d'observation représentant un mot à reconnaitre, on cherche le modèle qui donne le maximum de probabilité de génération de la séquence d'observation. L'algorithme utilisé est l'algorithme de Viterbi. Sa particularité réside dans le fait que, dans le calcul de la vraisemblance, au lieu de considérer toutes les séquences d'états possibles, on garde seulement la séquence d'états donnant la plus grande probabilité. Tous les algorithmes précédemment abordés sont implémentés dans la plateforme HTK que nous utiliserons pour manipuler les modèles de Markov cachés.

8. Conclusion :

Ce chapitre avait pour objet d'introduire le formalisme mathématique des modèles de Markov cachés et leur utilisation en RAP. Cette utilisation s'appuie généralement sur une hypothèse simplificatrice qui consiste à considérer la matrice de covariance de la gaussienne comme étant diagonale. Cette hypothèse simplificatrice réduit considérablement le volume de calcul requis, mais soulève des questions quant à sa validité vue la dégradation des performances du système de reconnaissance qui en résulte. C'est dans cette optique que s'inscrit le prochain chapitre dans lequel nous tentons de répondre à la question suivante : à quel degré la perte de performance de reconnaissance est-elle tolérée face à la réduction du volume de calcul à travers l'hypothèse simplificatrice d'une matrice de covariance diagonale.

CHAPITRE III

Résultats et Interprétations

1. Introduction :

Dans ce chapitre nous allons mener une série d'expérimentations afin de déterminer l'effet de la forme de la matrice de covariance sur les performances d'un système de reconnaissance de la parole à base de modèles de Markov cachés. La chronologie adoptée dans ce chapitre est la suivante :

- Présentation de la base de données utilisée
- Description des principaux outils HTK utilisés pour manipuler les modèles de Markov cachés.
- Réalisation du système de reconnaissance par HTK.
- Etude de performance en fonction de la matrice de covariance choisie
- Conclusions

1. Description de la base de données utilisée :

La base de données est constituée de 20 mots isolés prononcés en langue arabe, chaque mot est prononcé 10 fois par un ensemble de 50 locuteurs arabes natifs. Le nombre total de prononciations est de 9991 au lieu de 10000 étant donné que 9 fichiers étaient corrompus. Le corpus a été enregistré avec une fréquence d'échantillonnage de 44100 Hz, les échantillons sont sous forme d'entiers de 16 bits. Il a été convenu de diviser la base de données en deux parties : corpus d'apprentissage (20%) de la base et corpus de test (80%) de la base. [15]

2.1 Vocabulaire : Les mots constituant le vocabulaire du système sont :

1. صفر(zero)
2. واحد (one)
3. اثنان (two)
4. ثلاثة (three)
5. أربعة (four)
6. خمسة (five)
7. ستة (six)
8. سبعة (seven)
9. ثمانية (eight)
10. تسعة (nine)
11. التنشيط (activation)
12. التحويل (transfert)
13. الرصيد (balance)
14. التسديد (payment)
15. نعم (yes)
16. لا (no)
17. التمويل (funding)
18. البيانات (data)
19. الحساب (account)
20. انهاء (end)

2.2 Système d'étiquetage des fichiers de la base :

Les fichiers ont été étiquetés en utilisant la notation suivante :

S (numéro de locuteur). (Nombre de répétition). (Numéro du mot)

Exemple :

S01.01.01.wav représente le fichier correspondant au premier locuteur (parmi 50) prononçant pour la première foi (parmi 10) le premier mot (parmi 20).

2. Description de l'outil de reconnaissance HTK :

L'outil « HTK »(Hidden Markov Model Toolkit) est un ensemble de routines permettant la créationet la manipulation des modèles de Markov cachés.Il est principalement utilisé dans la recherche sur la reconnaissance vocale mais il est tout à fait utilisable dans nombreuses autres applications telles que la synthèse vocale, la reconnaissance de l'écriture ou la reconnaissance de séquences d'ADN (acide désoxyribonucléique)[16].Originalement développé à l'université de Cambridge,HTK consiste en un ensemble de modules de bibliothèque et d'outils disponibles sous langage source C. HTK prend en charge les HMMs à observations continues à densité de distribution avec mélange de gaussiennes ainsi que les HMMs discrets et peut être utilisé pour créer des systèmes HMM complexes[17]. L'outil en question vient avec une documentation complète et des exemples d'illustration.

3. Réalisation du système de reconnaissance par HTK :

La réalisation d'un système de reconnaissance nécessite l'achèvement des étapes suivantes : le codage, le choix de la topologie des modèles utilisés, l'initialisation des modèles, l'apprentissage, la définition de la grammaire et enfin le test de reconnaissance.

4.1 Le codage : cette étape consiste à extraire les paramètres spectraux à partir du signal vocal, différents types de paramètres peuvent être extraits à savoir : MFCC, LPC et autres. La figure(III.1) illustre la procédure de codage d'un seul fichier (une seule prononciation), chaque trame d'une vingtaine de millisecondes du signal parole est pondérée par une fenêtre de Hamming avant d'en extraire les paramètres informatifs pertinents (coefficients MFCC ou LPC), on obtient ainsi une suite de vecteurs de paramètres pour chaque prononciation. Le codage peut être réalisé sur un groupe de fichiers à la fois au lieu d'un seul fichier, ceci est illustré sur la figure (III.2) il suffit pour cela de spécifier, dans un script, la liste des noms des fichiers à traiter (fichiers sources) et la liste des noms des fichiers après le codage (fichiers cibles). Le codage

est réalisé en invoquant la fonction **HCopy. C**'est unefonction à usage général pour copier, coder et manipuler des fichiers de parole, cette fonction est invoquée avec la syntaxe suivante :

Syntaxe : HCopy –A –D –C config.txt –S script.txt

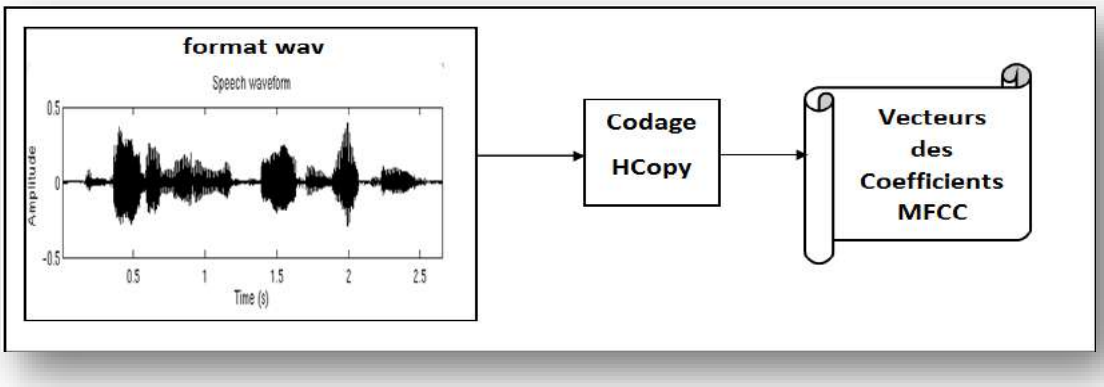


Figure (III. 1) : procédure de codage

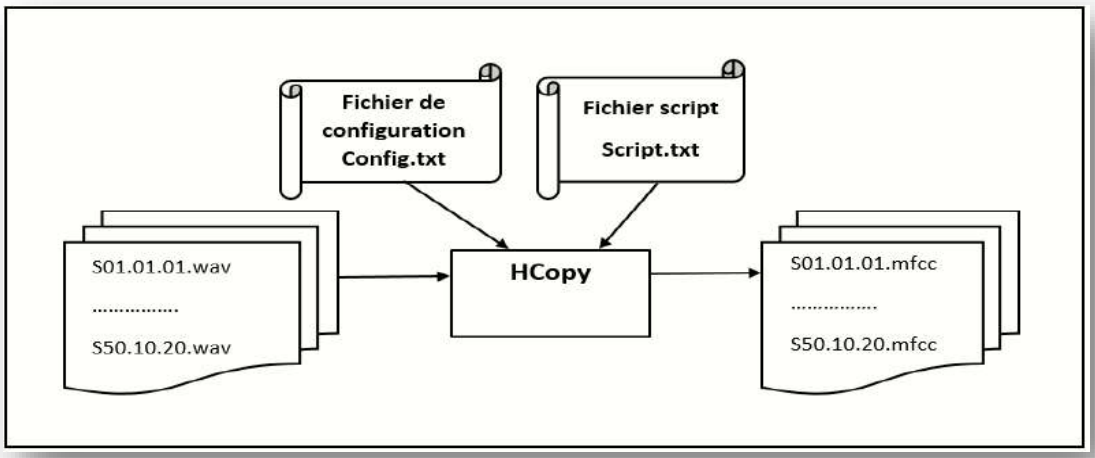


Figure (III. 2) : utilisation de HCopy pour un groupe de fichiers

```
SOURCEKIND = WAVEFORM
SOURCEFORMAT = NOHEAD
SOURCERATE = 113
TARGETKIND = MFCC_0_D_A
WINDOWSIZE = 250000.0
TARGETRATE = 100000.0
NUMCEPS = 12
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
```

```
# Identifier of the coefficients to use
   # = 25 ms = length of a time frame
   # = 10 ms = frame periodicity
         # Number of MFCC coeffs (here from c1 to c12)
      # Use of Hamming function for windowing frames
    # Pre-emphasis coefficient
      # Number of filterbank channels
        # Length of cepstral liftering
```

Figure (III. 3) : fichier de configuration

Dans le fichier de configuration, illustré dans la figure (III.3) on peut lire les informations suivantes :

- SOURCEKIND : type de fichier source (ici WAVEFORM)
- SOURCEFORMAT : format du fichier source (ici NOHEAD : sans entète)
- TARGETRATE : 113 (période d'échantillonnage en unités de 100 ns)
- TARGETKIND : type du fichier cible (ici MFCC_0_D_A)
- WINDOWSIZE : largeur de trame (en unités de 100 ns)
- TARGETRATE : périodicité des trames (en unités de 100 ns)
- NUMCEPS : nombre de coefficients MFCC utilisés
- USEHAMMING = T fenêtre utilisée de type Hamming(T pour dire True)
- PREEMCOEF : coefficient de préaccentuation
- NUMCHANS : nombre des canaux du banc de filtre utilisé dans le calcul des MFCCs
- CEPLIFTER : utilisé dans le calcul de la IDCT
- Le symbole # est utilisé pour les commentaires.

Chaque vecteur MFCC contient 39 coefficients (13 coefficients : C1 à C12 plus le coefficient C0, à ces 13 coefficients seront annexés les coefficients dérivés d'ordre 1 et 2).

- **Script.txt** est le fichier texte spécifiant le nom et la localisation de chaque prononciation à transformer (fichier signal à traiter), avec le nom et la localisation des fichiers de coefficients des cibles.

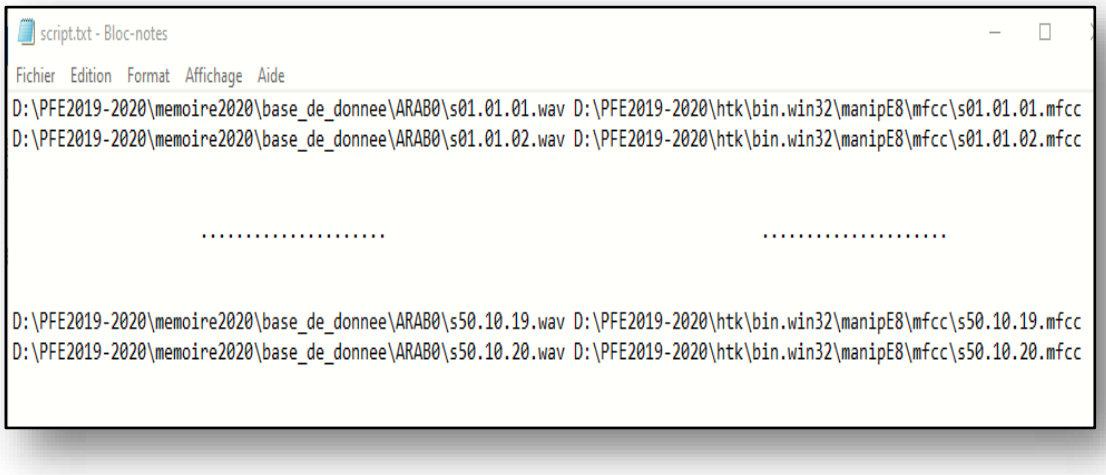


Figure (III. 4) : représente un exemple d'un script

4.2 la topologie du HMM choisie :

La topologie adoptée est celle qui a donné les meilleurs résultats pour cette base de données d'après [6].Cette topologie est représentée par un modèle gauche-droite (figure (III.5)) :

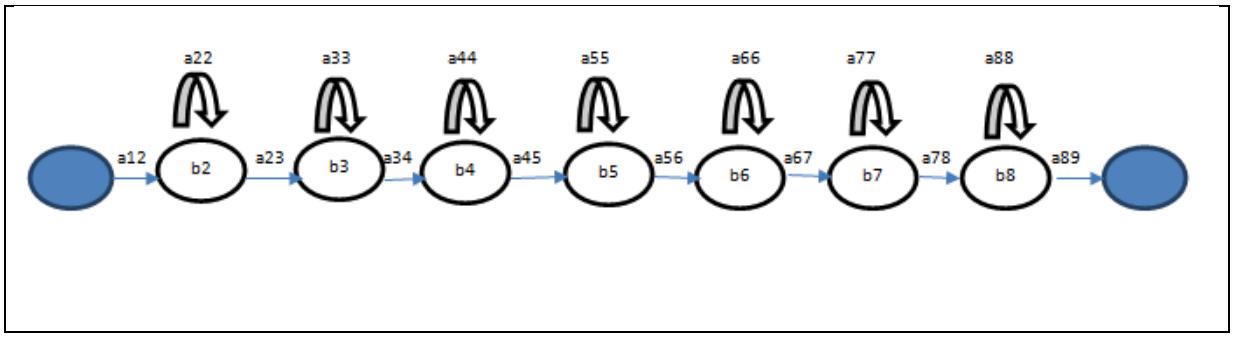
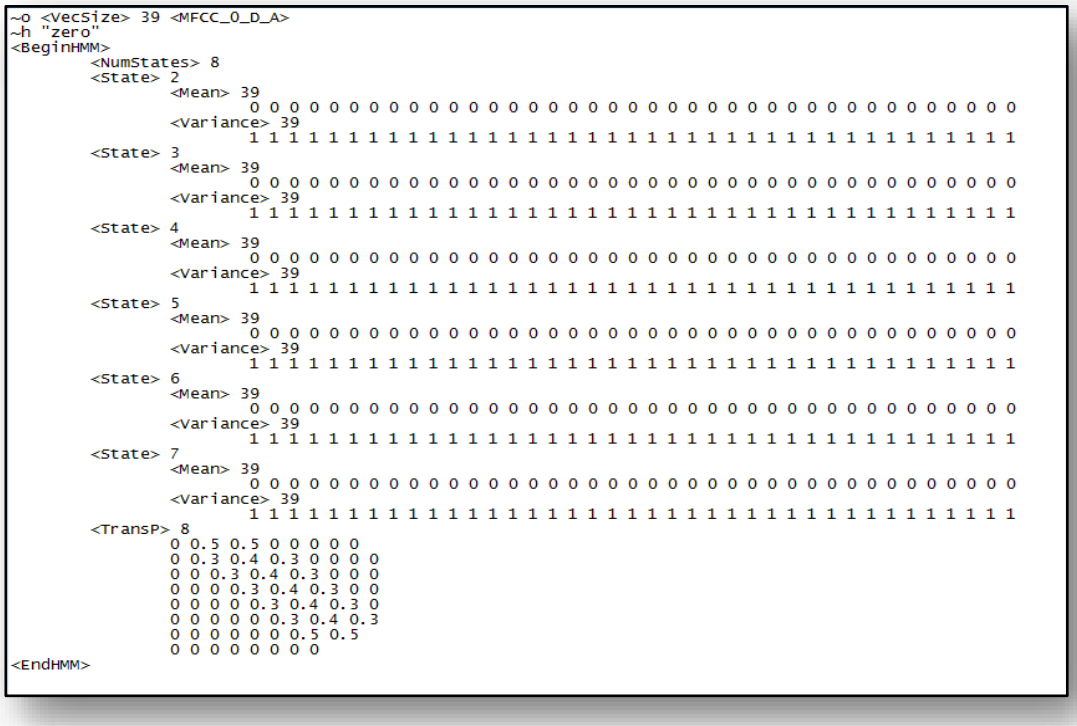


Figure (III.5) : la topologie choisie pour les HMMS

- Un nombre d'états égale à 9 dont 7 états émetteurs. Les probabilités de transition sont données par la matrice de probabilité de transition.
- Chaque état émetteur est caractérisé par une loi d'émission continue à une ou plusieurs gaussiennes caractérisées par un vecteur moyen μ (39composantes) et une matrice de covariance ∑.

La figure (III.6) représente un exemple du fichier de description du HMM, Ce Dernier est un fichier texte qui comporte tous les éléments du HMM. Les éléments du fichier de description HMM sont les suivants :

- **~o <VecSize> 39 <MFCC_0_D_A> :** est l'entête du fichier, donnant la taille du vecteur coefficient (ici 39 coefficients), et le types de coefficient ici (MFCC-0-D-A).
- **~h "zero" <BeginHMM>…<EndHMM> :** nom du HMM (ici zéro), et commencer et finir la description d'un HMM.
- **<NumStates>9** donne le nombre total d'états dans le HMM incluant les deux états (1et 9).
- Chaque état (sauf le premier et le dernier) possède une loi de probabilité d'émission. On s'est limité à un exemple représentant le cas continu pour ne pas alourdir le document, il s'agit en l'occurrence d'une seule gaussienne à un vecteur de moyenne **<Mean> 39**et une matrice de covariance à diagonale non nulle **<Variance> 39**



```
~o <VecSize> 39 <MFCC_0_D_A>
~h "zero"
<BeginHMM>
        <NumStates> 8
        <State> 2
                <Mean> 39
                        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                <Variance> 39
                        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        <State> 3
                <Mean> 39
                        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                <Variance> 39
                        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        <State> 4
                <Mean> 39
                        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                <Variance> 39
                        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        <State> 5
                <Mean> 39
                        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                <Variance> 39
                        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        <State> 6
                <Mean> 39
                        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                <Variance> 39
                        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        <State> 7
                <Mean> 39
                        0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
                <Variance> 39
                        1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
        <TransP> 8
                0 0.5 0.5 0 0 0 0 0
                0 0.3 0.4 0.3 0 0 0 0
                0 0 0.3 0.4 0.3 0 0 0
                0 0 0 0.3 0.4 0.3 0 0
                0 0 0 0 0.3 0.4 0.3 0
                0 0 0 0 0 0.3 0.4 0.3
                0 0 0 0 0 0 0.5 0.5
                0 0 0 0 0 0 0 0
<EndHMM>
```

Figure (III.6) : Fichier de description HMM

- **<TransP>9**donne la transition 9×9 du HMM, où a_{ij} est la probabilité de transition de l'état i à l'état j. les valeurs nulles indiquent que les transitions correspondantes ne sont pas autorisées. Les autres valeurs sont arbitrairement initialisées (la somme sur chaque ligne de la matrice doit être égale à 1), les valeurs seront modifiées plus tard au cours du processus d'apprentissage.

4.3 Initialisationdes modèles :

Avant de commencer le processus d'apprentissage les paramètres HMM doivent être initialisés correctement avec les données d'apprentissage. Afin d'obtenir une convergence rapide de l'algorithme d'apprentissage. HTK propose les outils d'initialisation : **HCompv** pour le cas continu et **Hinit** pour le cas discret

HCompv doit être invoquée comme suit :

HCompv -S manip/aprent.txt -M manip/5hmm0 -H manip/prototypes5/zero.txt zero

Hinit est invoquée de manière similaire, la seule différence c'est que cette fonction ne traite pas l'initialisation des moyennes et variances du cas continu gaussien

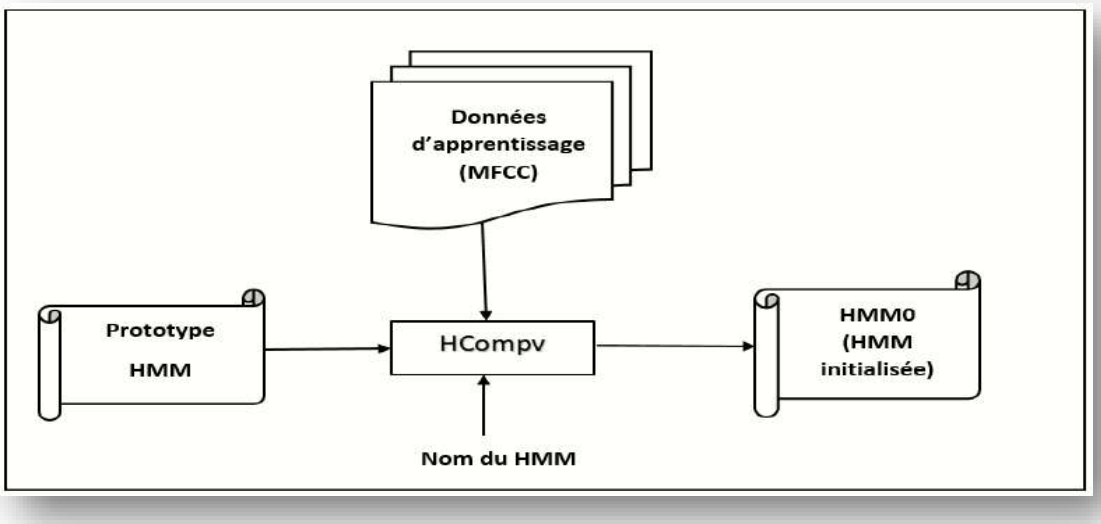


Figure (III.7) : fonctionnement de HCompv

L'exemple d'initialisation des modèles est illustré dans la figure (III.6).

- **manip/aprent.txt :** représente la liste complète des fichiers MFCC formant le corpus d'apprentissage pour chacun des 20 mots de la base de données.
- **HMM0 :** représente le nom du dossier qui contient les descriptions des HMMs (résultats) après l'initialisation seront placées.
- **manip/prototypes5/zero.txt :** représente le chemin (manip) et le fichier de description contenant les prototypes HMMs (des 20mots).
- **Zero :** est le nom du HMM à initialiser (zero,one…end).

4.4 Apprentissage :

La réestimation des modèles est faite avec l'outil **HRest** comme illustré dans la figure (III.7) utilisant l'algorithme de **Baum-welch**, qui calcule les valeurs optimales pour les paramètres HMM (probabilités de transition, plus les vecteurs de moyenne et de variance pour chacune des fonctions d'observation)

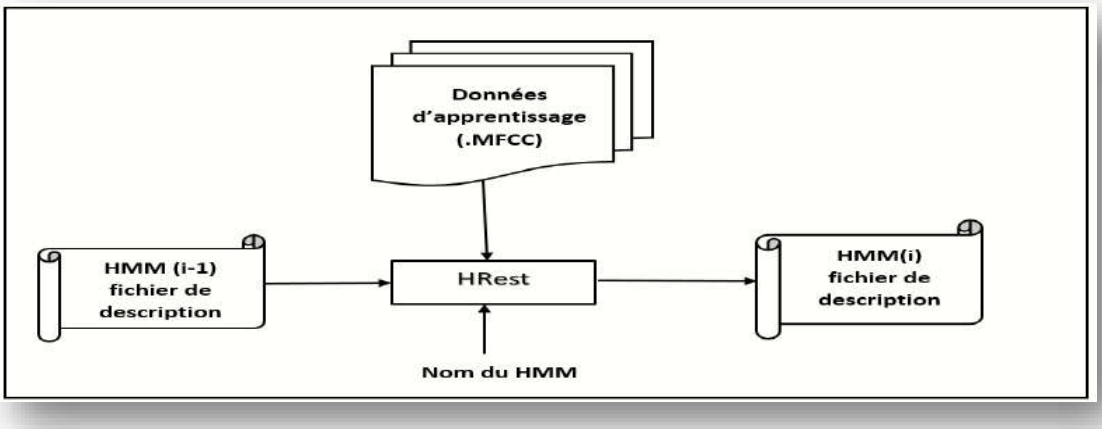


Figure (III.8) : fonctionnement de HRest

La ligne de commande suivante permet une itération de réestimation des modèles :

HRest-S manip/script/aprent01.txt-M manip/HMM0 -H manip/prototypes8/zero.txt zero

Avec:

- **HMM0 :** représente le répertoire de sortie, indique l'indice de l'itération courante i.
- **manip/prototypes8/zero.txt :** est le fichier de description de HMM appelé names of HMMs il est à stocker dans un répertoire dont le nom indique l'indice de la dernière itération (ici c'est manip/HMM0)
- **zero :** représente le nom du HMM à apprendre (entraîner).

Finalement on dispose maintenant les 20modèles entrainés qui sont prêts à être utilisés pour la reconnaissance.

4.5 Définition de la grammaire et du dictionnaire :

Avant d'utiliser nos modèles de mots entrainés, on doit définir la grammaire du système. En HTK, la grammaire est décrite dans un fichier texte selon certaines règles syntaxiques. [15] Dans notre cas la grammaire est assez simple, elle décrit un ensemble de 20 mots isolés. Le système traite chaque prononciation comme étant représentative d'un seul mot. Le résultat de reconnaissance sera l'un des 20 mots possibles appartenant au vocabulaire. Un exemple de fichier de la grammaire est illustré dans la figure (III.8) :

```
$word=zero|one|two|three|four|five|six|seven|eight|nine|activatio
n|transfert|balance|payement|yes|no|funding|data|account|end;
($word)
```

Figure (III.9) : le fichier de la grammaire

La variable **Word** peut être remplacée par l'un des 20 mots du vocabulaire (zéro one….. end). Le système doit savoir bien sûr à quel HMM correspond chacune des variables de grammaire zero, one, … end. Cette information est stockée dans un fichier texte appelé dictionnaire. Dans une telle tâche simple, la correspondance est simple, et la tache dictionnaire renferme tout simplement les 20 entrées. Comme illustre dans cette figure :

```
zero        zero
one         one
two         two
three       three
four        four
five        five
six         six
seven       seven
eight       eight
nine        nine
activation  activation
transfert   transfert
balance     balance
payement    payement
yes         yes
no          no
funding     funding
data        data
account     account
end         end
```

Figure (III.10) : le fichier du dictionnaire

Les éléments gauches se référant aux noms des variables de grammaire. Les éléments droits se référant aux noms des HMMs (introduit par ~h dans les fichiers de description HMM).Pour compléter la définition des taches on a besoin de créer un réseau de connexion reliant les 20 mots (comme illustré dans la figure suivante) par la compilation du fichier grammaire avec l'outil HParse :

Syntaxe :

HParse gram.txt reseau.slf

Tel que :

- **gram.txt :** est le fichier grammaire.
- **Reseau.slf :** est le fichier obtenu qui représente le réseau.

Nous avons à présent en notre possession tout ce qu'il nous faut pour faire la reconnaissance .La figure (III.10) représente le réseau de connexion entre les mots.

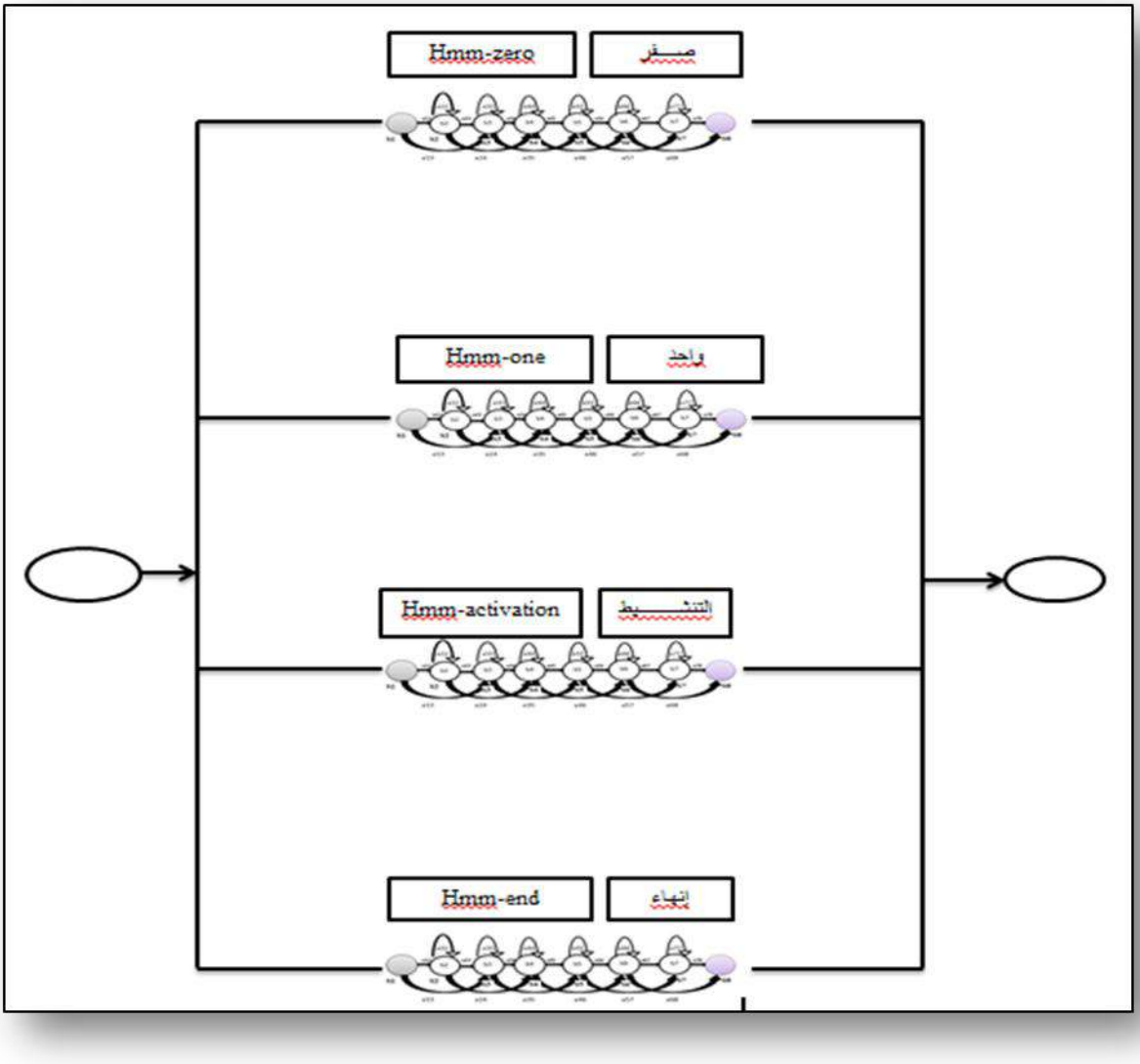


Figure (III.11) : le réseau de connexion entre mot

4.6 La reconnaissance :

Une fois les modèles HMMs entrainés obtenus, on peut procéder à l'étape de reconnaissance. La fonction utilisée est basée sur l'algorithme de Viterbi. Cette fonction est invoquée en HTk comme suit :

HVite -S test.txt -H Hmmdef -i reco.mlf -w reseau.slf dict.txt hmmlist.txt

Avec:

- **Hmmdef:** fichier texte contient la définition des HMMs.il est obtenu en copiant chaque définition des HMMs (de la dernière itération) dans un fichier unique. Chaque élément est séparé par une nouvelle ligne de l'autre élément, et une nouvelle ligne après le dernier élément.
- **Reco.mlf : c'est** le fichier de sortie (réseau de connexion obtenu).
- **Hmmlist :** fichier texte contenant les noms des 20 modèles à utiliser, les éléments sont séparés par une nouvelle ligne.
- **Test.txt :** le fichier script qui contient la liste des fichiers (MFCC) de teste

La sortie est stockée dans un fichier **reco.mlf** qui contient le fichier de reconnaissance avec la vraisemblance de reconnaissance pour chaque mot. La figure (III.11) illustre l'utilisation de Hvite.

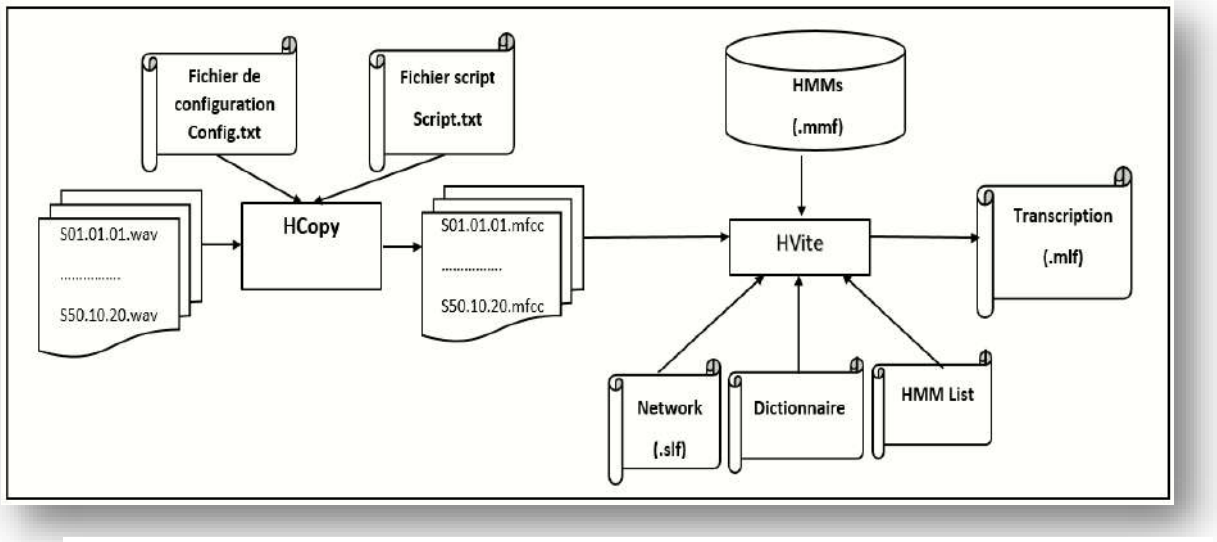
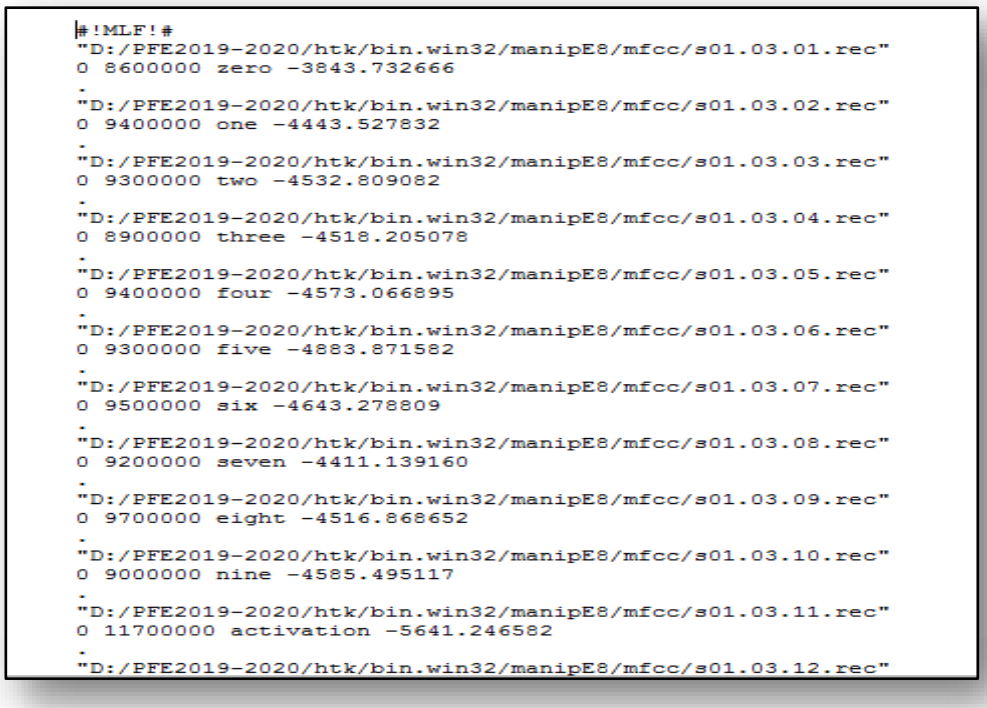


Figure (III.12) : processus de codage et reconnaissance

La figure ci-dessous illustre un extrait de fichier du résultat de transcription :



```
#!MLF!#
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.01.rec"
0 8600000 zero -3843.732666
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.02.rec"
0 9400000 one -4443.527832
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.03.rec"
0 9300000 two -4532.809082
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.04.rec"
0 8900000 three -4518.205078
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.05.rec"
0 9400000 four -4573.066895
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.06.rec"
0 9300000 five -4883.871582
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.07.rec"
0 9500000 six -4643.278809
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.08.rec"
0 9200000 seven -4411.139160
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.09.rec"
0 9700000 eight -4516.868652
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.10.rec"
0 9000000 nine -4585.495117
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.11.rec"
0 11700000 activation -5641.246582
.
"D:/PFE2019-2020/htk/bin.win32/manipE8/mfcc/s01.03.12.rec"
```

Figure (III.13) : Exemple d'un fichier reco

4. **Expérimentation :** étant donné qu'on cherche à comparer entre les performances des HMMs à matrice de covariance diagonale et celles des HMMs à matrice de covariance pleine on a opté pour la stratégie suivante :
   - Etude des performances des HMMs à matrice de covariance diagonale et des HMMs à matrice de covariance pleine en prenant les coefficients MFCC comme type de vecteur de paramètres.
   - Etude des performances des HMMs à matrice de covariance diagonale et des HMMs à matrice de covariance pleine en prenant les coefficients LPC comme type de vecteur de paramètres.
   - Etude des performances des HMMs à matrice de covariance diagonale et des HMMs à matrice de covariance pleine en prenant les coefficients LPCEPSTRA comme type de vecteur de paramètres.

5.1 Performances de reconnaissance obtenues pour les MFCC :

Le tableaux (III.1) donne le taux de reconnaissance obtenu pour différents niveaux de bruit dans les deux cas de matrice de covariance : pleine et diagonale

	Sans bruit (clean)	0 dB	5dB	10dB	15dB
Matrice covariance diagonale	92.36	87.73	90.02	91.19	91.79
Matrice covariance pleine	93.43	90.67	92.17	92.17	92.82

Tableau(III.1): Taux de reconnaissance dans le cas MFCC

Interprétations :

- La première constatation évidente est l'amélioration du taux de reconnaissance lorsque le rapport signal sur bruit (RSB) augmente et ce indépendamment de la forme de la matrice de covariance.
- On observe quand même que le taux de reconnaissance est meilleur dans le cas d'une matrice de covariance pleine par rapport au cas d'une matrice de covariance diagonale.
- Le point important à soulever est le suivant : cette amélioration obtenue justifie-t-elle l'utilisation d'une matrice de covariance pleine quand on sait que cela entraine une complexité de calcul beaucoup plus importante ? il importe dans ce contexte de rappeler que les vecteurs de paramètres utilisés ont une taille de 39, ce qui donne une matrice de covariance de taille 39*39 (1521 valeurs) pour chaque gaussienne utilisée. L'utilisation d'une matrice pleine implique donc un volume de calcul exorbitant. Si on utilise une matrice diagonale, seuls les éléments de la diagonale seront prix en compte (39 valeurs), ce qui suppose que les coefficients utilisés sont décorrélés. Ce cas de figure diminue largement la complexité de calcul et c'est pour cela que cette hypothèse est toujours adoptée lorsqu'on utilise les coefficients MFCC.

5.2 Performances de reconnaissance obtenues pour les LPC :

Le tableaux (III.2) donne le taux de reconnaissance obtenu pour différents niveaux de bruit dans les deux cas de matrice de covariance : pleine et diagonale

	Sans bruit (clean)	0 dB	5Db	10dB	15Db
Matrice covariance diagonale :	60.64	56.65	69.93	71.77	72.64
Matrice covariance pleine :	86.39	52.96	65.00	75.46	83.05

Tableau(III.2): Taux de reconnaissance dans le cas LPC

Interprétations :

- La première constatation est la baisse de performances de reconnaissance par rapport au cas MFCC.
- La deuxième constatation est que, dans le cas des LPC et pour les niveaux faibles de bruit, l'utilisation d'une matrice de covariance pleine apporte une amélioration nettement significative par rapport au cas d'une matrice diagonale, ce qui n'était pas vraiment visible dans le cas des MFCC. L'explication évidente est que les coefficient LPC ne sont pas décorrélés, donc les éléments de la matrice de covariance qui sont hors de la diagonale ne sont pas nuls. L'adoption dans ce cas de l'hypothèse d'une matrice de covariance diagonale va conduire à une dégradation de performances de reconnaissance. Une telle hypothèse n'est donc pas justifiée dans le cas des LPC.

5.3 Performances de reconnaissance obtenues pour les LPCEPSTRA :

Le tableaux (III.3) donne le taux de reconnaissance obtenu pour différents niveaux de bruit dans les deux cas de matrice de covariance : pleine et diagonale

	Sans bruit (clean)	0 dB	5dB	10dB	15dB
Matrice covariance diagonale :	81.89	51.29	69.94	74.28	80.41
Matrice covariance pleine :	89.10	55.09	68.91	77.37	82.67

Tableau(III.3): Taux de reconnaissance dans le cas LPCEPSTRA

Interprétations :

- Les résultats obtenus montrent des performances inférieures à celles du cas MFCC mais quand même meilleures que celles du cas LPC

- La différence n'est pas nette entre le cas d'une matrice de covariance pleine et celui d'une matrice de covariance diagonale, ce qui n'était pas le cas pour les LPCs, ceci est due au fait que le calcul des coefficients LPCEPSTRA dans sa dernière phase est similaire à celui des MFCCs, et donc les coefficients obtenus sont décorrélés, conduisant à une matrice de covariance diagonale

6. Conclusion :

Ce chapitre a été entièrement consacré à la réalisation d'un système de reconnaissance à base de modèles de Markov cachés (HMMs). L'objectif était de vérifier la validité de l'une des hypothèses les plus adoptées dans les HMMs à savoir l'utilisation d'une matrice de covariance diagonale pour les gaussiennes des modèles dans le but de réduire la complexité des calculs dans les deux phases : apprentissage reconnaissance.

Il était question dans ce travail de tester la validité d'une telle hypothèse en mettant en balance l'apport de performance obtenu lorsqu'on utilise une matrice de covariance pleine par rapport à la complexité accrue de calcul qui en résulte.

Il s'avère que cette hypothèse est amplement justifiée dans le cas de l'utilisation des coefficients MFCCs et LPCEPSTRA mais ne l'est pas pour le cas des LPC. A noter que le meilleur résultat de reconnaissance étant obtenu pour les MFCCs.

Le travail présenté dans ce mémoire avait pour ambition de mettre la lumière sur la validité de l'une des hypothèses les plus adoptées dans la modélisation Markovienne, à savoir l'utilisation d'une matrice de covariance diagonale pour les gaussiennes des modèles. D'après la série d'expérimentations réalisée dans le troisième chapitre, il s'avère que cette hypothèse est justifiée dans le cas de l'utilisation des MFCCs comme schéma de codage du signal vocal. En effet, les résultats obtenus dans ce cas de figure lorsqu'on utilise une matrice de covariance pleine, bien que légèrement supérieurs à ceux obtenu pour une matrice de covariance diagonale, ne justifient pas l'énorme volume supplémentaire de calcul qui en résulte. L'adoption d'une matrice diagonale est bien fondée dans ce cas. La même remarque s'applique dans le cas d'un codage de la parole par les coefficients LPCEPSTRA. La raison principale à cela provient du fait que les coefficients MFCC et LPCEPSTRA sont décorrélés, ce qui conduit à une matrice de covariance ayant des valeurs significatives seulement sur sa diagonale.

Dans le cas de l'utilisation des coefficients LPC, l'approximation de la matrice de covariance par sa diagonale n'est pas justifiée car il y a une grande différence de performance lorsqu'on utilise une matrice pleine et ce par rapport à rapport au choix d'une matrice diagonale. En d'autres termes, la diminution de la complexité du calcul lors de l'adoption d'une matrice diagonale s'obtient en contre partie d'une dégradation très significative des performances de reconnaissance. Ce travail s'inscrit dans la continuité d'une série de travaux de PFEs antérieurs dans l'optique de cerner tous les aspects de la modélisation Markovienne et de mettre la lumière sur un certain nombre d'hypothèses communément adoptées afin de tester leur validité.

Dans le même d'ordre d'idées, il peut être judicieux de tester l'effet de la durée des unités phonétiques dans un mot sur les performances de reconnaissance d'un système à base de HMM. En effet, cet aspect étant considéré comme une des faiblisses de cette approche, on peut être tenté d'explorer à quel point ce détail pourra-t-il pénaliser les performances de reconnaissance

Liste de références bibliographiques

[1] Eléa, "Fiche pédagogique Le son ", page 1-2 :2012.

[2] BENDAHMANE Abderrahmane, "Cours de Traitement Automatique de la Parole" :2013-2014.

[3] René Boite, "Traitement de la parole " collection électricité, presses polytechniques et universitaires Romandes, PPUR press polytechniques : 2000.

[4] Maurice Bellanger, "Traitement numérique du signal ", page 32.

[5] Steve Young Gunnar Evermann Mark Gales Thomas Hain Dan Kershaw Gareth Moore Julian Odell Dave Ollason Dan PoveyValtchoValtchev Phil Woodland, "The HTK book", Microsoft corporation : 1995..

[6] Bouyahia Farida et Essaidi Hanane , " Etude comparative entre les HMMs continus et les HMMs discrets. Application à la RAP", thèse de master, Université Dr Yahia Fares Médéa : 2020-2021.

[7] MESS ADEG Djemil, "SYSTÈME D'AIDE AUX NON-VOYANTS PAR COMMANDE VOCALE" :2011-2013.

[8] L.R. Rabiner and B.H.Juang ," fundamentals of speech Recognition Prentice-Hall Inc":1993

[9] BENDAHMANE Abderrahmane, "Cours de traitement de la parole", Université Mohammed Boudiaf d'Oran : 2013-2014.

[10] Jean Baptiste ET Voumale Roundji, "Modelés de Markov cachés", Université de Québec, Montréal : septembre 2014.

[11] Meberbeche Kamel, Modèles de Markov cachés : "application à la reconnaissance automatique de la parole", Université Mouloud – Mammeri, TiziOuzou.

[12] Steve Young Gunnar Evermann Mark Gales Thomas Hain Dan Kershaw Gareth Moore Julian Odell Dave Ollason Dan PoveyValtchoValtchev Phil Woodland, "The HTK book", Microsoft corporation : 1995.

[13] Esma Fatıma Bilgin Tas¸demir, "A Large Vocabulary Online Handwriting Recognition System for Turkish", These, Sabancı University June: 2018.

[14] Ferrah Dalila, "Reconnaissance automatique de la parole arabe par les HMMs A L'aide de HTK toolkit ", Université SAAD Dahleb, Blida : 2013-2014.

[15] Khelfi Amel, "fenêtrage multiple pour la reconnaissance robuste de la parole", thèse de master, Université Dr Yahia Fares Médéa : 2016-2017.

[16] Nicolas Moreau, Basic tutorial HTK (v.3.1) : Un Système de Reconnaissance Automatique de la Parole, Ceci est la traduction française par Antoine Chemardinde:02-02-2002.

[17] Ferrah Dalila, "Reconnaissance automatique de la parole arabe par les HMMs A L'aide de HTK toolkit ", Université SAAD Dahleb, Blida : 2013-2014.