République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Amar Telidji- Laghouat

FACULTÉ : Technologie
DÉPARTEMENT : Electronique

MÉMOIRE DE MASTER

Présenté par : BEDJELADJEL FIROUZE

DOMAINE : Technologie
FILIERE : Télécommunications
OPTION : Réseaux et Télécommunications

Thème

La Reconnaissance Automatique de la Parole Arabe pour les mots isolés.

Jury de soutenance :

Nom et Prénom	*Qualité*
Mr. GUEFFAF Hamza	Président
Mr. REGGAB Mourad	Examinateur
Mr. KORIBA Mustapha	Encadreur
Mr. REGUIGUE Mourad	Co-Encadreur

Promotion : septembre - 2020

ملخص

الهدف من هذه الرسالة هو الحصول على قاعدة بيانات وتنفيذها، تتكون من أول عشرة أرقام من اللغة العربية الكلاسيكية من 0 إلى 9 ومجموعة من الجمل الصحيحة نحويًا ودلالة. تم تسجيل قاعدة البيانات هذه في ظل الظروف الحقيقية للتحليل الصوتي لقاعدة البيانات هذه تم إجراؤها باستخدام طريقة MFCC وزودتنا بسلسلة من ناقلات الإدخال لنظام التعرف التلقائي على الكلام المشار إليه (RAP) .التي قمنا بتطويرها أيضًا. يعتمد ذلك على استخدام نماذج ماركوف المخفية .(HMMs)لهذا ، قمنا باستغلال الوظائف التي يوفرها برنامج HTK (Hidden Markov Model ToolKit) لتحقيقه. سيسمح تقييم أداء نظام RAP لطريقة تحليل قاعدة البيانات بإبراز تأثير المعلمات.

مفاتيح :HTK, PLP, MFCC, LPC, HMM, RAP

Résume

L’objectif de ce mémoire est l’acquisition et la mise en œuvre d’une base de données, constituée des dix premiers chiffres de l’arabe classique de 0 à 9 et d’un corpus de phrases syntaxiquement et sémantiquement correctes. Cette base de données a été enregistrée dans des conditions réelles de l’analyse acoustique de cette base de données a été effectuée en utilisant la méthode MFCC et nous a fourni une série de vecteurs d’entrée pour le système référer de Reconnaissance Automatique de la Parole (RAP) que nous avons, par ailleurs, élaboré. Celui-ci est basé sur l’utilisation des modèles de Markov Cachés HMM (Hidden Markov Model). Pour cela, nous avons exploité les fonctionnalités offerts par le logiciel HTK(Hidden Markov Model ToolKit) pour sa réalisation. L’évaluation des performances du système RAP pour la méthode d’analyse de la base de données permettra de mettre en exergue influence de la paramétrisation.

Mots clés : RAP, HMM, LPC, MFCC, PLP, HTK.

Summary

The objective of this thesis is the acquisition and implementation of a database, consisting of the first ten digits of classical Arabic from 0 to 9 and a corpus of syntactically and semantically correct sentences. This database was recorded in real conditions of the acoustic analysis of this database was carried out using the MFCC method and provided us with a series of input vectors for the referenced Automatic Speech Recognition system. (RAP) that we have also developed. This is based on the use of Hidden Markov Models (HMMs). For this, we have exploited the functionalities offered by the HTK software (Hidden Markov Model ToolKit) for its realization. The evaluation of the performances of the RAP system for the database analysis method will allow to highlight the influence of the parameterization.

Key words: RAP, HMM, LPC, MFCC, PLP, HTK.

Dédicace

Toutes les lettres ne sauraient trouver les mots qu'il faut … tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance …

Je dédie ce modeste travail à :

A mes chers et respectueux PARENTS

Mon soutien moral et source de joie et d bonheur, celui qui s'est toujours sacrifié pour me voir réussir à toi **MON PERE**.

A la lumière de mes jours, la source de mes efforts, ma vie et mon bonheur ; **MAMAN** que j'adore.

Puisse **ALLAH** tout puissant vous garder et vous procurer santé et bonheur

A mes chères sœurs : **FATIMA, FERIAL, HANINE** et **AFAF**, et mes frères **IMAD** et **YACINE** de leur souhaite tout le succès … tout le bonheur

Spéciale dédicace pour la personne qui a partagé tout le travail, a supporté mon humeur au moment de stresse, mon meilleure ami **ÃSSĘM**

A mes chères amis **KHADIDJA, FERIAL, NAWAL, BOUCHRA** et **AMOULA** d'être toujours à mes coté

Et à tous ceux qui ont contribué de près ou de loin pour que ce projet soit possible, je vous dis merci.

B. FIROUZE

REMERCIMENT

Nous tenons tout d'abord à remercier **ALLAH** le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail, et qui nous a donné le courage durant ces langues années d'étude.

Ce modeste travail est la résultante de la contribution de plusieurs personnes dont nous tenons à remercier vivement :

Nous tenons à remercier **Mr. Koriba Mustapha** qui a accepté de nous encadrer, ses commentaires et réflexions nous ont permis de trouver des applications intéressantes aux travaux que nous menons. Merci pour l'aide et les conseils qu'il nous a fournis durant la rédaction de ce mémoire.

Nous voudrons également exprimer toute nos reconnaissances aux membres de jury pour l'intérêt qu'ils ont porté à ce travail et pour l'honneur qu'ils nous ont fait pour juger ce travail.

Nous souhaitons également à remercier le personnel du département d'électronique, de la faculté de technologie à l'université de Ammar Thelidji – Laghouat, pour les facilités qu'ils nous ont accordé.

Nous remercions aussi tous ceux qui ont aidé de près ou de loin à la réalisation de ce travail.

Nous tenons enfin, à remercier les membres de nos familles pour leur incessant soutien et plus particulièrement nos parents qui nous ont guidé sur le chemin des études.

Un grand merci à tous !

LISTE DES ACRONYMES

AR : Auto-Régressif

HMM : Hidden Markov Model (Modèle de Markov Cachées)

HTK: Hidden Markov Model Toolkiit

LA : Intelligence Artificielle

LP : Prédication Linéaire

LPC: Linear Prédictive Coding

MFCC: Mel-Scale Frequency Cepstral Coefficients

MLE: Maximum Likelihood

MMC : Modèles de Markov Cachés

PLP : Perceptual Linear Predictive

SAMPA: Speech Assessment Méthode Phonetic Alphabet

SR : Système de Reconnaissance

TF : La Transformée de Fourier

TIMIT: Texas Instruments Massachusetts Institute of Technology

TRF: Transformée de Fourier Rapide

TABLE DES MATIERES

LISTES DES FIGURES

LISTE DE TABLEAUX

Introduction Générale

La parole est la manière naturelle et, en conséquence, la forme la plus intéressante et la plus commune de communication humaine. À la différence d'autres moyens électroniques de communication, les systèmes utilisant la parole offrent à l'utilisateur non entraîné un accès simple et naturel. Elle permet d'avoir un accès immédiat à une information sans avoir à parcourir toute une arborescence hiérarchique de menus. L'utilisation des raccourcis clavier ou des langages de commande existe toujours, mais la prononciation d'un seul mot peut remplacer jusqu'à une dizaine de commandes élémentaires effectuées à l'aide de touches fonctions ou de souris et représente ainsi un effort mnémotechnique moindre.[1]

Le traitement de la parole est Aujourd'hui une composante fondamentale des sciences de l'ingénieur. Située au croisement du traitement du signal numérique et du traitement du langage (c'est-à-dire du traitement de données symboliques), cette discipline scientifique a connu depuis les années 60 une expansion fulgurante, liée au développement des moyens et des techniques de télécommunications. [2]

Le but de la Reconnaissance Automatique de la Parole (RAP) est de développer des techniques et des systèmes permettant aux ordinateurs d'accepter la parole comme entrée. La reconnaissance de la parole peut donc être vue comme une opération de transformation du signal de parole en texte en utilisant l'information contenue dans le signal et la connaissance a priori du domaine.

Les applications de la RAP sont aussi nombreuses que diversifiées, elles peuvent être grossièrement regroupées en quatre catégories :

- Les applications téléphoniques.
- Les applications multimédia,
- Les applications industrielles,
- Les applications médicales, etc…

Elles existent là où la parole peut remplacer ou compléter une interface existante pour communiquer avec une machine, par exemple, pour accéder à un service ou contrôler la fonctionnalité d'un équipement. La parole s'impose parfois comme le seul mode de communication comme, par exemple, dans les applications *mains-libres* où l'utilisateur ne touche pas l'équipement.

Les modèles de Markov cachés, introduits fin des années 60, début des années 70, sont devenus la solution par excellence aux problèmes de la reconnaissance de la parole. En effet, ces modèles sont riches en structures mathématiques et par conséquent peuvent être utilisés dans un large domaine d'applications. En outre, ces modèles donnent de remarquables résultats en pratique quand ils sont correctement appliqués.[1]

Le but de notre travail porte sur l'influence de la para-métrisation du signal vocal acquis dans des conditions réelles sur les performances d'un système de reconnaissance automatique de la parole (RAP) basé sur les modèles de Markov cachés (MMC ou HMM : Hidden Markov Models) que nous avons élaboré. L'analyse acoustique du signal vocal est basée sur différentes techniques : l'analyse cepstrale dans l'échelle Mel traduite par les coefficients MFCC (Mel Frequency Cepstral Coefficients).

Nous avons d'abord acquis puis enregistré une base de données constituée des dix premiers chiffres de l'arabe classique et d'un corpus de phrases syntaxiquement et sémantiquement correctes. Ce corpus été expertisé par des linguistes de l'université de Laghouat.

Afin de présenter convenablement notre travail, nous avons organisé ce mémoire en trois chapitres :

- Dans le premier chapitre, nous décrivons tout d'abord les caractéristiques du signal de la parole, et nous présenterons les méthodes d'analyse les plus adaptées à la RAP :
  - ❖ Par prédiction lineaire (LPC, Linear Prédictive Coding),
  - ❖ Par extraction des coefficients cepstraux à l'echelle mell (MFCC, Mel-scale Frequency Cepstral Coefficients),
  - ❖ Par prédiction linéaire perceptuelle (PLP, Perceptual Linear Predictive).
- Dans le deuxième chapitre, nous introduisons les notions mathématiques et les bases nécessaires à l'utilisation des modèles des Markov cachés pour la reconnaissance de la parole.
- Le dernier chapitre est consacré à la présentation des différents résultats que nous avons obtenus en appliquant les méthodes qui ont fait l'objet de ce mémoire.

Nous terminons par une conclusion résumant notre apport et donnant les perspectives éventuelles au travail réalisé.

CHAPITRE I

La Reconnaissance Automatique De La Parole

I.1. Introduction

Dans le traitement de la parole, le signal de parole doit d'abord être transformé et compressé pour un traitement ultérieur. Il existe de nombreuses techniques d'analyse de signal qui sont utilisées pour extraire les caractéristiques importantes et compresser le signal sans perdre aucune information importante [1] [2].

Dans ce chapitre, nous expliquerons d'abord les caractéristiques particulières du signal vocal, puis présenterons les méthodes d'analyse les plus appropriées pour la reconnaissance automatique de la parole.

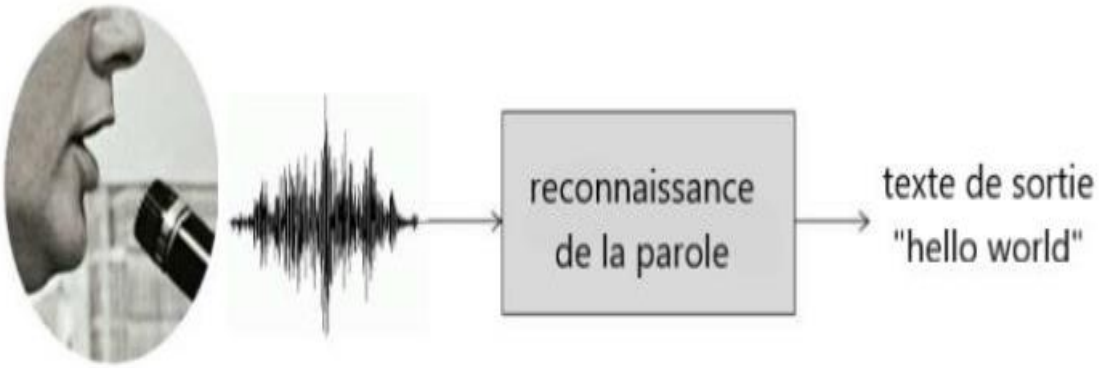

Figure I.1: mécanisme du système de reconnaissance vocale.

I.2. Signal de la parole

Avant de passer au processus de codage du son dans l'ordinateur ou sa synthèse, il faut d'abord bien comprendre le son lui-même. Nous commencerons donc par la définition du son.

I.2.1. Définition

Le son est une vibration de l'air. A l'origine de tout son, il y a du mouvement (par exemple, un fil vibrant, une membrane de microphone ...). Ce sont les phénomènes vibratoires créés par la source sonore qui définit les particules d'air en mouvement. Avant d'atteindre notre oreille, ce mouvement entre molécules à une vitesse de 331 m / s est transmis dans l'air à une température de 20 ° C. C'est ce qu'on appelle la diffusion.[3]

I.2.2. Caractéristiques du son

Le son est défini par trois paramètres :

- L'amplitude du son correspondant à la différence de pression maximale de l'atmosphérique due aux oscillations, (volume sonore)..
- La dynamique qui permet de mesurer la différence entre le volume sonore maximum et le bruit de fond. La dynamique se mesurer en décibels (dB). En fait, c'est le rapport entre le niveau maximum, à réduction de distorsion, et le niveau minimum acceptable, à la limite du niveau de bruit de fond.
- Le timbre qui est un paramètre beaucoup plus subjectif, est-ce qui distingue deux sons de même hauteur et amplitude. C'est une idée qualitative, qui dira, par exemple, que le son est clair ou profond ...[3]

I.2.3. Avantages de la parole

Si nous prenons le modèle humain comme référence, les avantages de la parole semblent cruciaux à première vue :

- **Naturel** : La parole est la méthode de communication la plus naturelle entre humains, du fait que l'apprentissage se fait dès l'enfance, ce qui est loin d'être une maîtrise de l'écriture.
- **Rapidité/efficacité** *:* plusieurs études d'ergonomie montrent que le débit en parole spontanée est de l'ordre de 200 mots/minute à comparer aux 60 mots/minute d'un expert pour la frappe au clavier. L'efficacité de la parole ne provient pas seulement de ce qu'elle permet un débit d'informations plus élevé que d'autres modes de communication, mais également de ce qu'elle peut être aisément utilisée en superposition avec ceux-ci. La parole laisse l'utilisateur libre de ses mouvements, elle est donc particulièrement adaptée aux applications dans lesquelles il s'agit pour l'utilisateur de conduire plusieurs tâches simultanément, ou de contrôler des processus complexes qui monopolisent gestes et/ou vision [4]
- **Extension du champ d'action :** la parole permet d'avoir un accès immédiat à une information sans avoir à parcourir toute une arborescence hiérarchique de menus. Il est toujours possible bien sûr d'utiliser des raccourcis clavier ou des langages de commande, mais la prononciation d'un seul mot peut remplacer jusqu'à une dizaine de commandes élémentaires effectuées à l'aide de touches fonctions ou de souris et représente ainsi un effort mnémotechnique moindre.[5]

Ces avantages sont si importants qu'il existe déjà sur le marché des appareils à usage limité, mais ils sont néanmoins efficaces. Citons quelques-unes des applications qui sont déjà apparues : .[6]

- Saisie de données audio.
- Donne des ordres en conduisant une voiture ou un avion.
- Aide aux handicapés.
- Chambre d'hôpital avec capacités de commande vocale pour les patients.
- Commande vocale de machines ou robots.
- Commande vocale dans une montre portable, etc.

I.2.4. Complexité du signal de parole

Pour appréhender le problème de la reconnaissance automatique de la parole, il faut comprendre les différents niveaux de complexité et les différents facteurs qui en font un problème difficile.

Le signal de la parole n'est pas un signal ordinaire, il est le vecteur d'un phénomène extrêmement complexe, la reconnaissance automatique de la parole pose de nombreux problèmes. D'un point de vue mathématique il est difficile de modéliser le signal de la parole car ses propriétés statistiques varient au cours du temps.

La complexité du signal de parole provient de la combinaison de plusieurs facteurs, principalement la redondance du signal acoustique, la grande variabilité intra-locuteurs et interlocuteurs, et les effets de la coarticulation en parole continue, qui doivent être pris en compte lors de la conception d'un système de RAP. .[6]

I.2.4.1. Redondance du signal de parole

Le signal de parole est extrêmement redondant. Cette grande redondance lui confère une robustesse à certains types de bruits. De nombreuses recherches sont menées afin de rendre les systèmes de reconnaissance robustes aux bruits, mais les performances humaines sont encore loin d'être atteintes. .[3]

I.2.4.2. Continuité et coarticulation

Tout discours peut être retranscrit par des mots, qui peuvent à leur tour être décrits comme une suite de symboles élémentaires appelés phonèmes par les linguistes. Cela laisse supposer que la parole est un processus séquentiel, au cours duquel des unités indépendantes

se succèdent. Malheureusement, les spécialistes de phonétique eux-mêmes ont parfois des difficultés à identifier individuellement ces unités discrètes dans le signal, même si quelques événements acoustiques particuliers peuvent être détectés. La parole est en réalité un flux continu, et il n'existe pas de pause entre les mots qui pourrait faciliter leur localisation automatique par les systèmes de reconnaissance.

De plus, les contraintes introduites par les mécanismes de production créent des phénomènes de coarticulation. La production d'un son est fortement influencée par les sons qui le précèdent mais aussi qui le suivent en raison de l'anticipation du geste articulatoire. Ces effets s'étendent sur la durée d'une syllabe, voire même au-delà, et sont amplifiés par une élocution rapide. L'identification correcte d'un segment de parole isolé de son contexte est parfois impossible. La prise en compte des phénomènes de coarticulation ne suffit pourtant pas à prédire la réalisation acoustique d'une phrase en raison de la grande variabilité de la parole. .[3] .[6]

I.2.4.3. Variabilité

On distingue généralement deux sources de variabilité qui peuvent rendre deux prononciations d'un même énoncé très différentes, la variabilité interlocuteurs et la variabilité intra-locuteur. .[6]

- ***Intra-locuteurs*** : Une même personne ne prononce jamais un mot deux fois de façon identique. La vitesse d'élocution en détermine la durée. Toute affection de l'appareil phonatoire peut altérer la qualité de la production. Un rhume teinte les voyelles nasales ; une simple fatigue et l'intensité de l'onde sonore fléchit, l'articulation perd de sa clarté. La diction évolue dans le temps : l'enfance, l'adolescence, l'âge mûr, puis la vieillesse, autant d'âges qui marquent la voix de leurs sceaux. .[6]
- ***Interlocuteurs :*** Est encore plus flagrante. Les différences physiologiques entre locuteurs, qu'il s'agisse de la longueur du conduit vocal ou du volume des cavités résonnantes, modifient la production acoustique. En plus, il y a la hauteur de la voix, l'intonation et l'accent différent selon le sexe, l'origine sociale, régionale ou nationale. .[6]

I.3. Méthode d'analyse du signal vocal

La parole est un signal acoustique qui contient des informations d'idée qui se forment dans l'esprit du locuteur. Le signal vocal transfère non seulement des informations sur la langue, mais donne également des informations sur les sons, la syntaxe et le locuteur, c'est-à-dire

l'âge, le sexe, l'origine locale, la santé, l'état émotionnel (humeur du locuteur) et sa caractéristique unique. Une reconnaissance automatique de la parole (RAP) ne prend en compte que les informations acoustiques contenues dans le signal vocal, d'où la nécessité d'une analyse acoustique.

L'objectif de l'analyse acoustique est d'extraire des coefficients représentatifs du signal de parole. Ces coefficients sont calculés à intervalles de temps réguliers. En toute simplicité, le signal de parole est transformé en une série de vecteurs de coefficients, ces coefficients doivent représenter au mieux ce qu'ils sont censés modéliser et doivent extraire le maximum d'informations utiles à la reconnaissance. Cette analyse nécessite un prétraitement du signal de parole avant le calcul des paramètres. .[7]

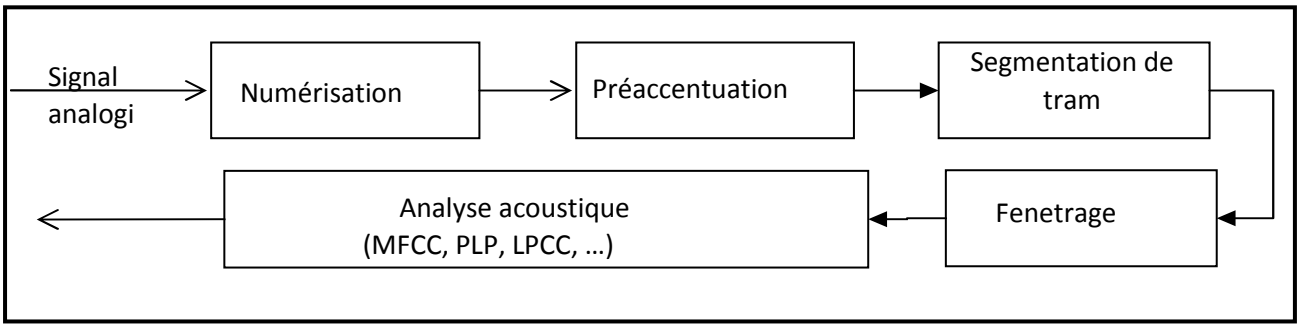

Figure I.2. Prétraitement du signal vocal.

I.3.1. Prétraitement du signal vocal

I.3.1.1. Numérisation

Les signaux que nous utilisons dans le monde réel, comme notre voix, sont des signaux analogiques continus en temps et en amplitude. Pour traiter ces signaux pour la communication numérique, nous devons les convertir sous forme "numérique" (discrète en temps et en amplitude).

Pour cela, nous utilisons un processus appelé théorème d'échantillonnage de Nyquist – Shannon, la fréquence d'échantillonnage fs est supérieure ou égale à la composante de fréquence la plus élevée du signal de message:

$$F_e \geq 2F_{max} \qquad \text{(I.1)}$$

Avec,

F_e : la fréquence d'échantillonnage

F_{max} : la fréquence maximum du signal vocal

I.3.1.2. Préaccentuation

La préaccentuation est une technique utilisée dans le traitement de la parole pour améliorer les hautes fréquences du signal. Il réduit la plage dynamique spectrale élevée.

Dans cette étape, le signal passe à travers un filtre passe-haut pour aplatir spectralement le signal et le rendre moins sensible aux effets de précision finie. La fonction de transfert de ce filtre est :

$$H(z) = 1 - az^{-1} \tag{I.2}$$

Avec : $0.9 < a < 0.98$

I.3.1.3. Segmentation de tram

Les caractéristiques statistiques d'un signal de parole sont généralement invariantes dans un court intervalle de temps. Ainsi, le signal pré-accentué est bloqué ou segmenté en trames. La longueur de trame la plus utilisée est espacée de 20 à 25 ms (millisecondes).

I.3.1.4. Fenêtrage

Une fois la procédure de segmentation de trame terminée, à chaque trame une fonction de fenêtrage est appliquée pour supprimer l'effet des discontinuités aux bords des trames. Le fenêtrage a pour but de réduire l'effet des artefacts spectraux qui résultent du processus de cadrage.

Le fenêtrage dans le domaine temporel est une multiplication ponctuelle de la trame et de la fonction de fenêtre. Par conséquent, chaque trame est multipliée par une fonction de fenêtre w [n] de longueur N + 1, où N est la longueur de trame. La fenêtre la plus populaire est la fenêtre de Hamming donnée par :

$$W[n] = \begin{cases} 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right) & ,0 \le n \le N \\ 0 & ,ailleurs \end{cases} \tag{II.3}$$

N : taille de la fenêtre

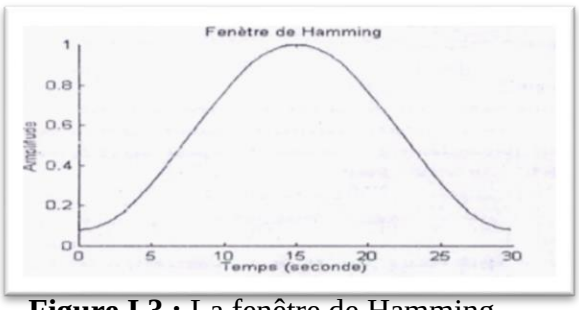

Figure I.3 : La fenêtre de Hamming.

I.3.2. Analyse par transformée de Fourier

L'analyse fréquentielle de la parole se ramène aux opérations de la transformée de Fourier (TF) et n'a d'intérêt que si elle s'applique à une période stable du signal vocal, donc sur une période assez courte. Le spectre à court terme du signal *s(n)* se calcule à partir d'une fenêtre *h(n)* qui permet d'isoler une portion du passé récent de *s(n)* :

$$S(w,n) = \sum_{k=-\infty}^{k=+\infty} s(n)h(n-k)\exp(-jwn) \qquad \text{(I.4)}$$

La quantité $|S(w,n)|^2$ est le spectre de puissance à court terme. L'implantation algorithmique efficace associée à la TF est la transformée de Fourier rapide (TFR). Elle présente de nombreux avantages en tant que méthode d'analyse fréquentielle. La rapidité de sa mise en œuvre l'a propulsé au rang d'élément incontournable des systèmes de traitement du signal.

La TFR permet aussi une représentation fréquentielle du signal aussi fine que l'on souhaite. De plus, pour une étude qualitative de la parole, la TFR est très intéressante parce qu'elle permet une représentation par spectrogramme (évolution du spectre dans le temps) de qualité. Mais, après la naissance de la notion de représentation temps-fréquence, des études théoriques ont permis de mettre à jour quelques désavantages de la TFR qui sont impossibles à éliminer et qui constituent ainsi les limites de l'exploitation de la TFR [4].

Malheureusement les limites théoriques relatives aux représentations temps-fréquence ne sont pas les seuls problèmes de la TF. Le défaut majeur de la TF pour l'étude de la parole vient de l'inévitable intermodulation source/conduit présente dans le spectre qui ne permet pas de connaître précisément la hauteur du fondamental. Cette intermodulation est due à la convolution qui est réalisée par le conduit vocal sur la fréquence fondamentale produite par les cordes vocales. La déconvolution ne pouvant pas être réalisée par une simple transformée, il a donc fallu développer une technique particulière capable de la réaliser pour fournir ces deux informations utiles à l'analyse de la parole. L'étude des représentations temps-fréquence et les limites de la TF ont donc poussé à créer des méthodes de traitements de signal plus adaptées à la parole.[8]

I.3.3. Analyse par codage prédictif linéaire LPC

Le codage prédictif linéaire (LPC, Linear Prédictive Coding) est une technique de codage et de représentation de la parole [8]. Elle s'appuie principalement sur l'idée que le système

phonatoire peut être modélisé par un filtre linéaire .Ce filtre est excité par un train d'impulsions pour les sons voisés et aléatoires pour les sons non voisés. Il s'agit donc de prédire le signal à un instant n à partir des p échantillons précédents :[8]

$$s(n) = -\sum_{k=1}^{p} a_k s(n-k) + Gu(n) \tag{I.5}$$

u(n) étant l'entrée du filtre prédicteur suivant :

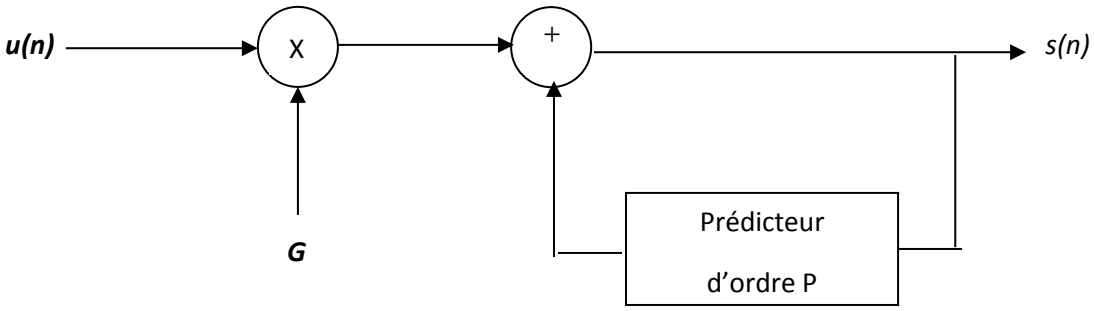

Figure I.4: Filtre prédicteur linéaire.

En supposant que cette entrée *u(n)* soit totalement inconnue, le signal de parole à un instant n peut être prédit par une combinaison linéaire des *p* échantillons précédents, soit :

$$\hat{S}(n) = -\sum_{k=1}^{P} a_k S(n-k) \tag{I.6}$$

L'erreur de prédiction *e*(*n*) est définie par :

$$e(n) = S(n) - \hat{S}(n) = S(n) + \sum_{k=1}^{P} a_k S(n-k) \tag{I.7}$$

L'énergie résiduelle de prédiction est définie par la somme :

$$E_P = \sum_{n1}^{n2} e^2(n) = \sum_{n_1}^{n_2} \left(S(n) + \sum_{k=1}^{P} a_k S(n-k) \right)^2 \tag{I.8}$$

Si les coefficients a_k sont choisis tels qu'ils minimisent l'énergie résiduelle de prédiction, il suffît pour les obtenir de poser :

$$\frac{\partial E_P}{\partial a_i} = 0 \qquad i = 1,2,.....,P \tag{I.9}$$

Le calcul de cette relation (équation I.9) conduit aux équations suivantes :

$$\sum_{k=1}^{P} a_k \Phi_{ik} = -\Phi_{i0} \qquad i = 1,2,.....,P \tag{I.10}$$

$$\Phi_{ik} = \sum_{n=n_1}^{n_2} S(n-i)S(n-k) \tag{I.11}$$

Ces équations normales (équation I.10), dites de Yule Walker, constituent un système linéaire de *P* équations à *P* inconnues. La résolution de ce système permettra d'obtenir les coefficients a_k du filtre. Parmi les méthodes de minimisation de l'énergie résiduelle de prédiction donc de résolution du système, on trouve principalement la méthode d'autocorrélation et la méthode de covariance.

I.3.3.1. Méthode d'autocorrélation

En raison de la nature variant dans le temps du signal de parole, les coefficients de prédiction doivent être estimés à partir de courts segments de signaux de parole (10 à 40 ms) où les caractéristiques des signaux de parole sont constantes dans cette plage [8].

L'énergie résiduelle de prédiction E_p, définie dans (l'équation I.8), est minimisée sur une durée infinie :

$$E_P = \sum_{n=-\infty}^{+\infty} e^2(n) \tag{I.12}$$

La fonction d'autocorrélation est définie par :

$$R(i) = \sum_{n=-\infty}^{+\infty} S(n)S(n-i) \tag{I.13}$$

$$R(i) = R(-i)$$

Le signal vocal est défini pour toutes les valeurs du temps ; il est identiquement nul en dehors d'une séquence de N échantillons ceci équivaut à multiplier le signal vocal par une fenêtre de longueur finie correspondant à N échantillons. La sommation infinie de (l'équation I.12) se ramène donc à une somme finie, soit :

$$E_P = \sum_{n=0}^{N-1+P} e^2(n) \tag{I.14}$$

L'équation I.11 devient alors :

$$\Phi_{ik} = \sum_{n=0}^{+\infty} S(n-i)S(n-k) = \sum_{n=-\infty}^{+\infty} S(n)S(n+i-k) \tag{I.15}$$

Dans ce cas Φ_{ik} n'est autre que la fonction d'autocorrélation évaluée pour ($i-k$), soit :

$$\Phi_{ik} = R(i-k) \tag{I.16}$$

Le système donné par l'équation I.10 s'écrira alors sous la forme suivante :

$$\sum_{k=1}^{P} a_k R(i-k) = -R(i) \qquad i = 1,2,.....,P \tag{I.17}$$

La méthode d'autocorrélation est largement utilisée parce que d'une part elle conduit à un système d'équations d'une structure particulière. La matrice des valeurs d'autocorrélation est une matrice Toeplitz. En effet, elle est symétrique et tous les éléments sur chaque diagonale sont identiques, ce qui facilite la résolution. D'autre part, elle assure la stabilité du modèle auto-régressif trouvé. Différents algorithmes permettent en effet la résolution du système (équation I.17) par une récursion sur l'ordre de prédiction. Parmi ces algorithmes nous pouvons citer l'algorithme de Levinson-Durbin et l'algorithme de Leroux-Gueguen [9] [10].

I.3.3.2. Méthode de covariance

Le signal est étendu par p échantillons en dehors de la plage normale de 0≤n≤N-1 pour inclure p échantillons se produisant avant n = 0 (ils sont disponibles) et élimine le besoin d'une fenêtre effilée

Dans cette méthode l'énergie de prédiction E_P est minimisée sur une durée finie :

$$E_P = \sum_{n=M_1}^{M_2} e^2(n) \tag{I.18}$$

Soit :

$$E_P = \sum_{n=0}^{N-1} e^2(n)$$

La relation (équation I.11) devient alors :

$$\Phi_{ik} = \sum_{n=0}^{N-1} S(n-i)S(n-k) \tag{I.19}$$

Φ_{ik} définie dans (l'équation I.18) n'est rien d'autre que la fonction de covariance $\sigma(i,k)$. Ainsi le système de (l'équation I.10) s'écrira alors :

$$\sum_{k=1}^{P} a_k \sigma(i,k) = -\sigma(i,0) \ , \qquad i = 1,2,.....,P \tag{I.20}$$

Dans ce cas la matrice (*P x P*) des valeurs de covariance est symétrique mais non de Toeplitz. Dans cette méthode, les propriétés statistiques de l'estimateur sont meilleures que dans le cas de la méthode d'autocorrélation, mais la matrice a une structure moins simple que celle de Toelplitz ce qui rend la résolution des équations de Yule-Walker un peu plus coûteuse. Un autre inconvénient de cette méthode est qu'elle ne garantit pas la stabilité du modèle auto-régressif trouvé. Parmi les algorithmes de résolution nous pouvons citer l'algorithme de Gram-Schmit et l'algorithme de Morf [11] [12] [13].

I.3.4. Analyse cepstrale

L'intermodulation source conduit observée sur le spectre calculé par TFR rend son utilisation pour la mesure des formants F_i et de la fréquence fondamentale F_0 très difficile. Le lissage cepstrale est une méthode qui vise à séparer la contribution du conduit et de la source d'excitation par déconvolution. Pour cela on fait l'hypothèse que le signal vocal *s(n)* est produit par un signal excitateur *g(n)* (source glottique) traversant un système linéaire passif de réponse impulsionnelle *b(n)* (conduit). Avec ces hypothèses on peut écrire :

$$s(n) = g(n) \otimes b(n) \tag{I.21}$$

Pour déconvoluer plus aisément *s(n)* il suffit de transposer le problème par homomorphisme dans un espace ou l'opérateur $\otimes$ (convolution) correspond à un opérateur + (addition).

En pratique cette transposition par homomorphisme est réalisée par les étapes schématisées sur la figure suivante :

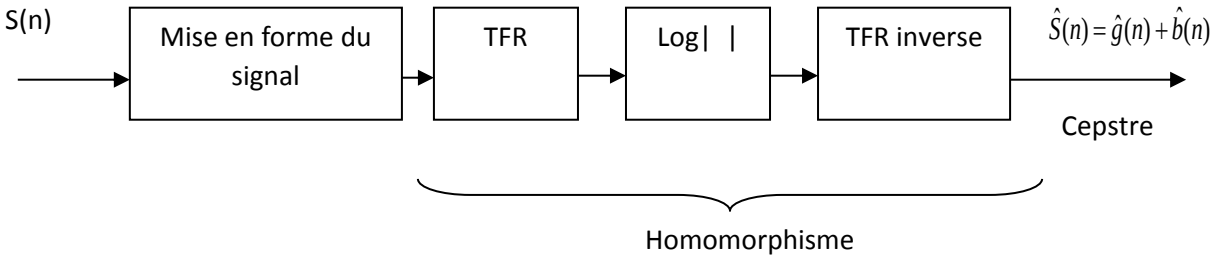

Figure I.5 : Analyse homomorphique de la parole.

Où $\hat{S}(n)$ sont les coefficients cepstraux approchés prenant leurs valeurs dans un domaine pseudo-temporel réel appelé domaine quéfrentiel. La structure de la parole et les hypothèses sur la source d'excitation et du conduit vocal permettent de dire que :

$\hat{g}(n)$ Se réduit théoriquement à une séquence d'impulsions de période n_0 (n_0 correspond a la fréquence fondamentale F_0).

$\hat{b}(n)$ décroît rapidement (en 1/n) avec n et devient négligeable pour $n > n_0$.

Dans ces conditions, on peut admettre que la contribution du conduit est localisée dans les basses quéfrences ($n < no$) et que la séquence d'impulsions reflète la contribution de la source. Cette méthode de calcul des cepstres est élémentaire, il existe d'autres méthodes itératives effectuant un lissage, ce qui permet d'obtenir des cepstres de meilleure qualité. [11]

I.3.4.1. Calcul des cepstres à partir des coefficients LPC

Le calcul des cepstres par la méthode décrite précédemment (analyse homomorphique) est généralement moins utilisé du fait de la charge de calcul importante associée au calcul de la TFR et de la TFR inverse. On lui préfère une méthode paramétrique. Cette méthode paramétrique permet de déterminer les coefficients cepstraux des signaux de parole à partir des coefficients LPC. C'est ainsi que le signal de parole est considéré comme engendré par un filtre auto-régressif (AR) dont il faut déterminer les coefficients a_i en utilisant des méthodes classiques de prédiction linéaire comme la méthode d'autocorrélation par exemple. Le calcul des cepstres est alors basé sur une procédure récursive liant les coefficients cepstraux (C_m) et les coefficients de prédiction (a_i). Cette procédure récursive est traduite par les équations suivantes :

$$\ln\left[\frac{1}{A_P(z)}\right] = \sum_{n=1}^{\infty} C_q(n) Z^{-n} \qquad \text{(I.22)}$$

$$C_0 = \ln G^2 \quad , \; G^2 \text{ est le gain du modèle LPC} \qquad \text{(I.23)}$$

$$C_m = a_m + \sum_{k=1}^{m-1}\left(\frac{k}{m}\right) C_k a_{m-k} \quad , \quad \text{pour} \quad 1 \leq m \leq \text{P} \qquad \text{(I.24)}$$

$$C_m = \sum_{k=1}^{m-1}\left(\frac{k}{m}\right) C_k a_{m-k} \quad , \quad \text{pour} \quad m > \text{P} \qquad \text{(I.25)}$$

I.3.4.2. MFCC (Mel-scale Frequency Cepstral Coefficients)

Les coefficients MFCC (Figure I.6) sont une extension des coefficients cepstraux par le passage de l'échelle fréquentielle linéaire à une échelle fréquentielle non linéaire proche de l'audition humaine. Cette échelle non linéaire est l'échelle perceptive Mel. Celle-ci est plus

exactement linéaire pour les basses fréquences (inférieures à 1000Hz) et logarithmique pour les hautes fréquences. C'est ainsi que des filtres répartis linéairement en basses fréquences et logarithmiquement en hautes fréquences (Figure I.7) sont utilisés afin de capturer les caractéristiques phonétiques importantes du signal de parole. Ces filtres possèdent la caractéristique suivante : plus la fréquence est élevée, plus la bande passante est large ce qui permet une meilleure résolution temporelle des hautes fréquences.

L'échelle *Mel* peut être définie par la relation suivante entre la fréquence en Hertz et sa correspondante en mels :

$$B(f) = x \ \log_{10}\left(1+\frac{f}{y}\right) \tag{I.26}$$

Où f est la fréquence en Hz, $B(f)$ est la fréquence en échelle mel de f.

Plusieurs valeurs sont utilisées pour x et y par exemple x =1000/log(2) et y =1000. De nos jours, les valeurs les plus couramment utilisées sont x =2595 et y =700.

Soit un signal discret $\{x[n]\}$ avec $0 \le n \le N-1$, N est le nombre d'échantillons d'une fenêtre analysée, Fe est la fréquence d'échantillonnage, la transformée de Fourier discrète $S[k]$ est obtenue par :

$$S[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} \tag{I.27}$$

Avec : $0 \le k < N$

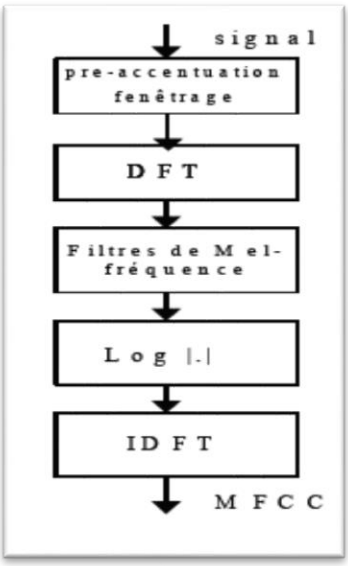

Figure I.6 : Calcul des MFCC

Le spectre du signal est multiplié par des filtres triangulaires (Figure I.7) dont les bandes passantes sont équivalentes en domaine mel-fréquence. Les points frontières $B[m]$ des filtres en mel-fréquence sont calculés ainsi :

$$B[m] = B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1} \tag{I.28}$$

Avec : $0 \le m \le M+1$

Où M est le nombre de filtres, f_h est la fréquence la plus haute et f_l est la fréquence la plus basse pour le traitement du signal. Dans le domaine fréquentiel, les points $f[m]$ discrets correspondants sont calculés par l'équation :

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_l) + m\frac{B(f_h) - B(f_l)}{M+1}\right) \tag{I.29}$$

Avec : $B^{-1}(b) = 700 * \left(10^{b/2595} - 1\right)$

Où B^{-1} est la transformée de mel-fréquence en fréquence.

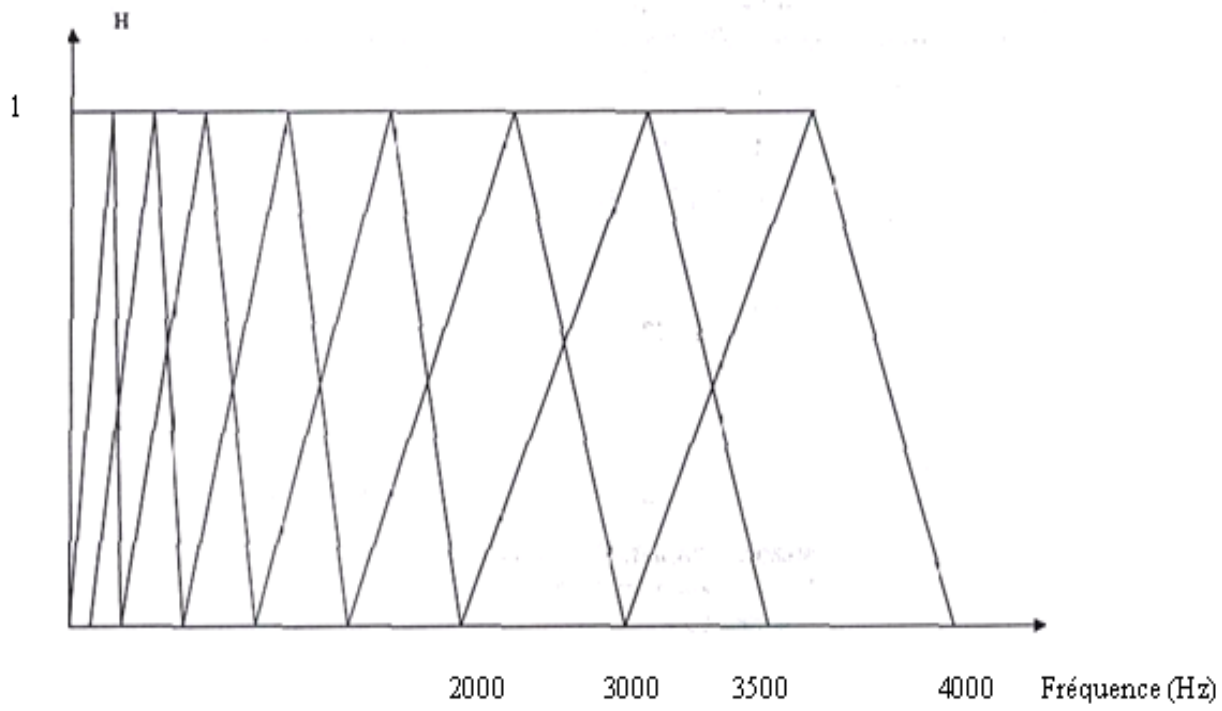

Figure I.7 : Bancs de filtres Mel.

Le coefficient $H_m[k]$ de chaque filtre est déterminé par le système suivant :

$$H_m[k]=\begin{cases}0 & si\ k \le f[m\text{-}1] \\ \dfrac{k-f[m-1]}{f[m]-f[m-1]} & si\ f[m\text{-}1] \le k \le f[m] \\ \dfrac{f[m+1]-k}{f[m+1]-f[m]} & si\ f[m] \le k \le f[m+1] \\ 0 & si\ k \ge f[m+1]\end{cases} \tag{I.30}$$

Pour un spectre lissé et stable, à la sortie des filtres un logarithme d'énergie (ou un logarithme de spectre d'amplitude) est calculé :

$$E[m]=\log\left[\sum_{k=0}^{N-1}|S[k]|^2 H_m[k]\right] \tag{I.31}$$

Avec : $0 \le m < M$

Les coefficients cepstraux de mel-fréquence (MFCCs) peuvent être obtenus par une transformée de Fourier inverse à partir des coefficients aux sorties des filtres. Mais le nombre de MFCCs est moins grand que le nombre des filtres, donc une transformée de cosinus discrète est plutôt utilisée :

$$c[n]=\sum_{m=0}^{M-1}E[m]\cos\left(\frac{\pi n\left(m+\frac{1}{2}\right)}{M_s}\right) \tag{I.32}$$

Avec : $0 \le n < M$

I.3.4.3. Coefficients PLP (Perceptual Linear Predictive)

PLP (Perceptual Linear Predictive) est une technique d'analyse de la parole , fondée sur la modélisation du spectre par un modèle tout pôle suivant un principe identique à la technique de prédiction linéaire (LP). Cependant, la différence réside dans le fait que les paramètres d'un filtre auto-régressif tout pôle sont estimé en modélisant au mieux le spectre auditif. Ceci est fondé sur trois effets auditifs : sélectivité spectrale de bande critique, courbe d'intensité égale et loi de puissance. La (Figure II.8) représente le processus de calcul des coefficients PLP [11].

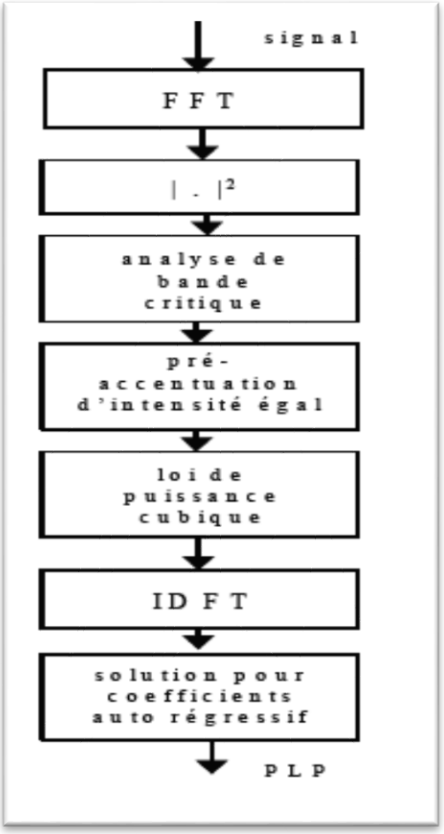

Figure I.8 : Processus de calcul des coefficients PLP.

Pour obtenir un spectre auditif, la courbe de masquage $\Psi(\Omega)$ est tout d'abord utilisée

$$\Psi(\Omega) = \begin{cases} 0 & si\ \Omega \le 1.3 \\ 10^{2,5(\Omega+0,5)} & si\ -1.3 \le \Omega \le -0.5 \\ 1 & si\ -0.5 \le \Omega \le 0.5 \\ 10^{-1,0(\Omega-0,5)} & si\ 0.5 \le \Omega \le 2.5 \\ 0 & si\ \Omega \ge 2.5 \end{cases} \qquad \text{(I.33)}$$

Où Ω est la fréquence de Bark calculée à partir de la fréquence angulaire *w* par la définition :

$$\Omega(w) = 6\ln\left(\frac{w}{1200\pi} + \left(\left(\frac{w}{1200\pi}\right)^2 + 1\right)^{1/2}\right) \qquad \text{(I.34)}$$

Le spectre de puissance du signal $p(\omega)$ (pair et périodique) est convolué avec la courbe de masquage :

$$\Theta(\Omega_k) = \sum_{\Omega=-1.3}^{\Omega=2.3} p(\Omega - \Omega_k)\Psi(\Omega) \tag{I.35}$$

Puis, l'algorithme tente de faire l'approximation de la sensibilité de l'oreille humaine à différentes fréquences par l'intermédiaire d'une fonction de transfert $E(\omega)$:

$$\Xi(\Omega(\omega)) = E(\omega)\Theta(\Omega(\omega)) \tag{I.36}$$

La non linéarité entre l'intensité d'un son et son niveau de perception par l'oreille est réalisé en l'approchant par une loi de puissance :

$$\Phi(\Omega) = \Xi(\Omega)^{\frac{1}{3}} \tag{I.37}$$

Enfin le spectre auditif est modélisé par un modèle tout-pôle. Une transformée de Fourier inverse discrète est appliquée sur le spectre auditif $\Phi(\Omega)$ pour obtenir les valeurs d'autocorrélation. M+1 premiers coefficients d'autocorrélation sont utilisés pour calculer les coefficients auto régressifs du modèle tout-pôle d'ordre M qu'on appelle les coefficients PLP.

I.4. Conclusion

Dans ce chapitre, nous avons présenté les mécanismes de production et de perception de la parole chez les humains ainsi que les caractéristiques du signal vocal responsables en grande partie des difficultés de la tâche de reconnaissance. Par la suite, nous avons décrit les différentes approches de reconnaissance avec le processus de traitement relatif à chacune d'entre elles. C'est ainsi que nous avons présenté les méthodes d'analyse du signal de parole les mieux adaptées et les plus utilisées.

CHAPITRE II

Modèles De Markovs Cachés (HMM)

II.1.Introduction

Les modèles acoustiques utilisés pour la reconnaissance de la parole sont depuis des années principalement basés sur les HMMs (Hidden Markov Models ou Modèles de Markovs Cachés). Les modèles de Markov cachés sont développés par *Andrew Markov* (étudiant de *Chebyshev*), et ils sont premièrement, orientés vers des objectifs linguistiques dans des travaux de littérature Russe. Ceux sont des outils efficaces en modélisation des données séquentielles ou '*time-series Data'*. Utilisés par la suite dans des problèmes de reconnaissance de la parole par *Baker*, leur théorie de base est introduite par *Baum* et ses collègues ont été décrits pour la première fois 1970, mais ce n'est qu'en 1975 qu'ils ont été proposés dans le cadre de la reconnaissance automatique de la parole et se sont imposés depuis comme modèles de référence dans ce domaine[14]

Actuellement, ces modèles sont, de plus en plus adoptés en reconnaissance automatique de la parole, la reconnaissance de formes, le traitement d'images et des signaux, la robotique, etc. Ces applications nécessitent une convergence d'approches du type compréhension, intelligence artificielle (IA), et celles du type traitement du signal, méthodes statistiques.[14]

Un modèle de Markov caché HMM (Hidden Markov Model) est caractérisé par un modèle Markovien à état fini et un ensemble de distributions de sortie. Les paramètres de transition dans la chaîne de Markov modélisent les variabilités temporelles, tandis que les paramètres des distributions de sortie modélisent les variabilités spectrales. Ces deux types de variabilités sont à la base de beaucoup de processus physiques tels que les signaux de la parole ou les signaux issus des systèmes dynamiques.

Les modèles de Markov Cachés doivent leurs succès à l'existence de plusieurs algorithmes très efficaces. Pour l'apprentissage, l'algorithme "Forward-Backward" estime automatiquement et efficacement les paramètres de transition et de sortie, et cette efficacité de l'algorithme "Forward-Backward" permet aux systèmes modélisés par des HMM d'apprendre leurs paramètres à partir d'une grande base de données (corpus d'apprentissage) [14].

En traitement de la parole, les HMM utilisés sont particulièrement d'ordre 1. Nous allons revoir les bases nécessaires à l'utilisation de ces modèles pour la reconnaissance de la parole [15].

II.2.Exemple introductif

Cet exemple sur le modèle des urnes et des balles [14] [16] [17] reflète parfaitement les deux composantes (l'état et l'observation) du processus stochastique d'un HMM.

Supposons que nous avons N urnes (états), voir figure II.1 :

$$S = \{s_1, s_2, s_3, ..., s_N\} \tag{II.1}$$

Chaque urne a son propre mélange de balles colorées (symboles). Chaque balle peut être colorée avec M couleurs possibles ($1 < v_k < M$).

Soit $b_i(v_k)$, la fraction (la probabilité du symbole d'observation) dans l'urne (état) s_i, $1 < i < N$, *o ù*

$$\sum_{k=1}^{M} b_i(v_k) = 1, \quad i = 1, 2, ..., N. \tag{II.2}$$

Soit N+1 goblets: *Go*, G_1, ..., G_N. Chaque goblet a son propre mélange de pierres portant des marques. La marque sur une pierre est considérée comme étant "état 1" ou "état 2" ou ... ou « état N ».

Soient $\pi_1, \pi_2, ..., \pi_N$ les fractions des pierres marquées "état *i*", i = 1,2, ..., N, dans *Go*.

Soient enfin a_{1i}, a_{2i}..., a_{Ni} les fractions des pierres marquées "état i" respectivement dans *Go*, G_1, ..., G_N.

Avec :
$$\sum_{i=1}^{N} \pi_{0i} = 1 \tag{II.3}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \le i \le N \tag{II.4}$$

Générons une suite d'observations de couleurs de balles $O = O_1 O_2 ... O_T$, (Figure II.1).

Tirons aléatoirement une pierre du goblet Go, sa marque est appelée « état i », $1 \le i \le N$. Tirons ensuite une balle aléatoirement de l'urne i, sa couleur est $O_1 = v_k$, $1 \le k \le M$.

Maintenant, tirons une pierre aléatoirement du gobelet G_i, sa marque est appelé « état j », $1 \le j \le N$. Continuons dans cette voie, en utilisant l'état courant pour obtenir à la fois l'observation courante et l'état suivant jusqu'à un total de T observations, (Figure II.1**)**.

A chaque tirage dans une urne, la balle est remise dans la même urne.

Le voile de Ferguson cache cet unique échantillonnage des gobelets. L'observateur obtient seulement une information probabiliste concernant les pierres.

Ce mécanisme, génératif pour créer une suite d'observations est un processus stochastique avec une composante cachée : En générant la suite d'observations des couleurs O, une suite de pierres $Q = q_1 q_2 ... q_T$ est aussi générée. Puisque la suite Q n'est pas observée, elle est alors une suite cachée (ou chemin caché).

Le paramètre vecteur du modèle stochastique est:

$$\lambda = \left[\pi_1 \pi_2 ... \pi_N \, , a_{11} a_{12} ... a_{NN} \, , b_1 (v_1) b_2 (v_2) ... b_N (v_M) \right] \tag{II.5}$$

Le vecteur de probabilité $\Pi = \left[\pi_{01} \pi_{02} ... \pi_{0N} \right]'$ est la distribution initiale des états.

La matrice stochastique $A = \left[a_{ij} \right]$, où la i^{ème} rangée est associée au goblet i, est la matrice de transition d'états.

Ce modèle est un modèle markovien caché d'ordre un à N états. Il est un modèle du premier ordre puisque chaque état sélectionner comme une fonction probabiliste du dernier état prédécesseur.

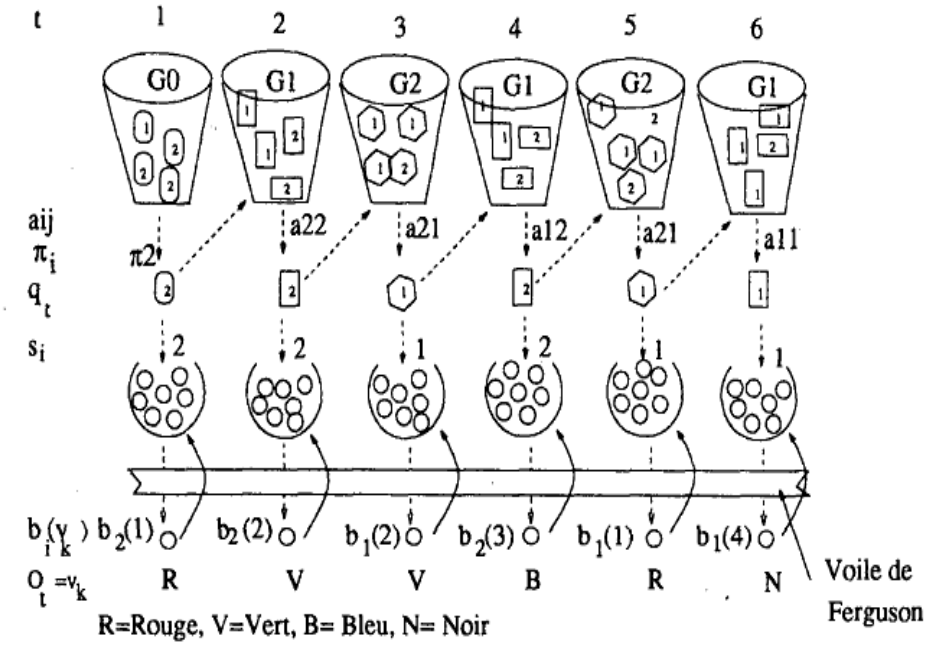

Figure II.1 : Un Modèle markovien caché (HMM) du premier ordre à deux états

Avec

- N = 2 et un ensemble de symboles discrets appelés alphabet de sortie

- M = 4, vk$\in$V = {1, 2, 3, 4} Les urnes sont supposées contenir un grand nombre de balles.

Un modèle markovien caché est dit ayant un **alphabet de sortie** fini si les articles observés s'étendent à un ensemble fini de k éléments. A titre d'exemple :

- L'ensemble de couleurs de balles dans l'exemple des urnes ;
- Les vecteurs caractéristiques de l'alphabet d'une langue quelconque plus l'espace entre les mots ;
- Les symboles d'un dictionnaire de quantification vectorielle

Pour chaque état *i*, le vecteur $b_i = \left[b_i(1) b_i(2) ... b_i(M)\right]'$ est appelé vecteur de probabilité de sortie pour l'état *i*. Ces probabilités de sortie tracent la suite d'états Q à partir de la suite d'observations *O*.

Alternativement, les observations peuvent s'étendre dans un ensemble continu (dénombrable fini).

Dans le choix de la sortie continue, chaque état *i* est associé à son propre paramètre de densité de probabilité b_i.

Un HMM est représenté par son vecteur paramètre $\lambda = [\Pi, A, b_1, b_2, ..., b_N]$.

Dans le cas d'un alphabet fini (discret), la matrice $B = [b_1, b_2, ..., b_N]$ s'appelle la matrice de probabilités d'observations et le modèle λ devient (Π, A, B).

II.3.Principe des modèles de Markov cachés (HMM)

Les HMM sont caractérisés par les paramètres suivants :

II.3.1. Le nombre des états

Nous désignons les états individuels par :

$$S = \{s_1, s_2, s_3, ..., s_N\} \quad \text{(II.6)}$$

Et l'état au temps *t* par q_t, $q_t \in S$.

II.3.2. Le nombre de symboles d'observations distincts

Dans le cas où l'observation O_t à la sortie physique du système est représentée sous forme discrète. Ces symboles correspondent à la sortie physique du système. Nous désignons ainsi l'ensemble de symboles d'observation par :

$$O_t = v_k, \quad v_k \in V = \{v_1, v_2, ..., v_M\} \tag{II.7}$$

II.3.3. La distribution des probabilités des transitions des états

$$A = \{a_{ij}\} \tag{II.8}$$

A : la distribution des probabilités des transition des états.

Où :

$$a_{ij} = p\left[q_{t+1} = s_j \middle| q_t = s_i\right], 1 \le i, j \le N \tag{II.9}$$

Et :

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \le i \le N \tag{II.10}$$

II.3.4. La distribution des probabilités des observations

Dans chaque état j notée B

$$B = \{b_j(O_t)\}, \quad j = 1, 2, ..., N \tag{II.11}$$

Dans le cas où l'observation est représentée sous forme continue, nous avons :

$$\int_{-\infty}^{\infty} b_j(x)dx = 1, \quad 1 \le j \le N \tag{II.12}$$

Et dans le cas où l'observation est représentée sous forme discrète nous avons :

$$b_j(O_t = v_k) = p\left[O_t = v_k \middle| q_t = s_j\right], \quad 1 \le j \le N, 1 \le k \le M \tag{II.13}$$

Avec :

$$\sum_{k=1}^{M} b_j(O_t = v_k) = 1, \quad 1 \le j \le N \tag{II.14}$$

Et dans ce cas B s'appelle la matrice de probabilités des symboles d'observations.

II.3.5. La distribution des probabilités initiales des états Π

$$\prod = \{\pi_i\} \tag{II.15}$$

Où :

$$\pi_i = p\left[q_1 = s_i\right], \quad 1 \le i \le N \tag{II.16}$$

Et :

$$\sum_{i=1}^{N} \pi_i = 1 \qquad \text{(II.17)}$$

On peut conclure que la spécification complète d'un HMM requiert :

- Deux paramètres (N et M pour un HMM discret);
- Définition des vecteurs d'observations ;
- Les distributions des probabilités A, B et Π.

Nous désignons par:

$$\boldsymbol{\lambda} = (\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{\Pi}) \qquad \text{(II.18)}$$

Pour indiquer un modèle complètement spécifié.

Etant donné des valeurs appropriées de N, M, A, B et Π, le HMM peut être utilisé comme un générateur donnant une suite d'observations.

$$O = O_1 O_2 ... O_T \qquad \text{(II.19)}$$

Où :

$$O_t = v_k, v_k \in V, \quad 1 \le k \le M \qquad \text{(II.20)}$$

Dans le cas où l'observation est représentée sous forme discrète, T est le nombre d'observations dans la suite.

La procédure suivante génère une suite d'observations à partir d'un modèle HMM :

1. Choisir un état initial à l'instant $t = 1$, $q_1 = s_i$ avec une distribution de l'état initial $\boldsymbol{\pi}_i$;
2. Choisir O_t selon la distribution de probabilité de l'observation dans l'état c'est-à-dire $b_i(O_t)$;
3. Si $t < T$ passer à un état $q_{t+1}=s_j$ avec la distribution de probabilité de transition d'état pour l'état s_i, c'est-à-dire a_{ij} ;
4. Poser $t = t + 1$.
   - Si $t<T$ retourner à l'étape 2
   - Si non fin de la procédure.

II.4. Trois problèmes fondamentaux d'un HMM

Etant donné un *HMM*, la plupart des applications sont réduites à résoudre les trois problèmes essentiels liés à une chaîne de Markov cachée []. Autrement dit, pour une modélisation à base de chaînes de Markov cachées, il faut répondre aux trois questions suivantes :

Soit $\lambda = (A, B, \Pi)$ un *HMM donné*, et soit $O = (O_1, O_2, \ldots, O_T)$ une séquence d'observation donnée :

II.4.1. Evaluation

Le problème d'évaluation est celui permettant de répondre à la question suivante :

- Calculer $p(O|\lambda)$ *?*

Où $p(O|\lambda)$ est la probabilité d'une séquence d'observations sachant le modèle λ. C'est le problème nommé aussi, par *'Scoring'*. Par exemple, si on dispose de plusieurs modèles compétitifs, cette démarche permet de choisir le meilleur modèle générant cette séquence d'observations.

II.4.2. Optimisation

Dans ce cas-là, on se tente de déterminer la démarche convenable, amenant à la réponse de la question posée ainsi :

- *Comment choisir une séquence d'état* $Q = (q_1, q_2, \ldots, q_T)$ *, qui est optimale ?*

On cherche, donc à décoder les états qui correspondent de mieux à la séquence d'observations ; c'est pour quoi on utilise souvent le nom : *'Decoding'*.

Alors, cette partie est la section dans laquelle on découvre les états cachés d'un *HMM* ; ou bien, trouver la séquence d'état *correcte*. Notons que cette dernière n'existe pas ; c'est plutôt une *estimation meilleure*. C'est pourquoi en pratique, on se sert des critères ; d'où le nom ***optimisation*** donné à ce problème.

II.4.3. Apprentissage

Ce problème s'occupe d'ajuster les paramètres du modèle $\lambda = (A, B, \Pi)$, dans l'objectif de maximiser la probabilité d'observation annoncée au premier problème ; donc :

- Trouver $\hat{\lambda}$? tel que $\hat{\lambda} = argmax\ P(O/\lambda)$.

Cela signifie d'adapter les paramètres (π, A, B) du modèle, pour mieux décrire l'application. La séquence d'observation *O* utilisée, dans ce cas est appelée ***séquence d'apprentissage*** ;

Ce dernier problème est crucial, puisqu'il permette de mieux adapter le modèle choisi au phénomène considéré.

II.5. Solutions des trois problèmes

Dans ce paragraphe, nous allons donner les solutions usuelles des trois problèmes ci-dessus selon.

II.5.1. Premier problème : évaluation

Etant donné une suite d'observations $O_1O_2...O_T$, et un modèle $\lambda = (\Pi, A, B)$, comment peut-on calculer efficacement la probabilité que la suite d'observation O soit produite par λ, c'est-à-dire $p(O|\lambda)$. Autrement dit, comment évaluer le modèle afin de choisir parmi plusieurs celui qui génère le mieux cette suite d'observation. Plusieurs techniques permettent de résoudre ce problème : méthode d'évaluation directe, procédure "Forward-Backward" et Algorithme de Viterbi.

II.5.1.1. Évaluation directe

On veut trouver la probabilité d'une séquence d'observations *O* sachant le modèle $\lambda = (A, B, \Pi)$. C'est déterminer la mesure de vraisemblance $P(O/\lambda)$.

Rappelons que :

$\boldsymbol{P(O_T/\lambda_T)=b_{Q_t}(O_T)}$

Et

$P(O/\lambda)=\sum_Q \boldsymbol{P}(\boldsymbol{O,Q}/\boldsymbol{\lambda})= \sum_Q \boldsymbol{P}(\boldsymbol{O}/\boldsymbol{Q,\lambda}).\boldsymbol{P}(\boldsymbol{O}/\boldsymbol{\lambda})$.

Où :

$\boldsymbol{Q = q_1, q_1, \dots, q_T, q_t = s_i,}$ $\boldsymbol{1 \leq i \leq}$**N**

Tel que :

- $P(O,Q/\lambda)$: est la probabilité jointe de *O* et *Q* sachant ce modèle λ .
- T : est le nombre d'observations

La démarche suivie pour mesurer la probabilité $P(O/\lambda)$.est d'énumérer toutes les séquences possibles de longueur *T*.

Mais pour le moment, considérons$Q = (q_1, q_2, \dots, q_T)$, une séquence d'état fixe. Selon la propriété d'indépendance conditionnelle des observations, la probabilité $P(O/Q,\lambda)$. Est donnée par :

$P(O/Q,\lambda)=\prod_{t=1}^{T} P(O_T/Q_T,\lambda)$

Accordée à l'équation (2.10), la formule (2.12) devient comme suit :

$P(O/Q,\lambda)=\prod_{t=1}^{T} b_{Q_t}(O_t)$

Ou encore :

$$P(O/_{Q,\lambda})=b_{Q_1}O_1.\, b_Q O_2 \ldots . b_{Q_{T-1}}O_{T-1}.\, b_{Q_T}O_T$$

D'une part.

D'autre part, rappelons la formule (2.9):

$$P(Q/_{\lambda})=\pi_{Q_1}\prod_{t=1}^{T-1} Q_t Q_{t+1}$$

La probabilité d'une séquence d'observations *Y* sachant λ est donnée donc, par la somme de toutes les probabilités jointes, associées aux combinaisons d'état possibles, donnant chacune une séquence *X* d'états de même longueur *T*. On écrit :

$$P\left(O/_{\lambda}\right)=\sum_{Q} P(O/_{Q,\lambda}).\, P(Q/_{\lambda})$$

En substituant (2.9) et (2.14) dans (2.15), on obtient la mesure de vraisemblance comme suit :

$$P\left(O/_{\lambda}\right)=\sum_{Q}\pi_{X_1}.\, b_{q_1}(O_1)\,.\, a_{q_1q_2}.\, b_{q_2}(O_2).\, a_{X_2X_3}.\ldots b_{q_{T-1}}(O_{T-1}).\, a_{q_{T-1}q_T}(O_T)$$

Remarque

Le calcul précédant de $P\left(Y/_{\lambda}\right)$, fait appel à un nombre de multiplications de l'ordre de (2.T-1).K^T, associé à (K^T-1)additions [31]. Il semble être difficile à effectuer même pour des petites valeurs de *K* et *T*

A titre d'exemple, pour un nombre d'états de *K*=2, et une séquence d'observations de *T* =100 de longueur, le nombre total d'opérations est de l'ordre de $(2.100.2^{100})=100.\,2^{101}$

II.5.1.2. Procédure Forward-Backward

Dans cette approche, on considère que l'observation peut se faire en deux étapes : d'abord, l'émission de la suite d'observations $O_1O_2...O_t$ et la réalisation de l'état q_i au temps *t*, puis l'émission de la suite d'observations $O_{t+1}O_{t+2}...O_T$ en partant de l'état q_i au temps *t*. Dans ce cas, l'évaluation de l'observation est:

$$p(O|\lambda)=\sum_i \alpha_t(i)\beta_t(i) \qquad \text{(II.24)}$$

Cet accès de calcul peut être résolu en utilisant la procédure *'Forward-Backward'*. Elle est basée sur la détermination des deux probabilités *'Forward'* ;$\alpha_t(i)$, et *'Backward'* ; $\beta_t(i)$,

Où α_t (i) est la probabilité d'émettre la suite $O_1O_2...O_t$ et d'aboutir à q_i à l'instant t sachant le modèle, (Figure II.2), et β_t(i) est la probabilité d'émettre la suite $O_{t+1}O_{t+2}...O_T$ en partant de l'état q_i à l'instant *t* sachant le modèle, (Figure II.3).

Le calcul de α_t (i) se fait avec *t* croissant tandis que celui de β_t (i) se fait avec *t* décroissant, d'où l'expression Forward-Backward [18].

Pour résoudre le problème 1, il suffit de calculer la partie Forward; le calcul de la partie Backward permettra de résoudre le problème 3.

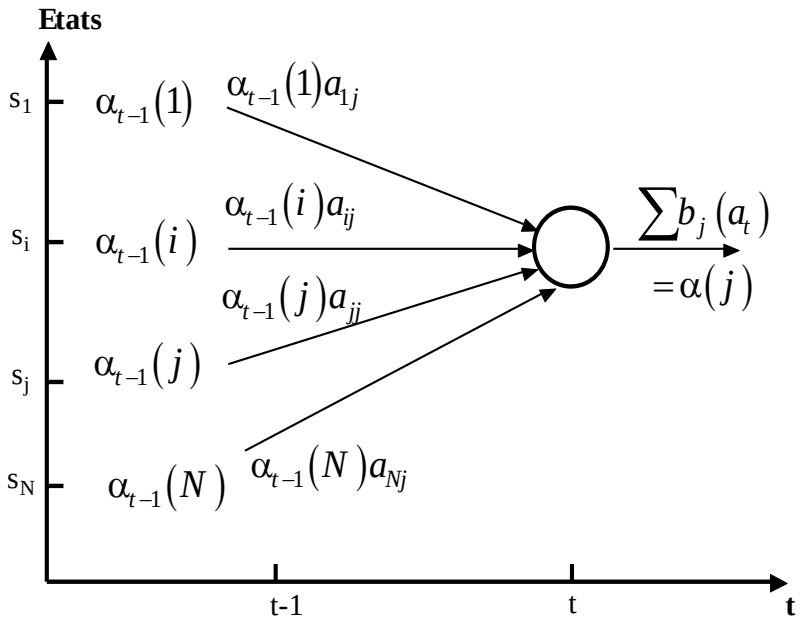

Figure II.2 : Suite partielle pour le calcul de α_t

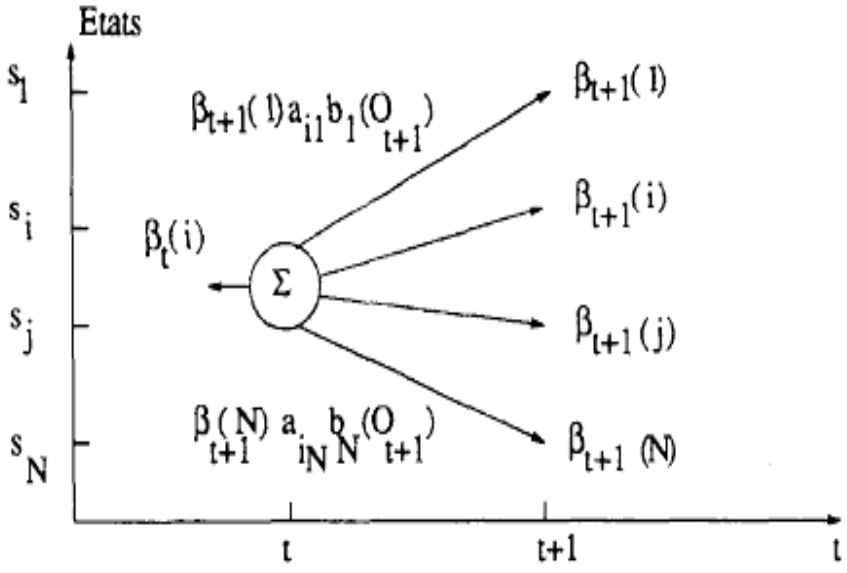

Figure II.3 : Suite partielle pour le calcul de β_t.

• Calcul de α

Soit la variable Forward $\alpha_t(j)$

$$\alpha_t(j) = p\left(O_1 O_2 ... O_t, q_t = s_i \middle| \lambda\right), \quad 1 \le j \le N, \quad 1 \le t \le T \qquad \text{(II.25)}$$

Algorithme Forward :

1. Initialisation, $t = 1$

$$\alpha_1(i) = \pi_i b_i(O_1), \quad i = 1, 2, ..., N \qquad \text{(II.26)}$$

Cette étape initialise la probabilité Forward. C'est la probabilité conjointe de l'état s_i, $i = 1, 2, \ldots, N$ et l'observation initiale O_1.

2. Induction

$$\alpha_t(j) = \left[\sum_{i=1}^{N} \alpha_{t-1}(i) a_{ij}\right] b_j(O_t), \quad j = 1, 2, ..., N, \quad t = 2, 3, ..., T \qquad \text{(II.27)}$$

Cette étape montre comment l'état s_j peut être visité au temps $t+1$ à partir de N états possibles s_i, $1 < i < N$ au temps t.

3. Terminaison

$$p\left(O \middle| \lambda\right) = \sum_{j=1}^{N} \alpha_T(j) \qquad \text{(II.28)}$$

Pour calculer la probabilité de l'observation par cette méthode $N(N+1)(T-1)+N$ multiplications et $N(N-1)(T-1)$ additions soit environ N^2T opérations sont effectuées. Par exemple, pour $N = 5$ et $T = 100$ on obtient environ 3000 opérations au lieu de 10^{72} opérations demandées par la méthode directe.

Toutes les transitions possibles entre les états peuvent être représentées sous forme de treillis, voir (figure II.4). Puisqu'il existe seulement N états (un noeud à chaque instant

t), toutes les suites possibles d'états se fusionnent dans ces *N* noeuds quelque soit la longueur des suites d'observations.

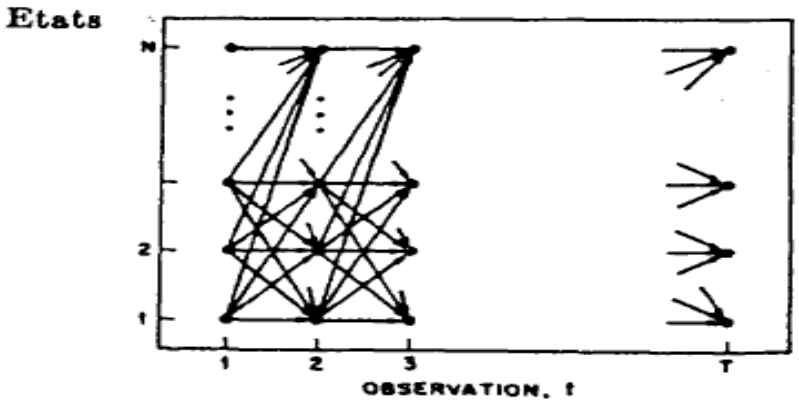

Figure II.4 : Implémentation du calcule de $\alpha_t(i)$ *ou*, $\beta_t(i)$ sous forme de treillis.

• Calcul de β

Soit la variable Backward $\beta_t(i)$ définie par :

$$\beta_t(i) = p\left(O_{t+1}O_{t+2}...O_T \middle| q_t = s_i, \lambda\right), \quad 1 \le i \le N, \quad T \le t \le 1 \tag{II.29}$$

Algorithme Backward :

1. Initialisation, (t = T)

$$\beta_T(i) = 1, \quad i = 1, 2, ..., N \tag{II.30}$$

Cette étape définie arbitrairement $\beta_T(i) = 1$ pour tous les états i.

2. Induction

$$\beta_T(i) = \sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j), \quad i = 1, 2, ..., N, \quad t = T-1, T-2, ..., 1 \tag{II.31}$$

Pour être dans l'état s_i au temps t, et pour tenir compte de la suite d'observation de *t+1* à T, nous devons considérer tous les états possibles s_j (toutes les transitions a_{ij}) aussi bien que l'observation O_{t+1} dans l'état j (les $b_j(O_{t+1})$), puis de tenir compte de la suite d'observations partielle restante à partir de l'état j ($\beta_{t+1}(j)$).

Pour calculer la probabilité $p(O|\lambda)$ par cette méthode $N(N+1)(T-1)+N$ multiplications et $N(N-1)(T-1)$ additions soit environ N^2T opérations sont effectuées. De même que l'algorithme Forward, toutes les transitions possibles entre les états peuvent être représentées sous forme de treillis, voir (figure II.4).

Les deux variables $\alpha(i)$ et $\beta(j)$ peuvent être utilisées pour calculer $p(O|\lambda)$ à chaque instant t, avec $1 \le t \le T$:

$$p(O|\lambda) = \sum_{i=1}^{N} a_t(i)\beta_t(i) \tag{II.32}$$

$$p(O|\lambda) = \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1})\beta_{t+1}(j) \tag{II.33}$$

Cette formule sera utilisée pour résoudre le problème 3.

Remarques

La probabilité $p(O|\lambda)$ peut être calculée en posant $t = T$ dans l'équation (II.33) :

$$p(O|\lambda) = \sum_{j=1}^{N} \alpha_T(i) \tag{II.34}$$

Les variables α_t et β_t peuvent s'écrire sous forme matricielle :

Soit :

$$\alpha_t = \left[\alpha_t(1)\alpha_t(2)...\alpha_t(N)\right]' \tag{II.35}$$

Et :

$$\beta_t = \left[\beta_t(1)\beta_t(2)...\beta_t(N)\right]' \tag{II.36}$$

Alors :

$$\alpha_t = \beta_t A' \alpha_{t-1} \tag{II.37}$$

$$\beta_t = \begin{pmatrix} b_1(O_t) & & 0 \\ & b_2(O_t) & \\ 0 & & \ddots \\ & & b_N(O_t) \end{pmatrix} \tag{II.38}$$

$$\alpha_1 = B_1\Pi \tag{II.39}$$

$$\beta_t = AB_{t+1}\beta_{t+1} \tag{II.40}$$

$$\beta_T = 1 \tag{II.41}$$

$$p\left(O|\lambda\right) = \beta_t'\alpha_t \tag{II.42}$$

Cas spécial :

$$t = 1 : p_1\left(O|\lambda\right) = \Pi' B_1\beta_1 \tag{II.43}$$

$$t = T : p_T\left(O|\lambda\right) = 1'\alpha_T = 1'B_T A'B_{T-1}...A'B_1 \tag{II.44}$$

Où (‘) signifie la transposée d’une matrice.

Dans chacune de ces formules la probabilité *p* peut être vue comme la trace d’une matrice [1x 1] qui est un produit de différentes matrices. **Deuxième problème : Estimation de la suite cachée ‘decoding’**

Etant donné une suite d’observations $O_1O_2...O_T$, et un modèle λ, Comment peut-on choisir une suite d’états $Q = q_1 q_2 ... q_T$ qui soit optimale selon un critère convenable. La difficulté réside dans la définition de la suite optimale d’états, c’est-à-dire qu’il existe plusieurs critères d’optimalité possibles. Selon le choix du critère nous proposons trois solutions :

II.5.1.3. Estimation de l’état q_t indépendamment des autres états

Cette technique consiste à choisir l’état q_t qui est la plus probable et ceci indépendamment des autres états ; ce qui revient à choisir au temps *t* l’état qui maximise $p\left(q_t = s_i|O,\lambda\right)$. Ce critère d’optimalité permet de maximiser le nombre espéré des états indépendants. L’un des problèmes posés par Baum [18] était de calculer l’estimation de q_t pour $1< t < T$, basée sur la réalisation de la suite d’observations *O*.

Sous le critère de la probabilité d’erreur minimale, il serait nécessaire de déterminer soit la vraisemblance conjointe :

$$\vartheta_t\left(s_i\right) = p\left(O_1O_2...O_T, q_t = s_i|\lambda\right), \quad i = 1,2,...,N \tag{II.48}$$

Soit les probabilités a posteriori :

$$\tilde{\vartheta}_t\left(s_i\right) = p\left(q_t = s_i|O_1O_2...O_T,\lambda\right), \quad i = 1,2,...,N \ , \quad t = 1,2,...,T \tag{II.49}$$

$$= p\left(q_t = s_i \left| O_1 O_2 ... O_t O_{t+1} ... O_T , \lambda \right.\right)$$
$$= p\left(q_t = s_i , O_1 O_2 ... O_t \left| \lambda \right.\right) p\left(O_{t+1} ... O_T \left| q_t = s_i , \lambda \right.\right)$$

$$= \alpha_t(i)\beta_t(i) , \quad t = 1,2,...,T \tag{II.50}$$

On peut écrire :

$$\gamma_t(i) = \frac{\tilde{\vartheta}_t(s_i)}{p\left(O\left|\lambda\right.\right)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N}\alpha_t(i)\beta_t(i)} \tag{II.51}$$

Le facteur de normalisation $p\left(O\left|\lambda\right.\right)$ fait que :

$$\sum_{i=1}^{N}\gamma_t(i) = 1 \tag{II.52}$$

En utilisant ainsi $\gamma_t(i)$ nous pouvons estimer l'état individuel q_t le plus probable au temps t

$$q_t = arg\ Max_{1\le i\le N}\left[\gamma_t(i)\right] \tag{II.53}$$

Remarques

❋ Les variables α sont calculées et stockées de façon récursive. Elles sont utilisées ensuite pendant l'étape de régression "Backward" pour calculer

$$\tilde{\vartheta}_t(s_i) = \alpha_t(i)\beta_t(i) , \quad i=1,2,..., N \text{ et } t = T,\ T-1,\ldots,1 \tag{II.54}$$

❋ Il est possible de résoudre le problème 1 vu précédemment par la formule suivante:

$$p\left(O\left|\lambda\right.\right) = \sum_{i=1}^{N}\alpha_T(i) = \sum_{i=1}^{N}\tilde{\vartheta}_t(i) \tag{II.55}$$

Bien que l'équation (II.53) maximise le nombre espéré des états individuels en sélectionnant l'état le plus vraisemblable à chaque instant, cependant on peut avoir quelques problèmes relatifs à la suite d'états produite par cette équation. Ainsi, quand le HMM possède des transitions d'états nulles pour certains états *i* et *j* (a_{ij} = 0), la suite d'états optimale estimée dans ce cas n'est pas valide. Ceci est dû à la solution de l'équation (II.48) qui détermine l'état le plus vraisemblable à chaque instant sans prendre en compte la probabilité d'occurrence des suites d'états.

II.5.1.4. Prise en compte des transitions deux à deux ou trois à trois entre les états

Dans certaines applications, nous choisissons des états qui ont le plus de chance deux à deux ou trois à trois. L'inconvénient de cette approche est qu'une partie des contraintes de transitions entre états ne sera pas prise en compte [17] .

II.5.1.5. Algorithme de Viterbi

Le critère le plus utilisé est celui de trouver l'unique trajectoire optimale de la suite d'états, c'est-à-dire Maximiser $p\left(Q|O,\lambda\right)$ ou Maximiser $p\left(Q,O|\lambda\right)$. Une technique formelle pour trouver le chemin optimal est basée sur les méthodes de programmation dynamique, c'est l'algorithme de Viterbi [19]

C'est un Algorithme récursif qui permet de trouver à partir d'une suite d'observations provenant d'un canal sans mémoire, une solution optimale au problème d'estimation de la suite d'états d'un processus de Markov à temps discret qui produit cette suite d'observations. Pour trouver une trajectoire unique et optimale de la suite d'états, $Q = q_1 q_2 \ldots q_T$ produisant la suite d'observations $O = O_1O_2...O_T$, nous définissons la quantité

$$\delta_t\left(j\right) = Max_{q_1q_2...q_{t-1}}\ ln\ p\left(q_1q_2...q_t = i, O_1O_2...O_t|\lambda\right), \quad t \geq 2 \qquad \text{(II.56)}$$

Qui représente le meilleur score (la probabilité maximale) correspondant à une trajectoire unique jusqu'au temps *t* et qui prend en compte les premières "t-observations" et s'arrête à l'état s_i. Par itération :

$$\delta_t\left(j\right) = Max_{1\leq i\leq N}\left[\delta_{t-1}\left(i\right) + ln\, a_{ij}\right] ln\, b_j\left(O_t\right), \quad 1 \leq j \leq N \qquad \text{(II. 57)}$$

Pour retrouver la suite optimale d'états, nous devons garder une trace des arguments qui maximise l'équation (II.57) pour chaque *t* et *j*.

II.5.1.5.1. Principe de l'Algorithme de Viterbi

Soit la suite d'observation $O = O_1O_2...O_T$, comment trouve-t-on une suite d'états $Q = q_1 q_2 \ldots q_T$ qui soit optimale en certain sens?

La réponse à cette question consiste à maximiser la probabilité conjointe $p\left(O,Q\right)$*:*

$$Max_Q\ ln\ p\left(O,Q\right) \Rightarrow Q_{optimal} \qquad \text{(II.58)}$$

$$p\left(O,Q\right) = p\left(Q\right)p\left(O|Q\right)$$

$$= p(q_1 = s_l)\, p(O_1 | q_1 = s_l) \prod_{t=2}^{T} p(q_t = s_j | q_{t-1} = s_i) \prod_{t=2}^{T} p(O_t | q_t = s_j)$$

$$= \pi_l b_l(O_1) \prod_{t=2}^{T} a_{ij} b_j(O_t) \qquad \text{(II.59)}$$

Avec : $1 \le l \le N$, $1 \le i \le N$, $1 \le j \le N$

On a alors,

$$ln\ p(O, Q) = ln\left(\pi_1 b_l(O_1)\right) + \sum_{t=2}^{T} \delta(q_t = s_i) \qquad \text{(II.60)}$$

Qui représente le coût total pour le chemin *Q*, où S est le coût d'un segment (une transition d'un état à un autre) de chemin *Q* :

$$\delta(q_t = s_j) = ln\, a_{ij} + ln\, b_j(O_t) \qquad \text{(II.61)}$$

Nous définissons $\psi_t(j)$ comme étant le chemin le plus court correspondant au nœud $q_t = s_i$ (survivant). A chaque instant *t*, il existe N survivants (un pour chaque noeud).

L'algorithme nécessite, à chaque instant t, la mémorisation de ces N survivants ainsi que leurs coûts.

Algorithme

1-Initialisation, t=1

Si q_1 est connu a priori, alors :

$$\delta_1(i) = 0, \forall i \quad \text{(Coût du survivant i)} \qquad \text{(II.62)}$$

$$\psi_t = i \quad \text{(Cette variable stocke l'état optimal à l'instant } t\text{)} \qquad \text{(II.63)}$$

Autrement, Si q_1 est inconnu a priori

Alors :

$$\delta_1(i) = ln\left(\pi_i b_i(O_1)\right), \quad i = 1, 2, ..., N \qquad \text{(II.64)}$$

$$\psi_i = 0 \qquad \text{(II.65)}$$

2-Induction

$$\delta_1(j) = Max_{1 \le i \le N}\left[\delta_{t-1}(i)\right] b_j(O_t), \quad 1 \le j \le N, \quad 2 \le t \le T \qquad \text{(II.66)}$$

$$\psi_t(j) = arg\ Max_{1 \le i \le N}\left[\delta_t(i) + ln\, a_{ij}\right], \quad 1 \le j \le N, \quad 2 \le t \le T \qquad \text{(II.67)}$$

3- Terminaison

$$ln\ p^* = Max_{1\le i\le N}\left[\delta_t(i)\right] \quad \text{(II.68)}$$

$$q_T^* = arg\ Max_{1\le i\le N}\left[\delta_t(i)\right] \quad \text{(II.69)}$$

Chemin obtenu "Backtracking"

$$q_t^* = \psi_{t+1}\left(q_{t+1}^*\right),\quad t = T-1, T-2, ..., 1 \quad \text{(II.70)}$$

II.5.2. Problème 3 : Optimisation des paramètres du modèle (Apprentissage)

Comment peut-on ajuster les paramètres du modèle λ =(Π,A,B) pour maximiser $p\left(O_t|\lambda\right)$*?.*

Le fait que la longueur de la suite d'observations (données d'apprentissages) est finie, il n'existe pas de solutions analytiques directes (d'optimisation globale) pour construire le modèle.

Néanmoins, nous pouvons choisir $\lambda = (\Pi, A, B)$ tel que $p\left(O_t|\lambda\right)$est un maximum local en utilisant une procédure itérative telle que celle de BAUM-WELCH [18] [20] (ou d'une façon équivalente l'algorithme d'identification de mélange de type EM (E pour Expectation, M pour Maximisation) [21] ou en utilisant les techniques de gradient telle que la méthode de Liporace [22].

L'idée de l'application est donc d'utiliser des procédures de ré-estimation qui affinent le modèle petit à petit en suivant les étapes suivantes :

- Choisir un ensemble initial de paramètres λ_0;
- Calculer λ_1 à partir de λ_0;
- Répéter ce processus jusqu'à un critère de fin.

II.5.2.1. Méthode de BAUM-WELCH

Basée sur l'estimation par le maximum de vraisemblance, "Maximum Likelihood (MLE)"

Soit :

$$\zeta_t(i,j) = p\left[q_t = s_i, q_{t+1} = s_j \middle| O, \lambda\right],\quad t = 1,2,3,...,T-1 \quad \text{(II.71)}$$

La probabilité de visiter l'état s_i au temps t et l'état s_i au temps $t+1$, sachant le modèle et la suite d'observations $O = O_1O_2...O_T$, voir (figure II.5). Nous pouvons écrire :

$$\zeta_t(i,j) = \frac{p\left[q_t = s_i, q_{t+1} = s_j \middle| O, \lambda\right]}{p(O|\lambda)}, \quad t = 1,2,3,...,T-1 \tag{II.72}$$

$$= \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \tag{III.73}$$

Et on peut démontrer que la formule (II.51) peut être écrite :

$$\gamma_t(i) = \sum_{j=1}^{N} \zeta_t(i,j), \quad t = 1,2,...,T-1 \tag{II.74}$$

On peut remarquer que le nombre espéré des transitions à partir de s_i est donné par la formule suivante :

$$\gamma_i = \sum_{t=1}^{T-1} \gamma_t(i) \tag{II.75}$$

$$= \sum_{j=1}^{T-1} \gamma_{ij} \tag{II.76}$$

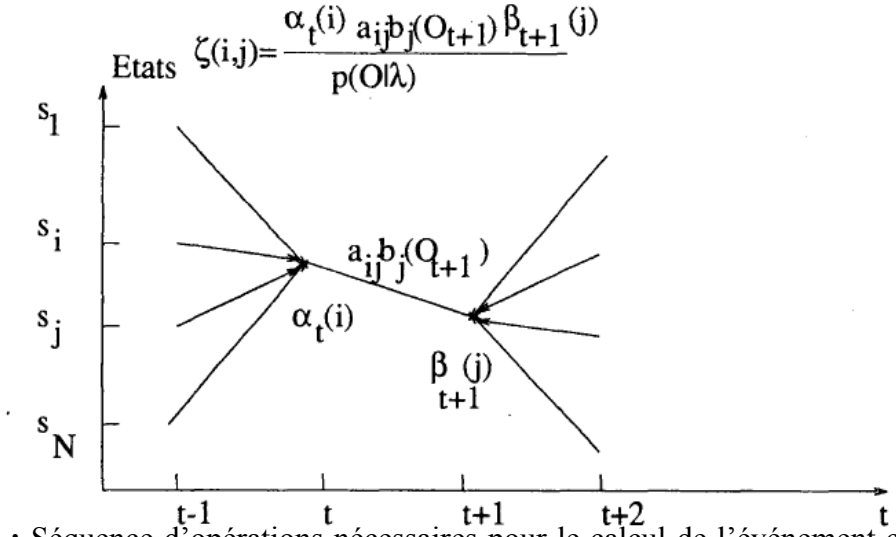

Figure II.5 : Séquence d'opérations nécessaires pour le calcul de l'événement conjoint pour que le système soit à l'état s_i au temps t et à l'état s_j au temps t + 1.

Où :

$$\gamma_{ij} = \sum_{t=1}^{T-1} \zeta_t(i,j) \tag{II.77}$$

$$= \frac{\sum_{t=1}^{T-1} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad \text{(II.78)}$$

γ_{ij} est le nombre espéré des transitions de s_i vers s_j.

Cette méthode de Maximum de Vraisemblance est la plus utilisée dans les applications.

L'algorithme de Baum-Welch

Cet algorithme peut être représenté sous la forme itérative suivante:

1-Fixer des valeurs initiales, *k*=0

$$\lambda = \left\{\pi_i^0, a_{ij}^0, b_j^0(O_t)\right\},\ 1 \le i \le N, 1 \le j \le N \quad \text{(II.79)}$$

2-Calculer

$$\zeta_t(i,j) \text{ et } \gamma_t(i) \text{ pour } 1<i<N,\ 1<j<N \quad \text{(II.80)}$$

En utilisant les fonctions Forward et Backward.

3-Nouvelles estimations, *k*=1,2,3, ...

$$\tilde{\lambda} = \left\{\tilde{\pi}_i^k, \tilde{a}_{ij}^k, \tilde{b}_j^k(O_t)\right\}, \text{ pour } 1 \le i \le N, 1 \le j \le N \quad \text{(II.81)}$$

4-Recommencer en 2 jusqu'à un certain point limite. [17]

II.8.Conclusion

Les Modèles de Markov Cachés (HMM), présentés dans ce chapitre sont des techniques les plus utilisés en reconnaissance de la parole, ils bénéficient d'algorithmes d'entraînement et de décodage performants.

En traitement de la parole les HMM utilisés sont particulièrement d'ordre 1.

Nous avons décrit, les différents algorithmes pour résoudre les trois problèmes fondamentaux d'un HMM qui sont : l'évaluation d'un modèle, estimation de la suite d'états cachés et l'apprentissage.

CHAPITRE III
Résultats Et Interprétations

III.1. Introduction

Ce chapitre est consacré à la présentation des différents résultats que nous avons obtenus en appliquant les méthodes qui ont fait l'objet de ce mémoire.

Notre objectif est dans un premier temps de construire une base de données sous forme de fichiers wav, puis de lui appliquer les commandes du logiciel HTK (Hidden Markov Model Toolkit), afin de calculer le taux de reconnaissance d'un mot ou d'une phrase test avec les mots ou expressions décrits dans la base de données.

Ainsi, nous avons exploité les fonctionnalités offertes par le logiciel HTK (Hidden Markov Model Toolkit), afin de concevoir notre système de reconnaissance. Pour cela, nous avons suivi les étapes suivantes :

• Création de l'ensemble d'apprentissage : Chaque élément du vocabulaire est enregistré plusieurs fois et étiqueté avec le mot correspondant sur le phonème.

• Analyse acoustique : les signaux enregistrés sont convertis en une série de vecteurs de caractéristiques (coefficients MFCC, LPC ou PLP).

• Définition des modèles HMM : un prototype HMM est construit pour chaque élément du vocabulaire des tâches de reconnaissance.

• Modèles de formation : chaque HMM est initialisé et formé avec l'ensemble d'apprentissage correspondant.

• Définition de la tâche de reconnaissance : la grammaire à suivre est définie.

• Reconnaissance et évaluation des performances sur un corpus de test.

III.2. Organisation de la base de données

III.2.1. Organisation du corpus

Le corpus comprend les dix premiers chiffres de l'arabe standard de 0 à 9 et 20 mots syntaxiquement et sémantiquement correctes. Les mots de notre corpus ont été vérifiées par des linguistes de l'Université de Laghouat.

1. نَشْكُرُكَ.
2. استِعْمَالِكَ.
3. خِدْمَة.
4. الدَّفْعِ.

5. إِجَابَتُكَ.
6. خَاطِئَةٌ.
7. يُرْجَى.
8. إِعَادَة.
9. المُحَاوَلَةْ.
10. الحُصولُ.
11. اللُّغَةِ.
12. العَرَبِيَّةِ.
13. اضْغَطْ.
14. الرَّقْمِ.
15. وَاحِدْ.
16. مَرْحَبًا
17. تَعْبِئَة.
18. حِسَابِكَ.
19. الانْتِظار.
20. البَحْث.

III.2.2. Identification et critères de choix du locuteur

Suivant le protocole TIMIT (Texas Instruments Massachusetts Institute of Technology) pour mieux gérer les locuteurs et pour une meilleure organisation de la base de données, chaque locuteur doit avoir un code unique et ce selon plusieurs critères (sexe W / M, adulte / enfant, niveau de l'éducation).

Les locuteurs doivent avoir une bonne prononciation de l'arabe standard et une bonne qualité vocale.

Nos locuteurs sont des étudiants de l'Université de Laghouat et d'autres locuteurs.

III.3. Phase d'enregistrement du corpus

III.3.1. Conditions d'inscription

Nous devons définir correctement tous les paramètres :

• Le lieu d'enregistrement doit être un lieu normal.

• Chaque enregistrement doit commencer et se terminer par un silence.

• Recommencez l'enregistrement s'il a été interrompu.

• La distance entre le locuteur et le capteur doit être ajustée en fonction de la voix de locuteur.

• Vérifiez chaque fois l'enregistrement pour éviter la saturation ou que le signal est faible avec un éditeur de signal dans notre cas nous avons utilisé Wave Editor.

III.3.2. Manipulation d'enregistrement

Le débit de lecture doit être normal (ni rapide ni lent). Il est également nécessaire de corriger le locuteur si le niveau d'émission tend à s'affaiblir (tendance à parler de moins en moins fort).

Donc :

• Nos locuteurs sont des étudiants de l'Université de Laghouat et d'autres locuteurs ;

• Paramètres d'entrée : fréquence d'échantillonnage (16 KHz), nombre de canaux (1), codés sur 16 bits ;

• Le lieu d'enregistrement est un endroit normal avec du bruit ;

• Le corpus se compose des dix premiers chiffres de l'arabe classique de 0 à 9 et 20 mots prononcées par 12 locuteurs qui ont répété ces mots 10 fois (dans notre cas 1200 pour les chiffres et 2400 pour les mots) pour avoir un bon taux de reconnaissance (tableau III.1).

Corpus		Locuteur	Nombre d'enregistrement	Total
Chiffres 0-9	0-9	12	10	1200
Mots	20	12	10	2400

Tableau III.1. Le corpus.

III.3.3. L'Acquisition des fichiers sons

On utilise le logiciel Wave Editor pour enregistrer nos fichiers sons.

Wave Editor est un outil très efficace qui nous a permis de lire un fichier son (le visualiser, l'écouter, le découper…) ou même en créer un nouveau, de faire une analyse acoustique (durées, Fo, intensité).

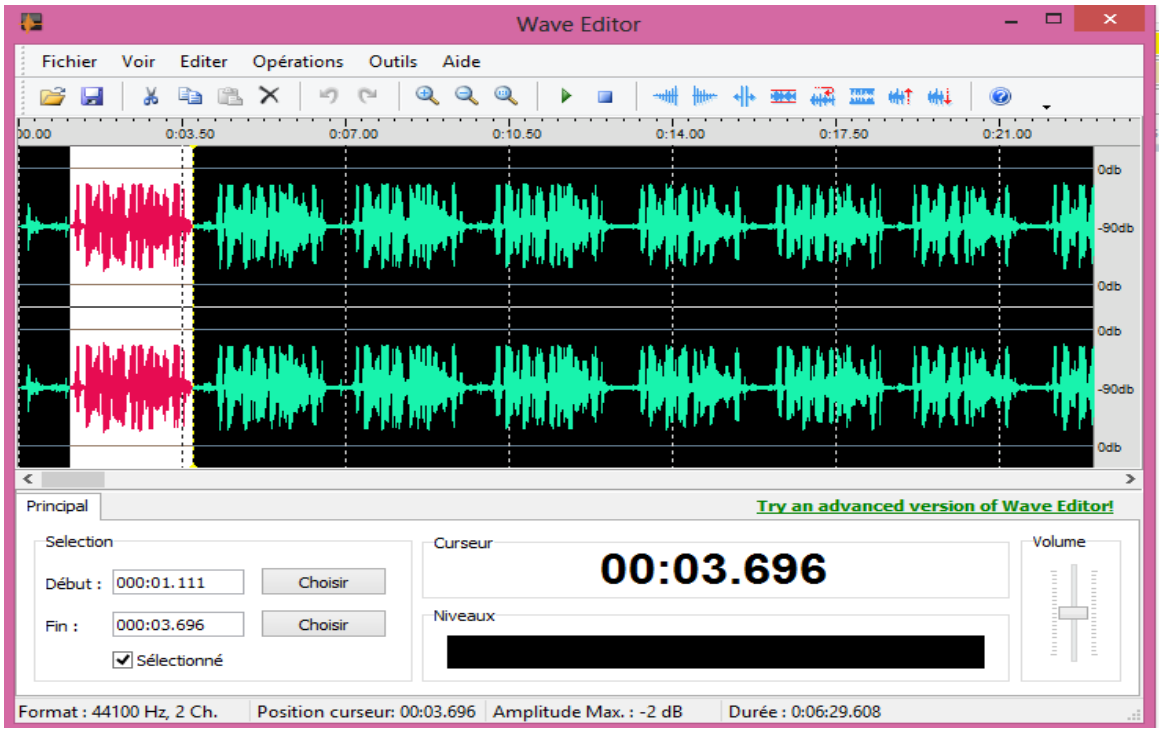

Figure III.1. Wave Editor.

Chaque élément du vocabulaire est enregistré, et étiqueté avec le mot correspondant.

Le résultat de cette phase sont les fichiers sons (.wav) représentants le vocabulaire.

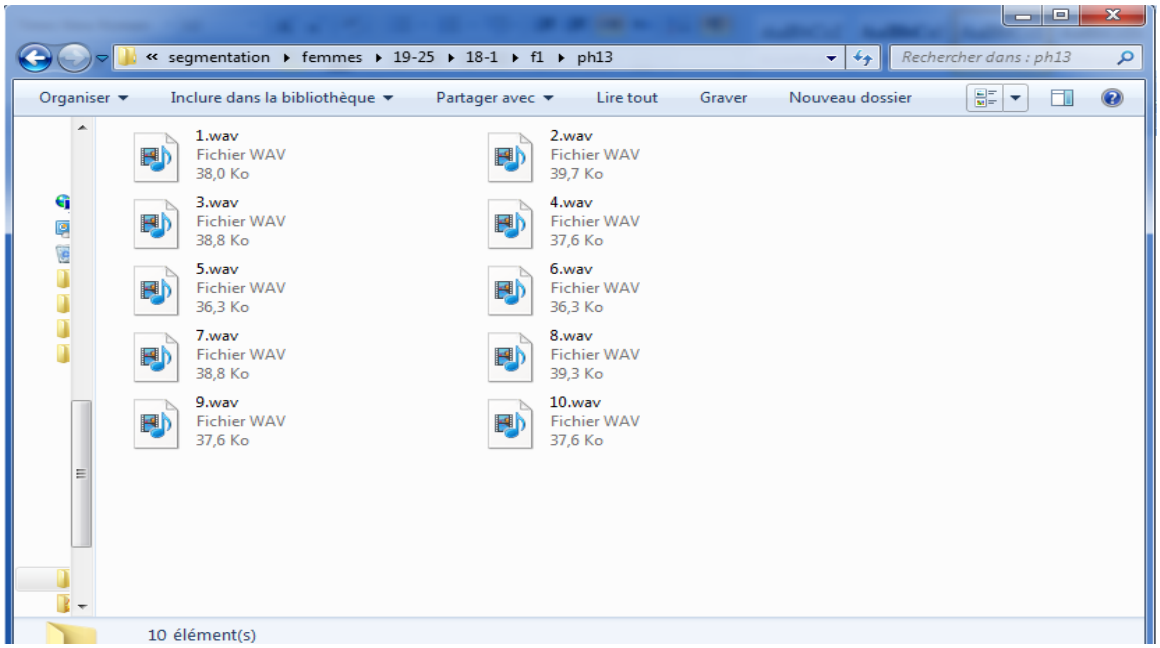

Figure III.2. Les fichiers sons de quelques enregistrements.

III.3.4. Segmentation et étiquetage

Nous avons utilisé l'outil Wave Editor pour faire l'étiquetage manuel du corpus et le système pour la transcription phonétique SAMPA (Speech Assessment Methods Phonetic Alphabet).

L'identification des différentes unités a été faite en utilisant l'outil Wave Editor, nous avons ainsi contrôlé, à chaque enregistrement la forme temporelle de l'onde acoustique correspondant à l'enregistrement, son spectrogramme, l'onde temporelle du pitch F_0 ainsi que son énergie en s'appuyant toujours sur l'écoute.

III.3.5. Transcription de corpus

La transcription orthographique et phonétique des mots de notre corpus est donnée ci-dessous :

Tableau III.2 : Transcription orthographique et phonétique des mots de notre corpus.

MOTS DE NOTRE CORPUS	LA TRANSCRIPTION ORTHOGRAPHIQUE	LA TRANSCRIPTION PHONETIQUE
نَشْكُرُكَ	NASHKURUKA	Na che ko ro ka
إِستِعْمَالِكَ	AISTIEMALIKA	Ie se ti aa ma li ka
خِدْمَةُ	KHIDMATO	Khi de ma to
الدَّفْعِ	ALDDAFEI	Aa le da fe ei
إِجَابَتُكَ	IJABATUKA	Ai ja ba to ka
خَاطِئَةْ	KHATIA	Kha ti aa tuu
يُرْجَى	YURJAA	Yo re ja aa
إِعَادَة	'IIEADATO	Ai aa da to
المُحَاوَلَةْ	ALMUHAWALAH	Aa le mo ha wa la to
الحُصول	ALHUSWL	Aa le ho so le

اللُّغَةِ	ALLUGHATO	Aa le gha ti
العَرَبِيَّةِ	ALEARABIATO	Aa le aa ra be ya to
اضْغَطْ	AIDGHAT	Ai de gha te
الرَّقْمِ	ALRAQM	Aa le ra ke mi
وَاحِدْ	WAHID	Wa aa hi de
مَرْحَبًا	MARHABANA	Ma re ha ba
تَعْبِئَةِ	TAEBIA	Ta ee bi aa to
حِسَابِكَ	HISABIKA	Hi sa be ka
الانْتِظار	ALANTIZAR	Aa le ei ne ti da re
البَحْث	ALBAHTH	Aa le ba he the

III.3.6. Création du dictionnaire

L'utilisation d'un dictionnaire est indispensable pour la mise en œuvre du système de reconnaissance à partir de la base de données. L'Exemple ci-dessous représente quelques mots et leur phonétique qui ont été utilisés dans notre travail :

Tableau III.3 : Transcription orthographique et phonétique des phrases de notre corpus

LES MOTS	LA PHONETIQUE CORRESPONDANTE
IDGHAT	Ai De GHa Te sp
IJABATOUKA	Ai ja: ba tu ka sp
YOURJA	yu re ja: sp
WAHID	wa: Hi De sp

III.4. Analyse acoustique

Après avoir acquis les fichiers sons, on va convertir les signaux enregistrés en une série de vecteurs de traits pertinents (coefficients MFCC) ou les paramètres du codage sont donnés comme suit :

```
# Coding parameters
SOURCEKIND = WAVEFORM          # Type de source de paramètre
SOURCEFORMAT = WAV             # Le format de fichier
TARGETKIND = MFCC_0_D_A        # Identifier les coefficients utilisés
TARGETRATE = 100000.0          # la période 100ms
SAVECOMPRESSED = T             # Compression du fichier
SAVEWITHCRC = T                # Attach a checksum to output parameter file
WINDOWSIZE = 250000.0          # Taille de la fenêtre d'analyse en unités de 100 ms
USEHAMMING = T                 # Utiliser la fenêtre de Hamming
PREEMCOEF = 0.97               # Coefficient préaccentuation
NUMCHANS = 26                  # Nombre des bancs de filtres
CEPLIFTER = 22                 # Longueur du filtre Cepstral
NUMCEPS = 12                   # Nombre de coefficiente MFCC
```

Figure III.3. Fichier Config (Cas MFCC).

Le nombre de coefficients utilisé est 13.

Le résultat est un ensemble de fichiers MFC (figure III.2)

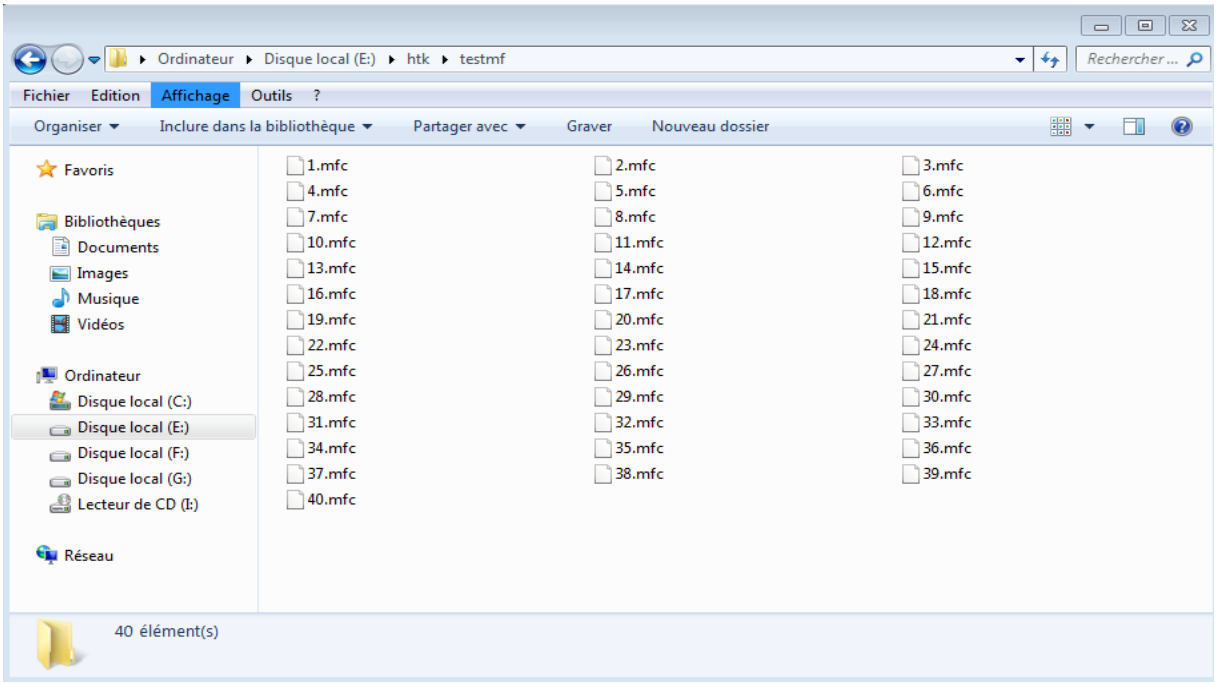

Figure III.4. Les fichier MFCC.

III.5. Apprentissage

Nos modèles doivent être appris : les moyennes, les variances et les probabilités de transition entre états sont réestimées jusqu'à atteindre un seuil de convergence ou un nombre maximum d'itérations.

Nous avons commencé par créer un prototype d'initialisation HMM dans un dossier appelé hmm0, puis l'apprentissage a été fait en utilisant la commande **HERest** qui est basée sur l'algorithme Baume-Welch pour réestimer tous les modèles à la fois.

Le résultat final de cette opération est le suivant:

```
~o
<STREAMINFO> 1 39
<VECSIZE> 39<NULLD><MFCC_D_A_0><DIAGC>#méthode MFCC
~s "silst"
<MEAN> 39    #la Moyenne
 -4.132289e+000 -5.852334e+000 -1.182486e+001 -1.602135e+001 -5.530322e+000 -
1.013116e+001 -4.129276e+000 -8.462726e+000 -1.056108e+000 -4.361499e+000 -
1.292784e+000 -1.613368e+000 5.496589e+001 -4.364110e-003 3.797082e-003
4.918423e-004 4.927451e-003 1.383462e-004 6.337059e-003 5.139787e-003 -2.571158e-
003 4.153035e-003 1.240190e-003 -1.165723e-003 7.066102e-003 -5.747891e-003 -
2.135548e-003 -6.577860e-004 1.448067e-004 6.605370e-004 -2.252217e-003 -
2.579721e-003 -2.768276e-003 -4.548113e-004 1.049615e-003 3.816137e-004 2.228063e-
003 -9.661466e-004 -1.031038e-003
<VARIANCE> 39     #la variance
 7.778294e+001 1.396601e+002 6.927835e+001 1.371290e+002 8.358485e+001
1.015789e+002 6.726229e+001 6.080323e+001 5.830947e+001 5.770854e+001
5.547543e+001 5.200193e+001 1.709822e+002 2.974011e+000 4.372619e+000
3.726609e+000 4.679754e+000 4.827283e+000 5.850727e+000 4.679337e+000
4.199592e+000 4.306425e+000 3.723794e+000 3.777670e+000 3.239922e+000
4.445760e+000 4.630888e-001 6.461843e-001 6.337044e-001 7.490763e-001 8.713586e-
001 1.001192e+000 8.887562e-001 7.856687e-001 8.359959e-001 7.061987e-001
6.994851e-001 6.083365e-001 6.023161e-001
<GCONST> 1.429011e+002
~h "Da"
<BEGINHMM>
<NUMSTATES> 5     #nombres d'états
<STATE> 2
<MEAN> 39
9.277765e-001 -1.443147e+001 -2.142343e+001 -1.319079e+001 -5.637074e+000
 .
 .

<ENDHMM>
```

Figure III.5. Fichier prototype

III.6. Reconnaissance

- Nous devons d'abord définir le modèle de notre langage ; la grammaire de notre langue est définie dans le fichier gram.txt

```
$name= IDGHAT |IJABATOUKA|YOURJA |WAHID ;
(SENT-START (<$name>) SENT-END)
```

Figure III.6. Fichier gram des 4 mots.

- Ensuite, nous générons le modèle des mots à partir de la grammaire. **HTK** nécessite en fait un réseau de mots, d'être défini à l'aide d'une notation de bas niveau dans

laquelle chaque exemple de mot et chaque transition mot par mot est énuméré explicitement. Ce réseau de mots peut être créé automatiquement à partir de la grammaire utilisée à l'aide de la commande **HParse**. Le modèle de mot présent dans le fichier **wdnet**:

```
VERSION=1.0# N=num-nœuds, L=num-arcs W=mot
N=9          L=15
I=0          W=SENT-END
I=1          W=WAHID
I=2          W=!NULL
I=3          W=YOURJA
I=4          W=IJABATOUKA
I=5          W=IDGHAT
.
.
# List arcs : J=num-arc, S=start-nod, E=end-nod
J=0     S=2     E=0
J=1     S=2     E=1
J=2     S=6     E=1
J=3     S=1     E=2
J=4     S=3     E=2
J=5     S=4     E=2
J=6     S=5     E=2
J=7     S=2     E=3
.
.
```

Figure III.7. Fichier Wdnet.

- Un fichier dict.txt contient le dictionnaire de chaque mot de notre corpus :

```
IDGHAT          Ai De Ga Te sp
IJABATOUKA      Ai ja: ba tu ka sp
SENT-END        [ ] sil
SENT-START      [ ] sil
YOURJA          yu re ja: sp
WAHID           wa: Hi De sp
```

Figure III.8. Fichier dict des 4 mots.

- Une reconnaissance sera ensuite effectuée sur chaque dossier pour évaluer l'apport des différents apprentissages. Pour cela nous utiliserons la commande HVite pour lancer la reconnaissance.
- Le résultat est un fichier "recout" pour chaque fichier à reconnaître.

III.7. L'évaluation des performances

Cette évaluation se fera avec la commande **HResults**.

Pour évaluer la fiabilité de notre système de reconnaissance, nous avons effectué trois tests pour la méthode d'analyse MFCC.

Les résultats de la reconnaissance des mots obtenus pour la méthode analytique utilisée sont présentés dans cette section.

III.7.1.Résultats du premier test

Le premier test consiste à reconnaître les mêmes enregistrements pour les deux locuteurs qui ont été utilisés dans l'apprentissage.

Le taux de reconnaissance des mots 100%:

```
E:\htk>HResults -I testref.mlf tiedlist recout.mlf
====================== HTK Results Analysis =======================
  Date: Thu Jun 25 18:22:15 2020
  Ref : testref.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=100.00 [H=40, S=0, N=40]
WORD: %Corr=100.00, Acc=100.00 [H=40, D=0, S=0, I=0, N=40]
===================================================================
```

Figure III.9. Résultats du premier test.

Avec :

WORD : Mots,

% : Le taux obtenu en pourcentage.

N : le nombre total d'éléments à reconnaître,

D : Le nombre d'éléments non pris,

S : le nombre d'éléments non reconnus,

H : Le nombre d'éléments reconnus,

I : nombre d'éléments insérés,

ACC : Précision = (H-I) / N × 100%,

III.7.2.Résultats du deuxième test

Pour le deuxième test, nous avons modifié les enregistrements des tests et conservé les mêmes locuteurs.

Le taux de reconnaissance des phrases est de 95% et des mots 100% :

```
E:\htk>HResults -I testref.mlf tiedlist recout.mlf
====================== HTK Results Analysis =======================
  Date: Thu Jun 25 21:27:56 2020
  Ref : testref.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=97.50 [H=39, S=1, N=40]
WORD: %Corr=97.50, Acc=97.50 [H=39, D=0, S=1, I=0, N=40]
===================================================================
```

Figure III.10. Résultats du deuxième test.

III.7.3.Résultats du troisième test

Pour le troisième test, la reconnaissance a été effectuée pour six locuteurs différents.

Le taux de reconnaissance des phrases est de 85,56% et des mots 96,25% :

```
E:\htk>HResults -I testref.mlf tiedlist recout.mlf
====================== HTK Results Analysis =======================
  Date: Thu Jun 25 22:21:01 2020
  Ref : testref.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=87.50 [H=35, S=5, N=40]
WORD: %Corr=87.50, Acc=87.50 [H=35, D=0, S=5, I=0, N=40]
===================================================================
```

Figure III.11. Résultats du troisième test.

III.8. Synthèse des résultats trouvés

Le présent document vise à mettre en évidence la pratique de HTK pour construire un système de reconnaissance des mots et de phrases (mots connectés) ; la méthode d'analyse MFCC a été utilisée.

Les résultats des trois tableaux montrent que la méthode d'analyse du MFCC a de bonnes performances, par exemple les taux de reconnaissance obtenus dans le tableau IV.4 qui représente la reconnaissance des mêmes enregistrements pour les deux locuteurs qui ont été utilisés dans l'apprentissage, sont de 100% pour les mots (WORD).

Par contre, les résultats du tableau IV.5 où nous avons modifié les enregistrements des tests et nous avons conservé les mêmes locuteurs sont de 97,50% pour les mots (WORD) tandis que les taux de reconnaissance dans le tableau IV.6 qui a été fait pour 6 locuteurs soit 87,50% pour les mots (WORD).

III.9. Conclusion

Pour évaluer la fiabilité de notre système de reconnaissance, nous avons effectué trois tests pour la méthode d'analyse MFCC.

Le premier test consiste à reconnaître les mêmes enregistrements pour les deux locuteurs qui ont été utilisés dans l'apprentissage, pour le deuxième test, nous avons modifié les enregistrements de test et nous avons conservé les mêmes locuteurs et le troisième test, la reconnaissance a été faite pour trois locuteurs.

Les résultats obtenus nous permettent de conclure que l'analyse MFCC est une méthode très efficace ainsi que la méthode HMM qui nous donnent de bons résultats. Cela confirme la robustesse de notre système.

Conclusion Générale

La parole est le principal moyen de communication dans toute société humaine, et certainement le moyen le plus naturel de communication. Pour autant il est tout aussi certain qu'il est plus facile, de point de vue sémantique. Il constitue un défi pour les chercheurs dans le développement des systèmes de reconnaissance de la parole, pour faciliter la communication homme / machine et permettre la manipulation des machines en langage naturel.

Le travail que nous avons développé ici s'inscrivent dans un objectif de mise en place d'un système de dialogue homme-machine en général. Il s'agit de la reconnaissance de la parole dans le domaine de la communication.

Dans le cadre de cette thèse, nous nous sommes particulièrement intéressés à la reconnaissance automatique de la parole Arab. Nous avons utilisé les techniques d'analyse MFCC et LPC pour la reconnaissance automatique. Notre choix s'est également porté sur les outils les plus adaptés (modèles Markov cachés, HTK, ...) pour l'RAP.

Notre contribution à ce sujet peut se résumer en plusieurs points :

- ✓ L'acquisition et le développement d'une base de données composée d'un corpus de mots syntaxiquement et sémantiquement correctes de l'arabe classique. Ils ont été évalués par des linguistes de l'Université de Laghouat.
    - ✓ Analyse acoustique des signaux audio et extraction des paramètres pertinents par la technique ; coefficients cepstraux à l'échelle du mel (MFCC).
    - ✓ Une modélisation statistique par l'utilisation des modèles HMM pour la reconnaissance de la parole.
    - ✓ Développement d'un système de référence pour l'RAP basé sur la modélisation HMM sous HTK
    - ✓ Les résultats obtenus nous ont permis de conclure que :

L'analyse MFCC est une méthode très efficace ainsi que la méthode HMM qui nous donnent de bons résultats. Cela confirme la robustesse de notre système.

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.

RÉFÉRENCES BIBLIOGRAPHIQUE

[1] Radha, V., and C. Vimala. "A review on speech recognition challenges and approaches." doaj. org 2.1 (2012): 1-7.

[2] Singh, Lalima. "Speech signal analysis using FFT and LPC." International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) 4.4 (2015): 1658-1660.

[3] Souadkia, Abdelhak. Reconnaissance automatique de la parole arabe: Approche évolutionniste. Diss. 2010.

[4] Chapanis, Alphonse. "Interactive Communication: A Few Research Answers for a Technological Explosion." (1978).

[5] Bellik, Yacine. "Multimodal text editor interface including speech for the blind." Speech communication 23.4 (1997): 319-332.

[6] Oualid, M. D. Reconnaissance Automatique De La Parole Arabe Par Cmu Sphinx 4. Diss. Doctoral Dissertation, Université Ferhat Abbas de Sétif 1, 2013.

[7] H.delassi, F.benharzellah, Automatic recognition of the speech by LPC, MFCC Application to GSM signals, Université Amar Telidji- Laghouat.2019

[8] M.benberech, W.ouazene. reconnaissance de la langue arabe par les méthodes d'intelegences artificielles, Université Amar Telidji- Laghouat, 2018.

[9] Boite, René. Traitement de la parole. PPUR presses polytechniques, 2000.

[10] FOURATI, Mohamed, Lamia CHAARI, and Lotfi KAMOUN. "Analyse de la Qualité Sonore Issue du Codage de Canal du Standard GSM dans le Contexte de la Radio-Logicielle."

[11] Calliope , la parole et son traitement automatique , édition masson,1999

[12] Debyeche, Mohamed. Reconnaissance automatique de la parole appliquée à la langue arabe. Diss. 2007.

[13] Boite, R., and M. Kunt. "Traitement de la parole, édition." (1997).

[14] Ferguson, J. D. "Hidden Markov analysis: an introduction." Hidden Markov Models for Speech (1980).

[15] Rabiner, Lawrence R. "Mathematical foundations of hidden Markov models." Recent advances in speech understanding and dialog systems. Springer, Berlin, Heidelberg, 1988. 183-205.

[16] Kriouile, Abdelaziz. La reconnaissance automatique de la parole et les modèles markoviens cachés: modèles du second ordre et distance de Viterbi à optimalité locale. Diss. 1990.

[17] Poritz, Alan B. "Hidden Markov models: A guided tour." Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP). 1988.

[18] Baum, Leonard E. "An inequality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes." Inequalities 3.1 (1972): 1-8.

[19] VITERBI, ANDREW J. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm." The foundations of the digital wireless world: Selected works of AJ Viterbi. 2010. 41-50

[20] Baum, Leonard E., and John Alonzo Eagon. "An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology." Bulletin of the American Mathematical Society 73.3 (1967): 360-363.

[21] Dempster, Arthur P., Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." Journal of the Royal Statistical Society: Series B (Methodological) 39.1 (1977): 1-22.

[22] Liporace, L. "Maximum likelihood estimation for multivariate observations of Markov sources." IEEE Transactions on Information Theory 28.5 (1982): 729-734.