

République Algérienne Démocratique et Populaire.  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique.

Université Saâd Dahlab - Blida  
USDB

Faculté des sciences  
Département d'informatique

**Mémoire pour l'obtention  
d'un diplôme d'Ingénieur d'Etat en Informatique.**  
Option : Intelligence Artificielle

Sujet :

**Reconnaissance Automatique  
Phonémique de l'Arabe Standard par les  
Modèles de Markov Cachés (HMM)**

Présenté par : AMEL ALLICHE  
DARINE ABOULOZ

Promotrice : Dr M.GUERTI

Organisme d'accueil : Ecole Nationale Polytechnique d'ALGER

Soutenu le : 18 Février 2006, devant le jury composé de :

N. BENBLIDIA	CC	USD-BLIDA
M. GUERTI	MC	ENP-ALGER
F.Z. REGUIEG	CC	USD-BLIDA
Z.A. BENSELAMA	CC	USD-BLIDA

**Présidente**  
**Promotrice**  
**Examinatrice**  
**Examinateur**

- N° / Promotion 2005-2006

MIG-004-133-1



# Remerciements

Nous ne saurions présenter ce PFE sans avoir à remercier DIEU tout puissant de nous avoir prodigués l'énergie pour l'élaboration de ce modeste travail.

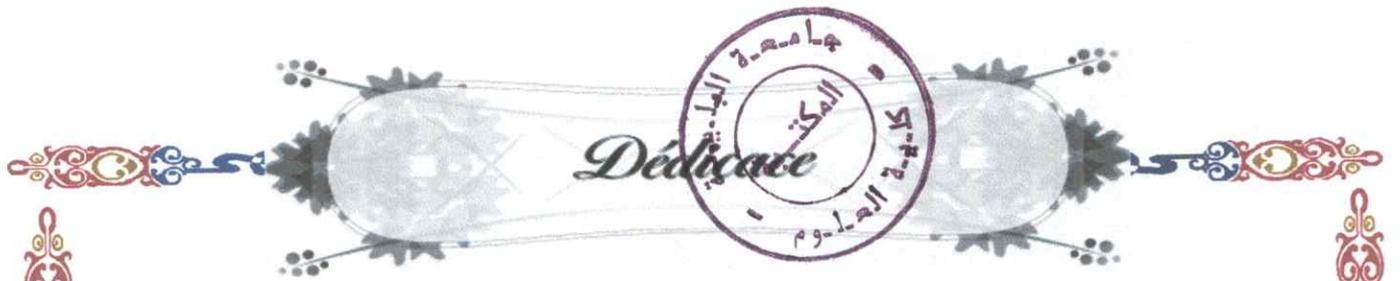
Nous remercions tout particulièrement Mme M.Guertti maître de conférences à l'ENP-Alger pour nous avoir offert l'occasion de nous former dans ce domaine et pour ses conseils précieux. Qu'elle trouve ici notre profonde gratitude.

Nous remercions aussi les membres du jury qui ont bien voulu nous faire l'honneur d'examiner et de juger ce travail.

Nous remercions aussi tous nos enseignants pour la formation qui nous a données durant toutes nos études.

Nous remercions également Mr B.Mezaourou qui nous a aidé dans la réalisation de ce travail.

Que tous ceux qui, d'une façon ou d'une autre, ont contribué de près ou de loin à la réalisation de ce travail, trouvent ici notre sincère reconnaissance.



*Je* dédie ce travail en signe de reconnaissance  
à mes très chers parents qui ont tout fait pour me  
donner une bonne éducation et me soutenir dans  
mes études.

*A* mes sœurs et mes frères.

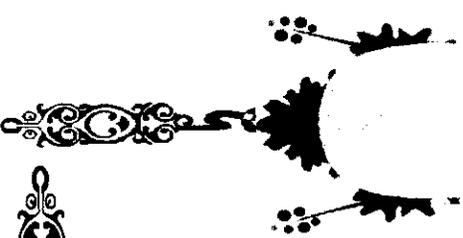
*A* M. Guerti.

*A* toute ma grande famille.

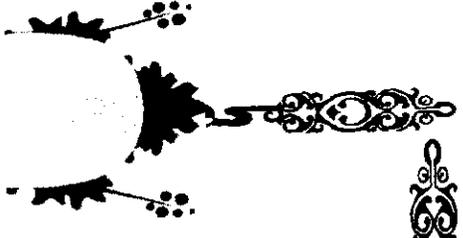
*A* mon binôme DARINE et sa famille.

*A* tous mes amis.

*ALLICHE AMEL*



*Dédicace*



*Je dédie ce travail en signe de reconnaissance  
à mes très chers parents qui ont tout fait pour me  
donner une bonne éducation et me soutenir dans  
mes études.*

*À mes sœurs et mes frères.*

*À ma grande mère.*

*À M. Guerti.*

*À toute ma grande famille.*

*À mon binôme AMEL et sa famille.*

*À tous mes amis.*

*ABOULOZ DARINE*



**Remerciements**

**Dédicaces**

**Liste des figures**

**Liste des tableaux**

**Liste des abréviations**

**Introduction générale**

## **Chapitre 1 : Généralités sur l'AS et la RAP**

1.1. Introduction .....	1
1.2. Production des sons .....	
1.3. Audition et perception .....	3
1.4. Propriétés du signal vocal.....	5
1.5. Phonétique acoustique .....	
1.5.1. La fréquence fondamentale (F0).....	
1.5.2. La durée .....	7
1.5.3. L'énergie du signal vocal .....	
1.5.4. Les formants.....	
1.6. Les spectrogrammes .....	
1.7. Description de l'Alphabet Arabe.....	8
1.7.1. Comment apprendre les voyelles ?.....	9
1.7.2. Les semi voyelles .....	
1.7.3. Comment apprendre les consonnes .....	
1.8. Reconnaissance automatique de la parole .....	13
1.8.1. Pourquoi et comment utiliser la RAP ? .....	
1.8.2. Historique .....	22
1.9. Différentes méthodes de la RAP.....	14
1.9.1. Approche globale.....	
1.9.2. Approche analytique .....	15
1.10. Analyse acoustique .....	
1.11. Décodage acoustique .....	16
1.12. Décodage acoustique .....	16
1.13. Conclusion.....	17

## Chapitre 2 : Le modèle des HMMs

2.1. Introduction.....	18
2.2. Modèle de Markov de base.....	
2.2.1. Processus stochastiques .....	19
2.2.2. Processus Markovien.....	20
2.3. Le modèle de Markov caché HMM.....	22
2.4. Les trois problèmes d'un HMM .....	23
2.4.1. Le 1 <sup>er</sup> problème (Evaluation).....	
2.4.2. Le 2 <sup>ème</sup> problème (Décodage).....	24
2.4.3. Le 3 <sup>ème</sup> problème (Apprentissage).....	25
2.5. Solution aux trois problèmes de base des HMMs .....	
2.5.1. Solution au 1 <sup>er</sup> problème .....	
2.5.1.1. L'approche directe .....	
2.5.1.2. Algorithme de Forward.....	28
2.5.1.3. Algorithme de Backward .....	30
2.5.2. Solution du 2 <sup>ème</sup> problème .....	32
2.5.3. Solution du 3 <sup>ème</sup> problème .....	
2.5.3.1. Apprentissage à partir d'une structure connue .....	34
2.5.3.2. Entraînement de Viterbi .....	35
2.5.3.3. Entraînement de Baum-Welch .....	37
2.6. Solution aux trois problèmes de base des HMMs .....	43
2.7. Conclusion.....	46

## Chapitre 3 : Application des HMMs à la RAP

3.1. Introduction.....	47
3.2. Présentation du système de RAP .....	
3.2.1. Phase d'apprentissage .....	
3.2.2. Phase de reconnaissance.....	
3.3. Acquisition des données .....	48
3.4. Elaboration du corpus .....	
3.5. Segmentation .....	49
3.6. Description du logiciel PRAAT .....	50
3.7. Extraction des formants .....	52

3.8. Phonèmes étudiés .....	52
3.9. Durées moyennes de quelques phonèmes .....	
3.10. Procédure d'apprentissage du modèle .....	53
3.10.1. Génération des observations dans HMM .....	54
3.10.2. Algorithme d'Evaluation (solution au 1 <sup>er</sup> problème) .....	56
3.10.3. Algorithme de Viterbi (solution du 2 <sup>ème</sup> problème) .....	
3.10.4. Algorithme d'apprentissage (solution du 3 <sup>ème</sup> problème) .....	59
3.11. Procédure de reconnaissance .....	62
3.12. Conclusion .....	62

## **Chapitre 4 : Expériences et Résultats de la Reconnaissance des phonèmes de l'AS**

4.1. Introduction .....	64
4.2. Procédure d'apprentissage .....	
4.2.1. Problème d'apprentissage .....	65
4.3. Trace d'exécution des algorithmes de l'apprentissage .....	
4.3.1. La trace d'exécution de l'algorithme Forward .....	
4.3.2. La trace d'exécution de l'algorithme Backward .....	
4.3.3. La trace d'exécution de l'entraînement de BaumWelch .....	66
4.4. Modèle du Phonème (MP) .....	
4.4.1. MP : ت en PI .....	
4.4.2. MP : ت en PM .....	67
4.4.3. MP : ت en PF .....	68
4.4.4. MP : د en PI .....	69
4.4.5. MP : د en PM .....	70
4.4.6. MP : د en PF .....	71
4.5. procédure de reconnaissance .....	72
4.5.1. Effet de la taille du corpus d'apprentissage .....	73
4.6. Choix du langage de programmation .....	
4.7. Description du logiciel .....	74
4.7. Conclusion .....	79

### **Conclusions générales et perspectives**

### **Références bibliographiques**

### **Annexes**

**Annexe A : Sonagrammes de tous les phonèmes de l'AS.**

# LISTE DES FIGURES

Figure 1.1 : L'appareil phonatoire humain.....	1
Figure 1.2 : L'appareil auditif humain .....	4
Figure 1.3 : Spectrogramme de la phrase : <i>الدرس الأول</i> .....	8
Figure 2.1 : Exemple d'un HMM appliqué à la météo .....	20
Figure 2.2 : Exemple d'un HMM appliqué à la vitesse d'une machine.....	21
Figure 2.3 : Représentation pratique d'un HMM.....	23
Figure 2.4 : Représentation du 1 <sup>er</sup> problème d'un HMM (Evaluation) .....	24
Figure 2.5 : Représentation du 2 <sup>ème</sup> problème d'un HMM (Décodage).....	24
Figure 2.6 : Exemple d'un HMM appliqué aux urnes et les balles.....	27
Figure 2.7 : : Illustration de l'ordre des opérations pour le calcul de la variable Forward $a_t(j)$ .....	29
Figure 2.8 : Illustration de l'ordre des opérations pour le calcul de la variable Backward $B_{t+1}$ .....	31
Figure 2.9 : Illustration de l'ordre des opérations pour le calcul de l'évènement de système .....	38
Figure 2.10 : Les trois modèles du HMM .....	43
Figure 3.1 : Système de reconnaissance automatique de la parole.....	47
Figure 3.2 : Différentes fonctionnalités du logiciel d'analyse PRAAT.....	51
Figure 3.3 : Sonagramme et formants du mot <i>زيت</i> [zayt].....	52
Figure 3.4 : Organigramme de la procédure d'apprentissage .....	55
Figure 3.5 : Organigramme de la procédure de reconnaissance phonémique.....	62

# LISTE DES TABLEAUX

<b>Tab 1.1</b> : Transcription Orthographique et Phonétique des consonnes de l'AS.....	12
<b>Tab 1.2</b> : Les grandes étapes de la RAP .....	14
<b>Tab 3.1</b> : Corpus utilisé .....	48
<b>Tab 3.2</b> : Les durées moyennes et les écart-types de quelques consonnes en AS.	53

# LISTE DES ABRÉVIATIONS

RAP	Reconnaissances Automatique de la Parole
AS	Arabe Standard
DAP	Décodage Acoustico-Phonétique
Fo	Fréquence Fondamentale
TOP	Transcription Orthographique Phonétique
HMM	Hidden Markov Models
ATR	Traduction Automatique en temps Réel
IA	Intelligence Artificielle
MP	Modèle du Phonème
PI	Position Initiale
PM	Position Médiane
PF	Position Finale

## Résumé :

Notre travail porte sur la Reconnaissance Automatique des phonèmes de l'Arabe Standard en mode monolocuteur en utilisant les Modèles de Markov Cachés ou Hidden Markov Model (HMM). Les paramètres des HMM sont obtenus après l'élaboration d'un corpus (constitué de 76 phrases et 15 mots en AS) et une étape de prétraitement basée sur une analyse sonographique afin d'extraire les formants qui constituent les résonances du conduit vocal.

Des algorithmes : Forward, Backward et Baum Welch ont été implémentés en vue de l'évaluation du taux de reconnaissance de ces phonèmes.

**Mots Clés :** Reconnaissance Automatique de la Parole, Analyse sonographique, Formants, HMM, Algorithmes : Forward, Backward, Baum-Welch.

## Abstract :

Our research is the Automatic Recognition of all the phonemes of the standard Arabic in mono speaker using hidden Markov model (HMM). The Parameters of the HMM are the results of the development of a corpus (constituted by 76 phrases and 15 words in Standard Arabic), this is a stage of post treatment based on the sonographic analysis in order to extract the formants that constitute resonances of the vocal tract.

The algorithms (Forward, Backward and Baum Welch) were implemented in order to assess the recognition rate.

**Key words:** Automatic Speech Recognition, sonographic analysis, Formants, HMM, Algorithms: Forward, Backward, Baum Welch.

## ملخص:

يدور عملنا حول التعرف الآلي على فونيمات اللغة العربية الفصحى و ذلك باستعمال نماذج ماركوف المستترة (ن.م.م) في النظام الأحادي النطق. المقاييس المتحصل عليها لـ (ن.م.م) باستعمال مدونة (متكونة من كلمات منفصلة باللغة العربية) تمت في مرحلة ما قبل المعالجة التي تعتمد على التحليل المطيافي لاستخراج الدوال الصوتية الناتجة عن الصدى في المسالك الصوتية. الخوارزميات : للأمام، للخلف، بوم والش , قد تم تجسيدها لتقدير نسبة التعرف عن الأصوات. كلمات المفتاح: نماذج ماركوف المستترة، الخوارزميات : للأمام، للخلف، فيتربي، بوم والش، دوال صوتية، التعرف الآلي على الكلام.

# Introduction générale

## Introduction générale

Parler à une machine et dialoguer avec elle est un vieux rêve de l'Homme, c'est aussi un des défis majeurs de l'Intelligence Artificielle (IA) que l'on est encore loin de savoir relever, même si des réalisations parfois spectaculaires, et d'une importance économique certaine, ont déjà vu le jour. Le problème est en fait d'une énorme difficulté et c'est seulement en introduisant des hypothèses significatives (nombre de locuteurs, niveau du bruit, taille des vocabulaires, complexité des tâches réalisées) que l'on peut espérer le résoudre.

L'accroissement des capacités et de la disponibilité des ordinateurs dans de nombreux secteurs stimulait des recherches appliquées dans les sciences de la documentation, sur la Reconnaissance Automatique de la Parole (RAP), sur les différentes formes de communication Homme-Machine etc.

Un système de reconnaissance est un système capable de traduire ou de décoder un signal acoustique de parole émis par un locuteur, et reçu à travers un capteur, en un mot ou une séquence de mots. L'importance d'un système de reconnaissance s'est montrée à travers les tâches qu'on veut réaliser sans pour cela utiliser les mains libres ou nous déplacer. On peut citer parmi ces tâches : la réservation des billets de train ou d'avion, la commande vocale pour personnes handicapées, etc.

Actuellement, des systèmes de RAP utilisés, emploient pour la phase de Décodage Acoustico-Phonétique (DAP), une méthode statistique, cette dernière est à base de Modèles de Markov Cachés ou Hidden Markov Models (HMM). Ces modèles servent à modéliser la production d'une unité (mot, syllabe, phonème, ...) par un automate probabiliste, ils ont eu beaucoup de succès grâce aux taux élevés de reconnaissance.

Notre travail s'inscrit dans le domaine de la RAP. D'une façon générale, il consiste à reconnaître les phonèmes de l'Arabe Standard (AS).

Après avoir enregistré un corpus (constitué de 76 phrases et 15 mots) en A S, comportant les consonnes et les voyelles à étudier dans les différentes positions (Initiale, Médiane et Finale). Une analyse sonographique est faite en

d'extraire les paramètres pertinents (F0, durée, formants, ...). En ce qui concerne notre travail, nous nous sommes intéressées particulièrement aux formants qui sont utilisés comme valeurs d'entrée pour les Modèles de Markov Cachés.

Notre PFE est composé de 4 chapitres :

- Le premier a pour but de présenter les notions fondamentales et les termes relatifs à la description de la parole et de son traitement automatique après des explications brèves ont été données concernant les appareils phonatoire et auditif de l'être humain.
- le deuxième expose le modèle de Markov caché (HMM), des exemples qui illustrent ce modèle ainsi que les problèmes fondamentaux des HMMs et leurs solutions par des algorithmes.
- le troisième concerne la mise en œuvre de notre travail, pour cela nous présentons en détail notre système de reconnaissance à base d'HMM.
- dans le dernier, nous présentons les différentes expériences d'apprentissage et les résultats obtenus de la reconnaissance phonémique ainsi que la description détaillée de notre logiciel d'application.

Des conclusions générales et des perspectives finissent ce travail.

# Chapitre 1 :

## Généralités sur L'AS et la RAP

## 1.1. Introduction

La parole est le principal moyen de communication dans toute société humaine. Son apparition peut être considérée comme concomitante à l'apparition des outils, l'homme ayant alors besoin de raisonner et de communiquer pour les façonner. Son abstraction par rapport à un support physique en fait un moyen de communication très simple à utiliser.

L'importance de la parole fait que toute interaction Homme-Machine devrait plus ou moins passer par elle. D'un point de vue humain, la parole permet de se dégager de toute obligation de contact physique avec la machine, libérant ainsi l'utilisateur qui peut alors effectuer d'autres tâches.

La recherche en Reconnaissance Automatique de la Parole (RAP) tente donc aujourd'hui de mieux comprendre le processus humain de génération et de compréhension de la parole, tant d'un point de vue mécanique par le biais de l'étude et de la modélisation des organes biologiques en charge de ces tâches, que d'un point de vue mathématique par le développement de méthodes de classification toujours plus fines et exactes.

## 1.2. Production des sons

Le processus de production de la parole est un mécanisme très complexe. Il repose sur une interaction entre le système neurologique et physiologique. Des organes et des muscles entrent dans la production de sons des langues naturelles (Figure 1.1).

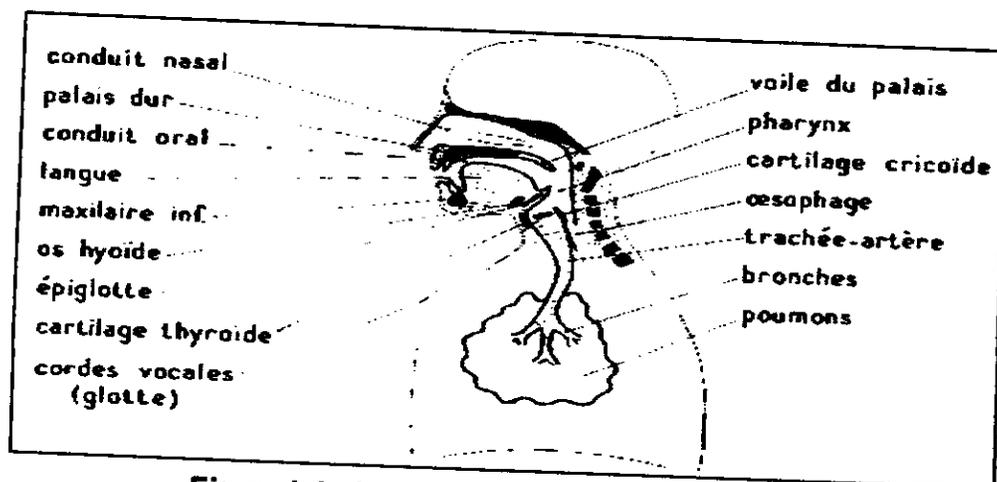


Figure 1.1 : L'appareil phonatoire humain [1].

Les différents organes de l'appareil phonatoire sont :

- les poumons : organe qui génère l'énergie (air) nécessaire à la production de la parole par un mouvement cyclique d'expiration ;
- la trachée artère : prend naissance juste après les poumons et se prolonge jusqu'au larynx ;
- le conduit vocal : partie située entre les cordes vocales et les lèvres, où circule l'air issu de la glotte ;
- les cordes vocales : ce sont deux replis de tissus fibreux situés dans le revêtement du larynx, lorsqu'on respire, elles sont très écartées formant une ouverture en V appelée glotte, par laquelle passe l'air pour former des sons, plus le son est aigu plus les cordes vocales sont tendues. Ces dernières sont soutenues par des ligaments allant de l'avant vers l'arrière du larynx ;
- la cavité buccale : elle est de forme variable selon les positions des articulateurs qui sont : la langue, les mâchoires, le palais, les dents, etc. Cette cavité se termine par les lèvres. Toute articulation peut être accompagnée d'une position des lèvres neutre, arrondie ou écartée ;
- la cavité nasale : se compose des fosses nasales qui ne sont que deux cavités de forme fixe. Le couplage entre la cavité buccale ou pharyngo-buccale et la cavité nasale s'effectue au moyen de la luette. Le phénomène résultant sera la nasalisation des sons produits [4].

La production de la parole est assurée, chez l'homme, par plusieurs organes successifs. Les poumons sont indispensables dans ce processus puisqu'ils assurent la génération d'un composant incontournable : de l'air sous pression. Cet air, expulsé, traverse alors les cordes vocales qui entrent ou non en action pour produire un voisement. Ce voisement correspond à la fréquence fondamentale qui est le timbre de la voix. Cette fréquence fondamentale étant produite, elle est propagée dans l'ensemble du conduit vocal. Ce conduit est de forme et de volume variable. Plusieurs organes concourent à ces possibles modifications qui permettent de produire des sons différents. Parmi ces organes se trouve la

langue, acteur principal des modifications qui peut agir par constriction ou occlusion du conduit vocal. Les dents et les lèvres agissent également par occlusion ou constriction, à des degrés, cependant, moindres. Le conduit vocal est, la plupart du temps, constitué du seul conduit buccal. La luette et son prolongement vers le palais, le vélum, assurent normalement la fermeture du conduit nasal pendant la production de parole. Le conduit nasal peut, dans certains cas, être connecté au conduit vocal. Cette connexion permet de générer des sons supplémentaires en modifiant le volume de la caisse de résonance normalement constituée par le seul conduit buccal.

### 1.3. Audition et perception

L'appareil phonatoire, émetteur d'informations, ne serait d'aucune utilité si l'information générée ne pouvait être captée et analysée par un récepteur. Parmi tous les récepteurs existants, l'homme a acquis la capacité de découvrir le sens caché sous les sons produits par son interlocuteur. Nous allons maintenant présenter l'anatomie de l'oreille, organe récepteur de l'information sonore, et les capacités de perception qui caractérisent cet organe lorsqu'il est en parfait état et n'a subi aucune atteinte venue amoindrir ses capacités intrinsèques.

L'appareil auditif comprend l'*oreille externe*, l'*oreille moyenne*, et l'*oreille interne*. Le conduit auditif relie le pavillon au tympan : c'est un tube acoustique de section uniforme fermé à une extrémité, son premier mode de résonance est situé vers 3000 Hz, ce qui accroît la sensibilité du système auditif dans cette gamme de fréquences. Le mécanisme de l'oreille interne (*marteau, étrier, enclume*) permet une adaptation d'impédance entre l'air et le milieu liquide de l'oreille interne. Les vibrations de l'étrier sont transmises au liquide de la *cochlée*. Celle-ci contient la *membrane basilaire* qui transforme les vibrations mécaniques en impulsions nerveuses. La membrane s'élargit et s'épaissit au fur et à mesure que l'on se rapproche de l'apex de la cochlée; elle est le support de l'*organe de Corti* qui est constitué par environ 25000 *cellules ciliées* raccordées au nerf auditif. La fréquence de résonance dépend de la position occupée par la cellule sur la

membrane; au-delà de cette fréquence, la fonction de réponse s'atténue très vite. Les fibres nerveuses aboutissent à une région de l'écorce cérébrale appelée *aire de projection auditive* et située dans le lobe temporal. En cas de lésion de cette aire, on peut observer des troubles auditifs. Les fibres nerveuses auditives afférentes (de l'oreille au cerveau) et efférentes (du cerveau vers l'oreille) sont partiellement croisées : chaque moitié du cerveau est mise en relation avec les deux oreilles internes [2].

Une description détaillée de l'oreille permettra de mieux appréhender les différents organes et de mieux visualiser leur répartition. (Figure 1.2)

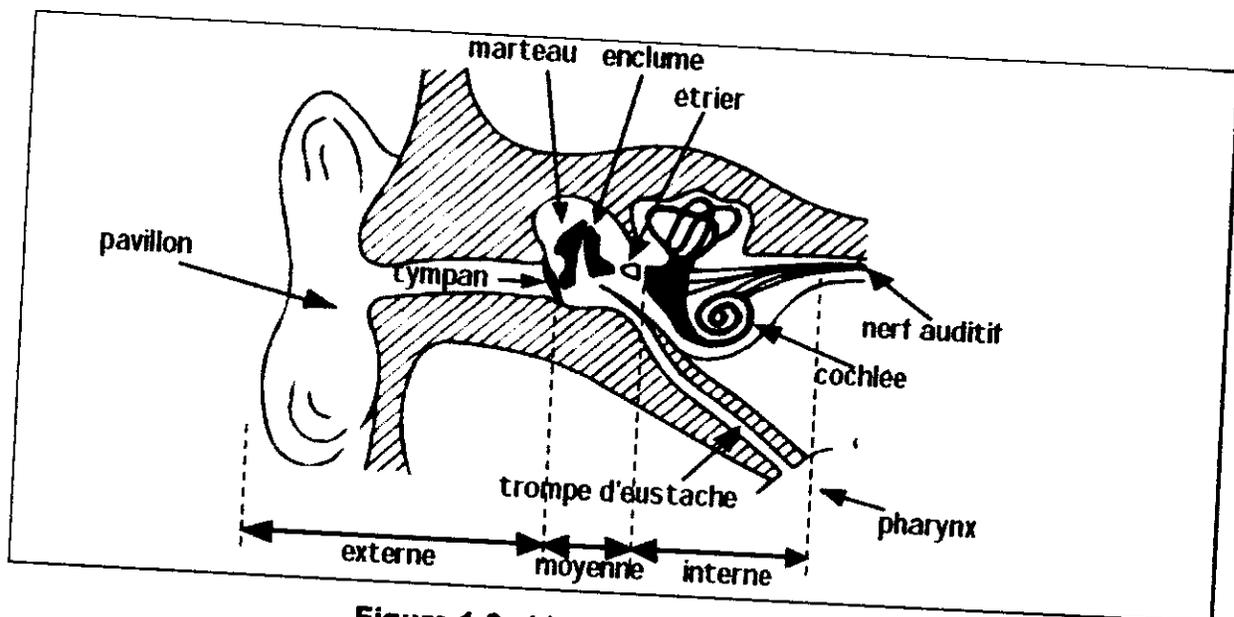


Figure 1.2 : L'appareil auditif humain [2].

Ce phénomène en apparence simple de la compréhension du langage dissimule un traitement complexe effectué par notre cerveau : après avoir séparé les sons du langage des bruits de l'environnement, le système de reconnaissance établit une première représentation qui se base sur les sons (appelés les segments) de la langue. Cette première représentation est le résultat du Décodage Acoustico\_Phonétique (DAP). Cette suite de sons perçus doit ensuite être découpée en mots : l'extraction d'un mot correspond à la sélection d'une entité mentale parmi un lexique, un peu comme une entrée de dictionnaire. Le lexique est composé de formes sonores stockées en mémoire, les entrées lexicales, qui correspondent aux mots connus, que

nous avons appris lors de l'acquisition du langage. Lorsqu'une suite de segments qui compose un mot parvient à l'entrée du système de reconnaissance, le système cherche à faire correspondre cette Introduction de suite de sons avec les formes mentales sonores dont il dispose. Cette sélection d'une forme mentale est aussi appelée l'accès au lexique. Dans une deuxième étape, la forme mentale permet de récupérer les significations reliées à ce mot, c'est-à-dire le sens de ce mot ainsi que les notions qui lui sont associées. Ce deuxième temps correspond à l'accès au sens. Enfin, les significations ou sa sémantique activées par les mots peuvent être combinées pour composer le sens entier de la phrase, lors de l'analyse syntaxique et pragmatique.

Toutes ces opérations sont effectuées dans un temps bref, et avec une grande fiabilité. Les caractéristiques acoustiques d'un même mot seront différentes s'il est prononcé par un homme, une femme ou un enfant, ou encore s'il est prononcé à voix haute, ou en chuchotant, même par la même personne. L'intonation, la qualité de la voix, le débit de parole sont autant de paramètres qui font varier le signal acoustique. S'y ajoutent d'autres facteurs, comme le fait par exemple que les propriétés articulatoires d'un son sont modifiées par ceux qui l'entourent. Ceci provoque une modification des caractéristiques acoustiques d'un son en fonction des sons qui le précèdent et le suivent (phénomène de la coarticulation : influence d'un son sur un autre son contigu).

#### **1.4. Propriétés du signal vocal**

Un signal vocal peut caractérisé par : la variabilité, la continuité.

- **Variabilité**

Une même personne ne prononce jamais un mot deux fois de façon identique. Un rhume teinte les voyelles de nasalité ; une simple fatigue et l'intensité de l'onde sonore fléchit, l'articulation perd de sa clarté. La déduction évolue dans le temps : l'enfance, l'adolescence, l'âge mûr, puis la vieillesse, autant d'âge qui marquent la voix.

Dans la variabilité inter-locuteur est encore plus flagrante. La hauteur de la voix, l'intonation l'accent diffère selon le sexe, l'origine sociale, régionale ou nationale.

- **Continuité**

Dans un signal de parole, il n'y a pas de limites précises entre les différents sons de la parole donc il n'y a pas de limite nette entre les mots.

### **1.5. Phonétique acoustique**

L'objet d'étude de la phonétique acoustique est constitué par l'onde sonore telle que produite par les organes de la phonation. Bien que cette onde soit audible,

La parole apparaît physiquement comme une variation de la pression de l'air causée et émise par le système articulatoire. La **phonétique acoustique** étudie ce signal en le transformant dans un premier temps en signal électrique. Ce dernier doit être souvent numérisé. Il est caractérisé par des paramètres pertinents, qui doivent représenter au mieux ce qu'ils sont censés modéliser et doivent extraire le maximum d'informations utiles pour la RAP.

#### **1.5.1. La fréquence fondamentale ( $F_0$ )**

Lors de la production de certains phonèmes (les sons voisés de la parole : les voyelles et certaines consonnes), la fréquence fondamentale ou  $F_0$  est la conséquence directe des variations de la pression sub-glottique (la tension des cordes vocales). Sa corrélation acoustique appelé pitch est généralement linéaire aux basses fréquences d'où la supposition d'une relation linéaire entre le pitch et la fréquence fondamentale.

#### **1.5.2. La durée**

La durée est le paramètre acoustique le plus délicat à évaluer. La difficulté de mesure réside dans sa grande variabilité qui est due au contrôle quasi impossible du système phonatoire. Chaque phonème se caractérise par

ses propres durées intrinsèques et co-intrinsèques de même que le facteur de compressibilité ou expansion.

### **1.5.3. L'énergie du signal vocal**

L'énergie correspond à la puissance du signal. Elle est souvent évaluée sur plusieurs trames successives de ce signal pour pouvoir mettre en évidence des variations.

## **1.6. Les Spectrogrammes**

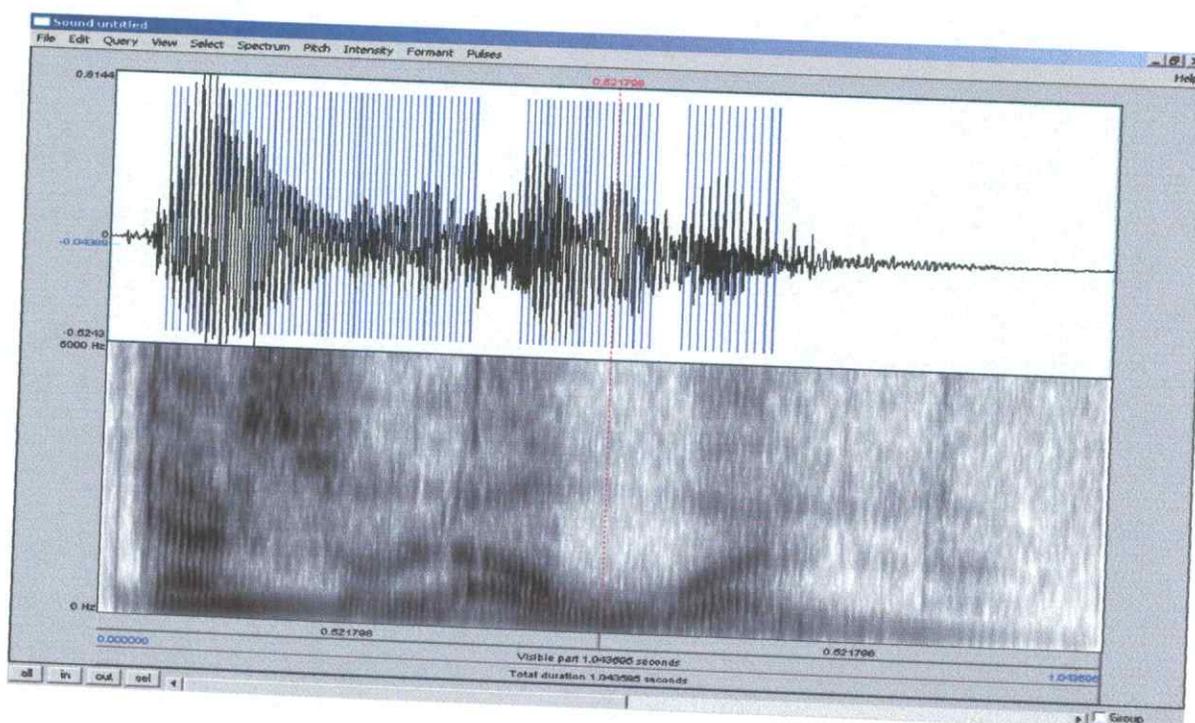
Pour illustrer notre propos, nous allons maintenant présenter une figure présentant la variabilité du signal parole (Figure 1.3). Cette figure utilise une représentation graphique largement employée en phonétique.

Le spectrogramme est un outil de visualisation utilisant la technique de la transformée de Fourier et donc du calcul de spectres. Il a commencé à être largement utilisé en 1947, à l'apparition du sonographe, et est devenu l'outil incontournable des études en phonétique pendant de nombreuses années.

Le spectrogramme permet de mettre en évidence les différentes composantes fréquentielles du signal à un instant donné.

Le spectrogramme est une représentation à 3 dimensions du signal de parole : la dimension verticale indique la composition en fréquence des sons (basses fréquences vers le bas, hautes fréquences vers le haut), la dimension horizontale donne le temps, enfin, l'intensité est représentée par le degré de noirceur.

La chaîne des caractères indiquée correspond à la suite des phonèmes, qui peuvent être identifiés sur un spectrogramme par un expert [3].



**Figure 1.3 :** spectrogramme de la phrase : « الدرس الأول ».

- La forme de l'onde acoustique correspondant au signal de parole est représentée en haut.
- Au-dessous la courbe d'énergie.

Lorsqu'un excitateur entre en vibration du spectre, il fournit un signal, dont le résonateur va amplifier certaines composantes. On obtient alors des **formants** qui sont un facteur fondamental dans la caractérisation du timbre. Ils servent, justement à former ce dernier. Le nombre des formants, selon les caractéristiques du résonateur (volume, forme et ouverture), est variable : d'un seul à une infinité (théoriquement). Néanmoins, du point de vue perceptif, seuls quelques-uns d'entre eux jouent un rôle central au niveau de la parole. Par exemple, on peut caractériser toute voyelle en ne prenant en compte que ses trois premiers formants : F1, F2 et F3.

### 1.7. Description de l'alphabet arabe

La recherche en traitement automatique de la parole et notamment en reconnaissance, dans une langue donnée doit nécessairement passer par l'étude de sa composante phonétique. Cette étude nous permet de dégager les principales caractéristiques relatives aux différentes phonèmes et ainsi de

cerner l'ensemble des paramètres acoustiques, en vue de les exploiter dans l'élaboration d'un système de reconnaissance de la parole.

Le système phonétique de l'Arabe Standard (AS) comprend six voyelles et vingt-neuf consonnes (en incluant la hamza) produits par seize lieux d'articulation.

### **1.7.1. Comment apprendre les voyelles ?**

L'articulation du son langagier est caractérisé par le libre écoulement de l'air expiré à travers le conduit vocal.

#### **1.7.1.1. Les voyelles brèves**

Il y en a trois :

[a] : fatha, elle s'écrit au dessus de la lettre.

[u] : damma, elle s'écrit au dessus.

[i] : kasra, elle s'écrit en dessous de la lettre.

Voyelle muette : elle s'écrit au dessus de la lettre.

#### **1.7.1.2. Les voyelles longues**

Il y en a trois: [aa], [uu], [ii]

#### **1.7.1.3. Les diphtongues**

Il n'existe que deux diphtongues en Arabe Standard : [ay] et [aw].

Dans la plupart des dialectes, elles sont généralement réalisées comme une voyelle longue.

### **1.7.2. Les semi voyelles**

Les semi voyelles possédant des structures formantiques similaires aux voyelles écoulement libre de l'aire. Ces phonèmes ont tout de même une obstruction qui les approche des consonnes.

### **1.7.3. Comment apprendre les consonnes?**

Son langagier impliquant une fermeture partielle ou totale du canal buccal (du passage de l'air par la bouche). Les consonnes s'opposent aux voyelles par le fait

que les voyelles sont relativement plus ouvertes que les consonnes. Il existe deux différences essentielles qui distinguent les voyelles des consonnes. D'abord, les voyelles sont toujours prononcées avec la bouche relativement plus ouverte que pour les consonnes. Ensuite, la place que chacune prend dans la syllabe est différente: les voyelles constituent le centre, le noyau de la syllabe (sans voyelle, il n'y a pas de syllabe) alors que les consonnes sont périphériques à la voyelle. Les voyelles et les consonnes sont décrites par des étiquettes différentes ou codes [4].

Traditionnellement, on dit que l'Arabe se caractérise par un consonantisme riche et un vocalisme pauvre.

Une consonne doit être accompagnée par un autre élément audible pour avoir un raisonnement donc elle ne peut pas se prononcer seule, on dit que c'est un élément non sonnant.

On peut répertorier les consonnes selon l'excitation du conduit vocale (Tab.1.1).

Les précisions ci-dessous ont pour but d'indiquer :

- les sons correspondants lorsqu'ils existent en Français ou dans une langue européenne ;
- les variantes régionales de réalisation des consonnes en AS ;
- les réalisations des mêmes consonnes dans quelques grandes familles de dialectes.

Dans la Transcription Orthographique Phonétique des consonnes de l'Arabe nous avons les différents modes : occlusifs, fricatifs, etc.

#### 1.7.2.1. Plosives ou Occlusives

Les consonnes occlusives sont obtenues momentanément, elles sont engendrées par des impulsions acoustiques provenant d'un arrêt ou déclenchement brusque de l'écoulement de l'air et elles sont caractérisées physiologiquement par 3 temps :

- la Métatase : mise en place des organes phonatoires ;
- la Tenue : qui correspond à la tension de l'occlusion plus au moins prolongé (grande énergie) ;

- la Catacase : qui coïncide avec le relâchement des organes.

#### 1.7.2.2. Fricatives

Les fricatives sont sous formes de bruit (absence de structure formatives). Elles sont produites par les sons émis lors de l'écourtement du flux d'air au niveau des points de constriction du canal buccal.

#### 1.7.2.3. Phonèmes Affriquées

Lors de sa prononciation, la langue ne s'écourte pas brusquement du palais, comme étant le cas pour les occlusives pures, mais plutôt d'une manière douce. L'air libéré sera sous forme de friction, pour cela, ils sont considérés comme des Semi Occlusives. Cette consonne particulière se comporte comme une occlusive et fricative à la fois.

Il existe en Arabe dialectal une seule affriquée; c'est [ج], comme [جلس]

#### 1.7.2.4. Nasales

Les consonnes nasales sont [م], [ن] elles sont produites lorsque il y a abstraction partielle du conduit vocal au niveau du vélum, une partie du flux d'air passe par la cavité nasale, ceci engendre une baisse d'énergie représentée par des anti-formants ou antirésonances.

#### 1.7.2.5. Liquides

La consonne liquide est engendrée par l'air traversant la cavité buccale et passe de part et d'autre de la langue.

#### 1.7.2.6. Vibrantes

La consonne vibrante est produite de la même façon que la consonne liquide mais elle est apico-alvéolaire.

**Tab 1.1:** Transcription Orthographique et Phonétique (TOP) des consonnes de l'Arabe Standard [6].

Mode	Type de phonème		Phonèmes Arabes	Transcription Arabisante	Lieux d'articulation
Occlusives	Voisées		ب د	b d	bilabiale alvéodentale
	Non-Voisées		ق ت ك ء	q t k ,	uvulaire alvéodentale postpalatale glottale
	Voisée	Emphatiques	ض	ḏ	alvéolaire
	Non-Voisée		ط	t̤	alvéodentale
Fricatives	Voisées		ز ذ س ج ح	z d g ,	sifflante dorsoalvéolaire interdentale uvulaire pharyngale
	Non-Voisées		س ش ف ح خ ح	s t̤ f š h h h	sifflante dentale interdentale labiodentale chuintante palatale vélaire glottale pharyngale
	Voisée	Emphatiques	ص	ṣ	dorsealvéodentale sifflante
	Non-Voisée		ظ	d̤	interdentale
Nasales	Voisées		م ن	m n	bilabiale alvéodentale
Liquide	Voisée		ل	l	dentale
Affriquée	Voisée		ج	ǧ	alvéopalatale
Vibrante	Voisée		ر	r	apicoalvéolaire
Semi-voyelles	Non-Voisées		و ي	w y	bilabiale palatale

## **1.8. Reconnaissance automatique de la parole RAP**

### **1.8.1. Pourquoi et comment utiliser la RAP ?**

L'utilité de la parole comme mode de communication avec une machine présente des avantages certains, notamment dans trois cas alliant souvent reconnaissance et synthèse de la parole [5] :

- Utilisateur ayant déjà les mains ou la vue occupée,
- Utilisateur occasionnel, non spécialiste d'un système,
- Accès à distance (téléphone, radiocommunication).

### **1.8.2. Historique**

La reconnaissance de la parole est une discipline récente. Vers 1950 apparut le premier système de reconnaissance de chiffres, appareil entièrement câblé et très imparfait. Vers 1960, l'introduction des méthodes numériques et l'utilisation des ordinateurs changent la dimension des recherches. Néanmoins, les résultats demeurent modestes car la difficulté du problème avait été largement sous-estimée, en particulier en ce qui concerne la parole continue. Vers 1970, la nécessité de faire appel à des contraintes linguistiques dans le décodage automatique de la parole avait été jusque-là considérée comme un problème d'ingénierie. La fin de la décennie 70 voit se terminer la première génération des systèmes commercialisés de reconnaissance de mots. Les générations suivantes, mettant à profit les possibilités sans cesse croissantes de la micro-informatique, posséderont des performances supérieures (systèmes multi locuteurs, parole continue).

## 1.9. Différentes méthodes de la RAP

Le problème de la RAP peut être résolu par deux approches différentes ; approche globale et approche analytique.

### 1.9.1. Approche globale

Dans l'approche globale, l'unité de base est considéré (le plus souvent) le mot comme une entité globale, c'est-à-dire non décomposée. L'idée de cette méthode est de donner au système une image acoustique de chacun des mots qu'il doit identifier par la suite. Elle a pour avantage d'éviter l'effet de coarticulation ou l'influence sur un son contigu à l'intérieur des mots.

Ce type de méthode est utilisé dans les systèmes suivants :

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

La reconnaissance globale comprend :

- **Reconnaissance par comparaison dynamique**

L'alignement temporel, plus connu sous l'acronyme de DTW (Dynamic Time Warping) c'est une méthode fondée sur un principe de comparaison d'un signal à analyser avec un ensemble de signaux stockés dans un dictionnaire de références.

- **Reconnaissance par les Réseaux de Neurones**

Les systèmes basés sur cette approche essaient d'imiter le comportement biologique de l'être humain capable d'apprendre à partir de son environnement. Les réseaux de Neurone ont été utilisés dans la reconnaissance de la parole pour des petits vocabulaires.

### 1.9.2. Approche analytique

Cette méthode fait intervenir un modèle phonétique du langage. Ils existent plusieurs unités minimales pour la reconnaissance qui peuvent être choisies (phonèmes, diphtonges, etc.). Le choix parmi ces unités dépend des performances des méthodes de segmentation utilisées. Cette méthode est beaucoup mieux

adaptée pour les systèmes à grand vocabulaire et pour la parole continue. Les problèmes qui peuvent apparaître dans ce type de système sont dus en particulier aux erreurs de segmentation (insertions, substitutions, recouvrements) et d'étiquetage phonétique. C'est pourquoi le DAP est fondamental dans une telle approche.

À l'heure actuelle, les modèles de Markov Cachés (HMM) pour Hidden Markov Models sont les outils de modélisation les plus employés en reconnaissance de la parole continue. Ils utilisent l'approche analytique comme méthode de reconnaissance.

Nous pouvons résumer les grandes étapes de la RAP (Tab 1.2).

**Tab1.2 : Les grandes étapes de la RAP**

<b>Année</b>	<b>Applications de la RAP</b>
1952	Reconnaissance des 10 chiffres, par un dispositif électronique câblé en mono locuteur.
1965	Reconnaissance de phonèmes en parole continue.
1968	Reconnaissance de mots isolés (500 mots) par des systèmes implantés sur PC.
1972	Premiers appareils commercialisés de reconnaissance de mots.
1978	Commercialisation d'un système de reconnaissance à microprocesseurs.
1983	Première commande vocale mondiale à bord d'un avion de chasse en France.
1985	Les premiers systèmes de reconnaissance de plusieurs milliers de mots.
1986	Lancement du projet japonais de téléphone avec traduction automatique en temps réel ATR.
1988	Apparition des premières machines à dicter par mots isolés.
1989	Recrudescence des modèles connexionnistes neurométriques.
1990	Premières véritables applications de dialogue oral Homme Machine.
1994	IBM lance son premier système de reconnaissance vocale sur PC.
1997	Lancement de la dictée vocale en continu par IBM.
2000	Développement des serveurs vocaux et l'exploitation au domaine de télécommunication.

### **1.10. Analyse acoustique**

Le signal de parole démontre une très grande variabilité. Il est en effet peu probable de mesurer deux signaux de parole totalement semblables même si des mots identiques sont prononcés par le même locuteur. Cette variabilité rend le processus de la RAP excessivement complexe. Le rôle du module d'analyse acoustique est de traiter le signal de parole de manière à réduire cette variabilité. Pour se faire, une analyse temps-fréquence est réalisée. Le signal de parole est observé au travers d'une fenêtre d'analyse d'une durée Classique de 30 millisecondes et se déplaçant par pas de 10

millisecondes. Pour chaque position de la fenêtre d'analyse, une estimation de l'enveloppe du spectre, c'est-à-dire de la répartition de l'énergie en fonction de la fréquence, du signal de parole observé est calculé. Cette estimation est exprimée de façon concise sous la forme d'un vecteur d'une dizaine de coefficients. On obtient ainsi une séquence de vecteurs acoustiques qui décrivent l'évolution temporelle de l'enveloppe spectrale du signal de parole.

### 1.11. Décodage acoustique

Pour pouvoir reconnaître un mot, le système RAP doit avoir appris à quoi ressemblent les réalisations acoustiques de ce mot en termes de vecteurs acoustiques. Durant une phase d'apprentissage, on présente au système RAP des exemples pour chacun des mots possibles. Un modèle statistique représentatif de la distribution des vecteurs acoustiques pour le mot considéré peut alors être construit.

On obtient ainsi un *modèle acoustique* de la parole, c'est-à-dire un ensemble de modèles statistiques pour chacun des mots considérés, qui est stocké dans le système RAP. Cependant, une modélisation acoustique basée sur le mot devient problématique lorsque le nombre de mots possibles devient trop important (>50 mots). En particulier, la collecte des données pour réaliser la phase d'apprentissage devient fastidieuse voire irréaliste. On préfère généralement utiliser des unités linguistiques plus petites que le mot mais en nombre plus restreint pour décrire la parole. Un choix classique est le *phonème*. La plupart des langues peuvent être définies complètement par quelques dizaines de phonèmes. Ils constituent l'ensemble des sons élémentaires dont est composé l'énoncé de tout mot, toute phrase.

### 1.12. Les problèmes de la RAP

La grande difficulté de la RAP provient du caractère du processus de la communication parlée et des caractéristiques du signal vocal.

### 1.12.1. Variabilité

Une même personne ne prononce jamais un mot deux fois de façon identique.

### 1.12.2. Continuité

Il n'y a pas de limites précises entre les différents sons de la parole donc il n'y a pas de limite nette entre les mots.

### 1.12.3. Coarticulation

Les sons émis dans une même chaîne acoustique subissent l'influence de ceux qui les suivent ou les précèdent.

*Exemple* : [apsã] elle s'écrit : absent ( [s] est un phonème sourd alors il influe sur le [b] ).

### 1.12.4. Homophonie

Il y a homophonie lorsqu'un même son correspond à plusieurs sens. Si ces différents sens s'écrivent pareil, ils sont *homographes*

*Exemple*: J'ai mangé un avocat .

L'avocat de la défense vient de plaider.

## 1.13. Conclusion

Ce chapitre nous a permis de présenter des notions fondamentales sur la parole, la reconnaissance automatique de la parole ainsi que la description des phonèmes de l'arabe standard.

Dans le chapitre suivant, nous allons décrire une des méthodes utilisées dans la reconnaissance automatique de la parole qui est le Modèle de Markov Caché ou Hidden Markov Model (HMM).

# Chapitre 2 :

## Les modèles de HMM

## 2.1. Introduction

Notre objectif dans ce chapitre est de présenter en détail le modèle de Markov de base et une vue d'ensemble sur la théorie des HMMs. Nous abordons les algorithmes : Forward, Backward, Viterbi et Baum-Welch qui nous permettront de résoudre les trois problèmes fondamentaux liés aux HMM, correspondant aux problèmes d'évaluation, de décodage et d'apprentissage. Ces derniers seront exposés et illustrés autant que possible par des définitions et des exemples, de plus la présentation détaillée des solutions des éventuels problèmes, enrichies par la description des algorithmes, et par la suite nous présentons les différentes topologies des HMM.

Markov est un chercheur Mathématicien qui a développé cette méthode qui porte son nom.

Pour aborder un phénomène difficilement explicable :

- soit on l'analyse pour en comprendre le fonctionnement par ensemble de causes et d'effets ;
- soit on refuse de comprendre et on fait des statistiques sur le nombre d'occurrences des différentes manifestations du phénomène à étudier, Ce qui signifie que le phénomène nous reste cachée et que notre compréhension n'ira au-delà des observations faites sur lui.

Les Modèles de Markov cachés reposant sur ce deuxième principe qui vient d'être cité.

## 2.2. Modèle de Markov de base

Les modèles de Markov sont des automates probabilistes à états finis. Dans le cadre de notre travail, nous allons nous intéresser plus spécifiquement aux modèles discrets. Tous ces modèles se basent sur l'hypothèse de Markov, qui

peut se résumer à : « *le futur ne dépend que de l'état présent* ». Cette hypothèse implique que l'état du modèle doit contenir suffisamment d'informations pour permettre une prédiction parfaite du comportement du système. Dans la majorité des situations réelles, cette hypothèse n'est pas vérifiée, car il faudrait pour cela, prendre en compte un trop grand nombre de paramètres. Cependant, si l'on se restreint à un sous-ensemble de propriétés, on peut souvent construire un modèle pour lequel la propriété Markovienne est presque respectée.

Le modèle Markovien de base correspond à un simple graphe d'états, doté d'une fonction de transition probabiliste. A chaque pas de temps, le modèle subit une transition qui va potentiellement modifier son état. Cette transition permet donc au système modélisé d'évoluer, selon une loi connue par avance. Néanmoins, cette loi de transition est probabiliste. En effet, l'évolution du système peut être incertaine, ou simplement mal connue. Cette fonction probabiliste permet donc d'exprimer simplement la loi d'évolution du modèle, sous la forme d'une matrice de probabilités. Cela ouvre la porte à un très grand nombre d'utilisations où l'évolution d'un système n'est connue qu'à travers des statistiques.

#### Notations

- $S$  : Ensemble fini d'états.
- $A$  : Fonction de transition :  $S \times S \rightarrow [0; 1]$

### 2.2.1. Processus stochastique

Un modèle stochastique est un processus aléatoire qui peut changer d'état si,  $i = 1, \dots, n$  au hasard aux instants  $t = 1, 2, \dots, T$ . A chaque état est associée une variable aléatoire notée simplement  $q(t) = s_i$ , qui prend ses valeurs dans un ensemble d'observations.

Un processus stochastique peut être noté comme suit :

$$\{q(t) / t \in T, q(t) \in E\} \quad (2.1)$$

### 2.2.2. Processus Markovien

Soit un ensemble d'instants  $T = \{t_1, t_2, \dots, t_n\}$ , tel que  $t_1 < t_2 < \dots < t_n$  et un ensemble d'états  $S = \{S_1, S_2, \dots, S_n\}$ , avec  $S \subseteq E$ . Un processus markovien est un processus qui satisfait la propriété suivante : 0

$$P\{q(t_n) \in S / q(t_{n-1}) = S_{n-1}, q(t_{n-2}) = S_{n-2}, \dots, q(t_1) = S_1\} = P\{q(t_n) \in S / q(t_{n-1}) = S_{n-1}\} \quad (2.2)$$

Cette propriété signifie que la variable  $q(t_{n-1})$  résume le passé de la séquence. Si l'espace des temps  $T$  est discret cette propriété peut aussi s'écrire :

$$P\{q_n \in S / q_{n-1} = S_{n-1}, q_{n-2} = S_{n-2}, \dots, q_1 = S_1\} = P\{q_n \in S / q_{n-1} = S_{n-1}\} \quad (2.3)$$

#### Exemple d'un Modèle météo hyper-simplifié :

- N=3 états :  $S_1$ =pluvieux,  $S_2$ =nuageux,  $S_3$ =ensoleillé
- t = une journée
- Grâce à des mesures passées de Météo France, on pourrait calculer la matrice de transition  $A = [a_{ij}]$

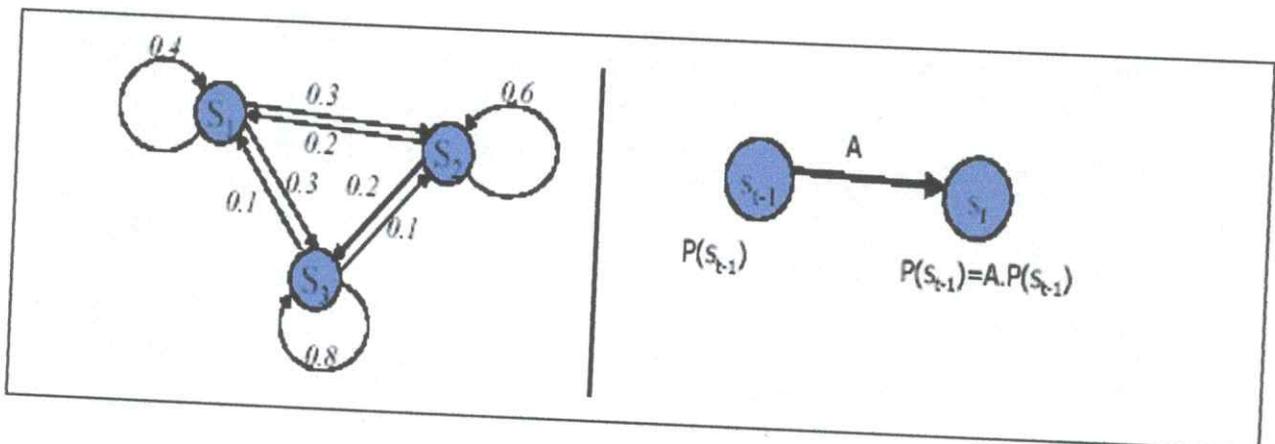


Figure 2.1 : Exemple d'un HMM appliqué à la météo

$$A = \{ a_{ij} \} = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix}$$

**Exemple d'utilisation :**

Aujourd'hui ( $t=1$ ), il fait beau  $P(s_1 = S_3)=1$ . Quelle est la probabilité que le temps des 7 prochains jours soit : soleil - soleil - pluie - pluie - soleil - nuage - soleil ?

**Résolution :**

Soit  $O$ , la séquence des observations de  $t=1$  à 8

$$O = \{S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3\}.$$

On peut utiliser le modèle pour calculer  $P(O)$  :

$$P(O/\text{modèle}) = P(S_3) \cdot P(S_3/S_3) \cdot P(S_3/S_3) \cdot P(S_1/S_3) \cdot P(S_1/S_1) \cdot P(S_3/S_1) \cdot P(S_2/S_3) \cdot P(S_3/S_2)$$

$$= \pi_3 \cdot a_{33} \cdot a_{33} \cdot a_{31} \cdot a_{131} \cdot a_{13} \cdot a_{32} \cdot a_{23}$$

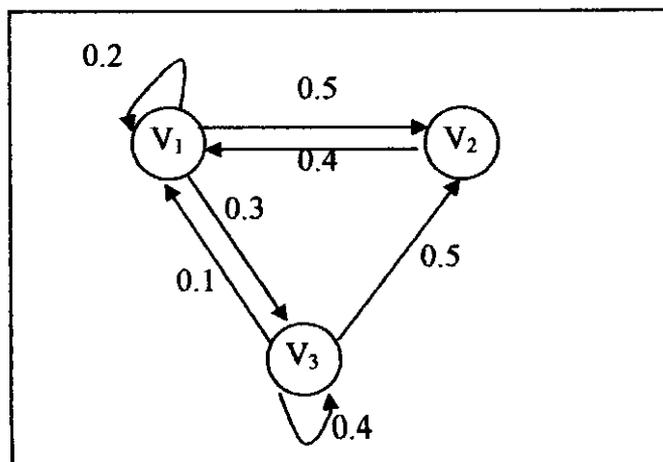
[En notant  $\pi_j = P(s_1 = S_j)$ ].

**Exemple 2 :**

Une machine possède 3 vitesses  $V_1, V_2, V_3$ . Initialement la machine prend la vitesse  $V_1$  avec une probabilité 0.7 et la vitesse  $V_2$  avec 0.2 et la vitesse  $V_3$  avec 0.1.

La probabilité de passer de :

- $V_1$  à  $V_2$  est de 0.5 et de  $V_1$  à  $V_3$  est de 0.3;
- $V_2$  à  $V_1$  " 0.5 et de  $V_2$  à  $V_3$  " 0.3;
- $V_3$  à  $V_2$  " 0.5 et de  $V_3$  à  $V_1$  " 0.3.



$$A = \{a_{ij}\} = \begin{pmatrix} 0.2 & 0.5 & 0.3 \\ 0.4 & 0 & 0.6 \\ 0.1 & 0.5 & 0.4 \end{pmatrix}$$

Figure 2.2 : Exemple d'un HMM appliqué à la vitesse d'une machine.

### 2.3. Modèle de Markov Caché (HMM)

Le modèle de Markov Caché est la technique stochastique la plus fréquemment utilisée, elle convient bien pour modéliser des données de type "signal" séquencées par le temps propres à la parole, à l'écriture, ...

Un HMM est un processus doublement stochastique au sens où il est constitué d'un processus stochastique sous-jacent qui n'est pas observable directement (il est caché), il ne peut être observé qu'au travers d'un autre ensemble de processus stochastiques, qui produisent la séquence de symboles observés. C'est-à-dire qu'il y a donc deux types de probabilités :

- une de changement d'état ;
- une autre d'émission de symboles.

Dans le HMM de base, tout état est relié à tout autre et à lui-même (modèles ergodiques). Mais, pour certaines applications, ce type de HMM n'est pas très indiqué, on préférera des modèles non-ergodiques, afin de pouvoir modéliser la séquentialité des phénomènes.

#### Notations HMM :

- $s$  : état (non observable) à valeurs dans  $\{S_i, i=1:N\}$
- $A = [a_{ij}] = [P(s_t = S_j | s_{t-1} = S_i)]$
- $S$  = séquence de  $s = \{s_1 s_2 \dots s_T\}$
- $o$  : observable à valeurs dans  $\{O_i, i=1:M\}$
- $B = [b_j(O_k)] = [P(o_t = O_k | s_t = S_j)]$
- $O$  = séquence d'observables =  $\{o_1, o_2, \dots, o_T\}$
- $\Pi$ : état initial du système

<b>HMM définit par <math>\lambda = (A, B, \Pi)</math></b>
---

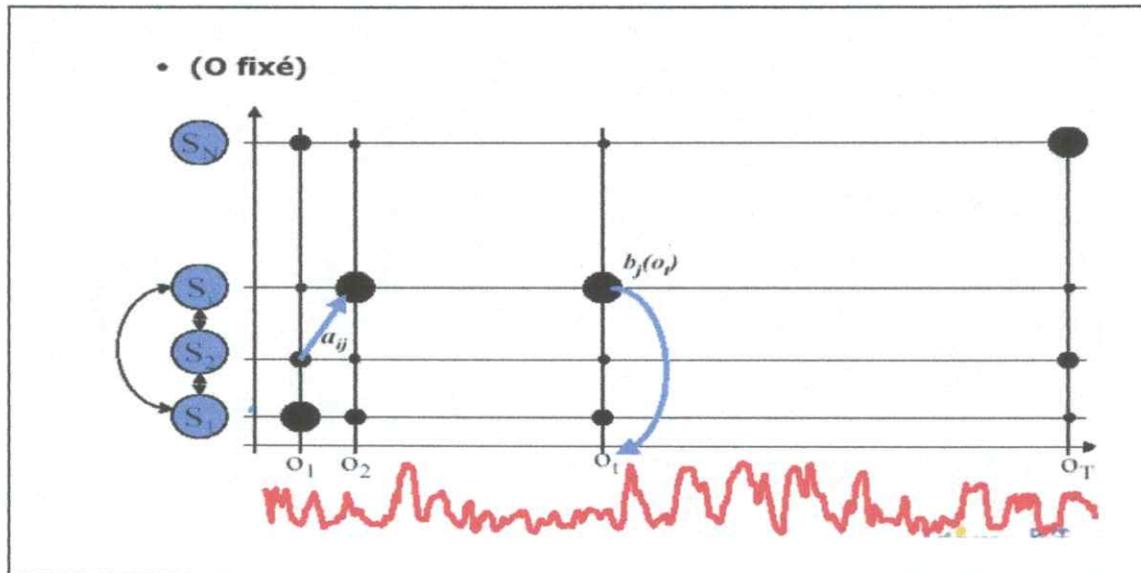


Figure 2.3. Représentation pratique d'un HMM.

## 2.4. Les trois problèmes d'un HMM

Il existe trois problèmes incontournables pour un Modèle de Markov Caché qu'on doit résoudre : l'évaluation, le décodage et l'apprentissage d'un HMM [7].

### 2.4.1. Le 1<sup>er</sup> problème (Evaluation)

Etant donné une séquence d'observations  $O = o_1, o_2, \dots, o_T$ , et un modèle  $\lambda = (\pi, A, B)$ , le 1<sup>er</sup> problème consiste à calculer (la probabilité de générer la séquence d'observations  $O$  suivant le modèle  $\lambda$ ). Ce point de vue est extrêmement utile, par exemple, si nous considérons le cas dans lequel nous essayons de choisir parmi plusieurs modèles concurrents, la solution au 1<sup>er</sup> problème nous permettant de choisir le modèle qui génère le mieux les observations.

**Illustration** : on parcourt toutes les séquences d'états possibles comme la figure 2.4 indique :

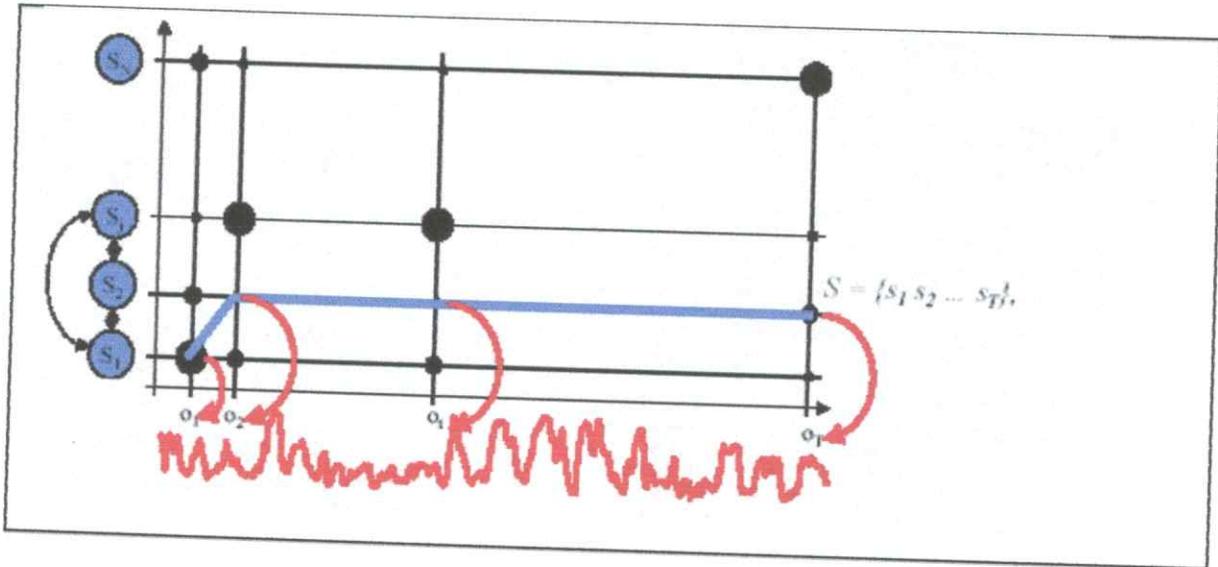


Figure 2.4 . Représentation du 1<sup>er</sup> problème d'un HMM (Evaluation).

### 2.4.2. Le 2<sup>ème</sup> Problème (Décodage)

Etant donné une séquence d'observations  $O = o_1, o_2, \dots, o_T$  et un modèle  $\lambda = (A, B, \pi)$ , Qu'elle est la séquence d'états  $Q = q_1, q_2, \dots, q_T$  suivant le modèle  $\lambda$  qui a la probabilité maximale de générer  $O$  ? (le chemin le plus probable), c'est-à-dire que nous essayons de découvrir la partie cachée du modèle.

Quelle est la séquence cachée  $S$  qui a pu produire  $O$  ?

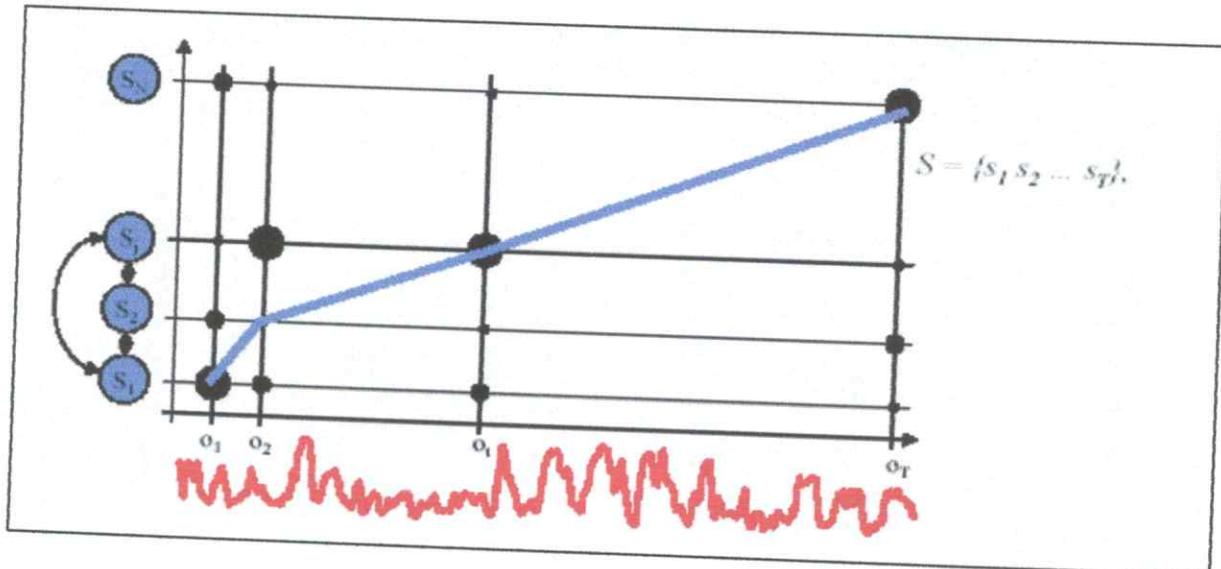


Figure 2.5. Représentation du 2<sup>ème</sup> problème du HMM (Décodage).

### 2.4.3. Le 3<sup>ème</sup> Problème (Apprentissage)

Etant donné une séquence d'observations  $O$ , Comment peut-on reestimer les paramètres  $\lambda = (\pi, A, B)$  afin de maximiser  $P(O/\lambda)$ ? La séquence d'observation employée pour ajuster les paramètres du modèle s'appelle une séquence d'apprentissage (ou séquence de formation puisqu'elle est employée pour former le HMM). Le problème d'apprentissage est le problème crucial pour la plupart des applications des HMMs, puisqu'il nous permet d'adapter de façon optimale les paramètres modèles aux données observées de formation, c'est-à-dire, pour créer les meilleurs modèles pour de vrais phénomènes.

## 2.5. Solution aux trois problèmes de base des HMMs

### 2.5.1. Solution au 1<sup>er</sup> problème (Evaluation)

Pour ce problème il y a trois solutions

#### 2.5.1.1. L'approche directe

Etant donné un HMM  $\lambda = (\pi, A, B)$ , Nous souhaitons calculer la probabilité de génération de la séquence d'observations  $O = o_1, o_2, \dots, o_T$ , (c'est-à-dire  $p(O/\lambda)$ ). La manière la plus franche de faire ceci est en énumérant chaque ordre possible d'états de longueur  $T$  (le nombre de symboles de la séquence d'observations, c'est-à-dire  $T=O$ ). Considérons un tel ordre fixe d'états :  $Q = q_1, q_2, \dots, q_T$ , où  $q_1$  est l'état initial, la probabilité de générer la séquence d'observations  $O$  suivant l'ordre d'état  $Q$  est :

$$\begin{aligned} P(O/Q, \lambda) &= \prod_{t=1}^T P(o_t / q_t, \lambda) \\ &= b_{q_1}(o_1) \cdot b_{q_2}(o_2) \dots b_{q_T}(o_T) \end{aligned} \quad (2.4)$$

La probabilité d'un tel ordre  $Q$  d'états peut être écrite comme suit :

$$P(Q, \lambda) = \pi_{q_1} \cdot a_{q_1, q_2} \cdot a_{q_2, q_3} \dots a_{q_{T-1}, q_T} \quad (2.5)$$

La probabilité commune de O et de Q est la probabilité que O et Q se produisent simultanément, c'est simplement le produit des probabilités des deux termes précités :

$$P(O, Q | \lambda) = P(O | Q, \lambda).P(Q, \lambda) \quad (2.6)$$

La probabilité de générer O (selon le modèle) est obtenue en additionnant cette probabilité commune, pour tous les ordres possibles d'états de longueur T :

$$\begin{aligned} P(O | \lambda) &= \sum_{q \in Q} P(O | Q, \lambda).P(Q, \lambda) \\ &= \sum_{q \in Q} \pi_{q_1} . b_{q_1}(o_1) . a_{q_1, q_2} . b_{q_2}(o_2) . a_{q_2, q_3} \dots a_{q_{T-1}, q_T} . b_{q_T}(o_T) \end{aligned} \quad (2.7)$$

Initialement (à t=1) nous sommes dans l'état  $q_1$  avec une probabilité  $\pi_{q_1}$  en produisant le symbole  $o_1$  (dans cet état) avec une probabilité  $b_{q_1}(o_1)$ . On passe du "t" à "t+1" (donc t=2) et nous transitons de l'état  $q_1$  vers l'état  $q_2$  avec une probabilité  $a_{q_1, q_2}$  en produisant le symbole  $o_2$  avec la probabilité  $b_{q_2}(o_2)$ . Le processus continue de cette manière jusqu'à ce que nous arrivions à t=T-1, nous faisons la dernière transition de l'état  $q_{T-1}$  vers l'état  $q_T$  avec une probabilité  $a_{q_{T-1}, q_T}$  en produisant le symbole  $o_T$  avec la probabilité  $b_{q_T}(o_T)$ .

**Exemple :** « Les urnes et les balles » [4].

- N : urnes qui contiennent chacune des balles de couleur. Au total, il y a M couleurs différentes de balles;
- protocole : un génie tire une balle d'une urne initiale. Il annonce la couleur de la balle et repose la balle dans son urne. Selon un processus aléatoire dépendant de la dernière urne concernée, le génie choisit une nouvelle urne (qui peut être la même que la précédente) et effectue le tirage d'une balle dans cette urne, etc ;

- alors le processus sous-jacent peut être modélisé par un HMM à N états, où la couleur de la balle sélectionnée est modélisée par une distribution probabiliste propre à chaque état (chaque urne) ;
- le choix d'une urne par le génie est modélisé par les probabilités de transition d'un état (urne) vers un autre.

Prenons par exemple, 3 couleurs : V J R respectivement Verte, Jaune et Rouge. 3 urnes (qui représentent les états  $S_1$ ,  $S_2$  et  $S_3$ ). On peut modéliser le processus par le HMM suivant (figure 2.5):

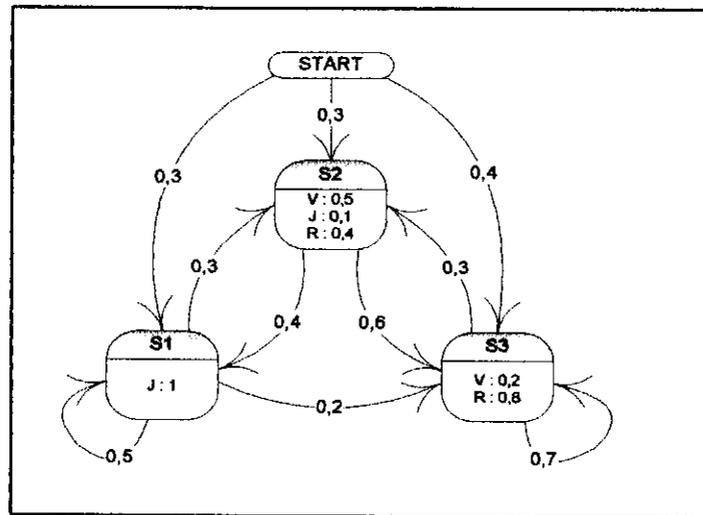


Figure 2.5 : Exemple d'un HMM appliqué aux urnes et les balles

Le modèle ci-dessus est spécifié par  $\lambda = (\pi, A, B)$  où :

$$\pi = [0.3 \quad 0.3 \quad 0.4]$$

$$A = \begin{bmatrix} & S_1 & S_2 & S_3 \\ S_1 & 0.5 & 0.3 & 0.2 \\ S_2 & 0.4 & 0 & 0.6 \\ S_3 & 0 & 0.3 & 0.7 \end{bmatrix}$$

$$B = \begin{bmatrix} & V & J & R \\ S_1 & 0 & 1 & 0 \\ S_2 & 0.5 & 0.1 & 0.4 \\ S_3 & 0.2 & 0 & 0.8 \end{bmatrix}$$

A, B sont respectivement les matrices de transition et d'émission, Soit l'observation O :

O = {VJR} – Question P (O/λ) ?

Il existe 5 chemins qui génèrent O suivant  $\lambda$  :

$$c_1 = \{S_2 S_1 S_3 S_3\} \quad c_2 = \{S_2 S_1 S_3 S_2\} \quad c_3 = \{S_2 S_1 S_2 S_3\} \quad c_4 = \{S_3 S_2 S_3 S_2\} \quad c_5 = \{S_3 S_2 S_3 S_3\}$$

$$p(O / c_1) = \pi_2 \cdot b_2(V) \cdot a_{21} \cdot b_1(J) \cdot a_{13} \cdot b_3(R) \cdot a_{33} \cdot b_3(V) = 0.00134000$$

$$p(O / c_2) = \pi_2 \cdot b_2(V) \cdot a_{21} \cdot b_1(J) \cdot a_{13} \cdot b_3(R) \cdot a_{32} \cdot b_2(V) = 0.00144000$$

$$p(O / c_3) = \pi_2 \cdot b_2(V) \cdot a_{21} \cdot b_1(J) \cdot a_{12} \cdot b_2(R) \cdot a_{23} \cdot b_3(V) = 0.00066400$$

$$p(O / c_4) = \pi_3 \cdot b_3(V) \cdot a_{32} \cdot b_2(J) \cdot a_{23} \cdot b_3(R) \cdot a_{32} \cdot b_2(V) = 0.00017280$$

$$p(O / c_5) = \pi_3 \cdot b_3(V) \cdot a_{32} \cdot b_2(J) \cdot a_{23} \cdot b_3(R) \cdot a_{33} \cdot b_3(V) = 0.00016128$$

$$p(O / \lambda) = \sum_{i=1}^5 p(O / c_i, \lambda) = 0.00377808$$

On peut facilement voir que le calcul de  $P(O/\lambda)$  selon l'approche directe implique un ordre de calcul de  $2T$ .

$N^T$ , puisque pour chaque  $t = 1, 2, \dots, T$ , il y a  $N$  états possibles qui peuvent être atteints (les ordres possibles d'états est  $N^T$ ), et pour chaque ordre d'états  $2T$  opérations sont exigés pour calculer la somme. (Pour être précis, nous avons besoin de  $(2T-1)N^T$  multiplications, et  $N^T-1$  d'additions). Pratiquement un tel calcul est indésirable, remarquons même pour de petites valeurs de  $N$  et de  $T$ , par exemple, pour  $N=5$  états,  $T=100$  observations, l'ordre de calcul est de  $2 \cdot 100 \cdot 5^{100} = 10^{72}$  opérations, Heureusement une procédure efficace pour résoudre le 1<sup>er</sup> problème existe, il s'agit de l'algorithme *Forward-Backward*.

### 2.5.1.2. Algorithme de Forward

Considérons la variable Forward définie par :

$$\alpha_t(i) = p(o_1 \dots o_t, q_t = S_i / \lambda) \quad (2.8)$$

Elle exprime la probabilité d'avoir émis la séquence partielle  $o_1, o_2, \dots, o_t$  tout en se trouvant dans l'état  $S_i$  [8]. Cette variable peut être calculé de manière inductive :

**Algorithme d'évaluation Forward**

**Etape 1 : Initialisation :** pour tout  $i \in [1, N]$

$$\alpha_1(i) = \pi_i b_i(o_1)$$

**Etape 2 : Induction :** pour tout  $t \in [1, T-1]$  et pour tout  $j \in [1, N]$

$$\alpha_{t+1}(j) = \left( \sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right) b_j(o_{t+1})$$

**Etape 3 : Terminaison :**

$$p(O | \lambda) = \sum_{i=1}^N \alpha_T(i)$$

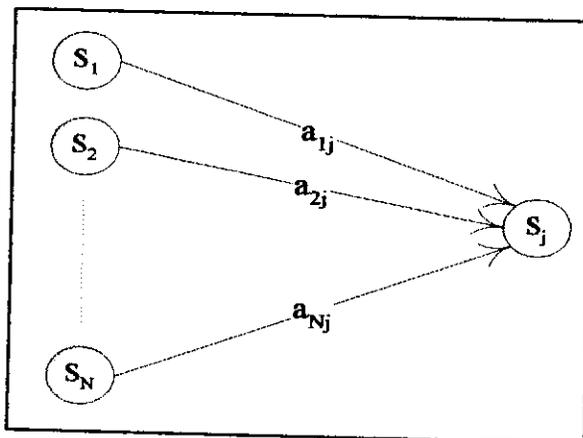
Nous expliquons ces différents étapes comme suit :

**Etape 1 : « étape d'initialisation »**

L'étape 1 Initialiser les probabilités Forward par la probabilité de passage de l'état initial (start) vers  $S_i$  en observant le premier symbole  $o_1$  de la séquence  $O$ .

**Etape 2 : « étape d'induction »**

Cette étape Qui est le coeur du calcul Forward (figure 2.4) :



**Figure 2.4 :** Illustration de l'ordre des opérations requises pour le calcul de la variable Forward  $\alpha_t(j)$  [7].

Cette figure montre comment l'état  $S_j$  peut être atteint à  $t+1$  à partir de  $S_i$ ,  $1 \leq i \leq N$  au temps  $t$ . Donc on peut démontrer l'induction du calcul Forward [8] :

$$\begin{aligned}
\alpha_{t+1}(j) &= p(o_1 \dots o_t o_{t+1}, q_{t+1} = S_j) \\
&= \left[ \sum_{i=1}^N p(o_1, \dots, o_t, q_t = S_i) \cdot p(q_t = S_i, q_{t+1} = S_j) \right] b_j(o_{t+1}) \\
&= \left[ \sum_{i=1}^N p(o_1, \dots, o_t, q_t = S_i) \cdot a_{ij} \right] b_j(o_{t+1}) \\
&= \left[ \sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right] b_j(o_{t+1})
\end{aligned} \tag{2.9}$$

Le calcul de l'induction est exécuté pour tous les états  $j$  tel que  $1 \leq j \leq N$ , pour une  $t$  donnée, le calcul est alors réitéré pour  $1 \leq t < T$ .

### Etape 3 : « étape de terminaison »

En conclusion, après avoir calculé les  $\alpha_T(i)$   $1 \leq i \leq N$ , il faut trouver la probabilité  $P(O/\lambda)$  de génération de la séquence  $O$  suivant le modèle  $\lambda$ . On a par définition:

$$\alpha_T(i) = p(o_1 o_2 \dots o_T, q_T = S_i) \tag{2.10}$$

On peut donc conclure  $P(O/\lambda)$  par qui suit :

$$\begin{aligned}
P(O/\lambda) &= \sum_{i=1}^N p(o_1, \dots, o_T / q_T = S_i) \cdot p(q_T = S_i) \\
&= \sum_{i=1}^N p(o_1, \dots, o_T, q_T = S_i) \\
&= \sum_{i=1}^N \alpha_T(i)
\end{aligned} \tag{2.11}$$

Si nous examinons le calcul impliqué dans cet algorithme, nous voyons qu'il exige un ordre de  $TN^2$  plutôt que  $2TN^T$  selon les exigences du calcul direct (Encore, pour être précis, nous avons besoin de  $N(N+1)(T-1) + N$  multiplications et  $N(N-1)(T-1)$  additions). Reprenons l'exemple précédent  $N=5$  états,  $T=100$  observations, nous avons besoin d'environ 3000 opérations pour la méthode Forward, contre  $10^{72}$  opérations pour l'approche directe, l'épargne est d'environ 69 ordres de grandeur.

Cet algorithme est appelé Forward car l'induction est réalisée en avant, on calcule tout d'abord la probabilité de générer le premier symbole de la séquence, puis à chaque étape de l'induction on rajoute un symbole et on réitère la procédure

jusqu'à ce qu'on ait calculé la probabilité de génération de la séquence entière. Bien que moins naturel, un algorithme similaire, l'algorithme Backward, peut être utilisé pour réaliser le calcul en arrière [9].

### 2.4.2.3. Algorithme de Backward

Il est aussi possible de réaliser le calcul en arrière, ce qui donne l'algorithme Backward. Pour cela, on définit la variable Backward :

$$\beta_t(i) = p(o_{t+1}, \dots, o_T / q_t = S_i, \lambda) \quad (2.12)$$

Cette équation signifie la probabilité de générer la séquence d'observations  $O_{t+1}, \dots, O_T$  sachant qu'on se trouvait dans l'état  $S_i$  à l'instant  $t$  [4]. Cette variable se calcule de manière inductive, comme suit :

#### Algorithme d'évaluation Backward

**Etape 1 :** Initialisation : pour tout  $i \in [1, N]$

$$\beta_T(i) = 1$$

**Etape 2 :** Induction : pour tout  $t \in [1, T-1]$  et pour tout  $j \in [1, N]$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j)$$

**Etape 3 :** Terminaison :

$$p(O / \lambda) = \sum_{i=1}^N \pi_i \cdot \beta_1(i) \cdot b_i(o_1)$$

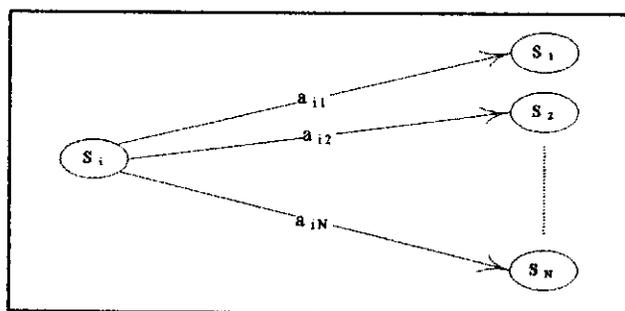
Nous expliquons les différents étapes de cet algorithme comme suit :

**Étape 1 :** « étape d'initialisation »

L'étape 1 définit arbitrairement les valeurs de  $\beta_T(i)$  par 1 pour toute  $i \in [1, N]$ .

**Etape 2 :** « étape d'induction »

L'induction du calcul Backward est illustrée dans la figure 2.5 :



**Figure 2.5 :** Illustration de l'ordre des opérations requises pour le calcul de la variable Backward  $\beta_{t+1}(i)$  [7].

### Etape 3 : « étape de terminaison »

En conclusion, après avoir calculé les  $\beta_1(i)$   $1 \leq i \leq N$ , c'est-à-dire la probabilité de générer la séquence  $O$  en partant de l'état  $S_i$  et d'arriver sur le dernier état qui génère le dernier symbole de la séquence, le calcul de  $P(O/\lambda)$  est simplement la somme du produit de la probabilité de passage de l'état initial vers  $S_i$  par  $\beta_1(i)$  par la probabilité de générer le 1<sup>er</sup> symbole de la séquence  $b_i(o_1)$ ,  $1 \leq i \leq N$ .

Encore, le calcul du  $\beta_t(i)$   $1 \leq i \leq N$ ,  $1 \leq t \leq T$ , exige un ordre de  $TN^2$  d'opérations. Nous verrons plus tard comment les variables Forward-Backward sont servies intensivement pour aider à résoudre le 3<sup>ème</sup> problème des HMMs.

### 2.5.2. Solution du 2<sup>ème</sup> problème (Décodage)

Le problème que l'on se pose ici est de trouver, la séquence d'états du HMM qui a la probabilité maximale de générer  $O$ , étant donnée une séquence de symboles  $O = o_1, o_2, \dots, o_T$  et un HMM  $\lambda = (\pi, A, B)$ . Ce qui nous préoccupe n'est pas la valeur de la probabilité maximale mais le chemin appelé chemin de Viterbi - qui permet de générer la séquence  $O$  avec cette probabilité. De manière similaire à l'approche utilisée pour le calcul de  $P(O/\lambda)$ , l'approche directe pour résoudre ce problème consiste à calculer la probabilité de génération suivant tous les chemins possibles et de choisir parmi ces chemins celui qui a la probabilité la plus élevée. Cette approche a, comme pour le calcul de  $P(O/\lambda)$ , une complexité en  $TN^T$  et est donc également indésirable en pratique. L'algorithme de Viterbi est un algorithme de programmation dynamique très similaire à l'algorithme Forward et qui permet de résoudre efficacement ce problème [4].

Une technique formelle pour trouver le meilleur ordre d'état  $Q = q_1, q_2, \dots, q_T$  qui expliquent le mieux la séquence donnée  $O = o_1, o_2, \dots, o_T$ , basé sur l'algorithme de Viterbi, on cherche donc à maximiser  $P(Q/O, \lambda)$  ce qui revient au même que de maximiser  $P(O, Q/\lambda)$ , car on a :

$$P(O, Q/\lambda) = P(Q/O, \lambda).P(O/\lambda) \quad (2.13)$$

Pour cela, définissons la probabilité maximale d'une séquence au temps  $t$  qui se termine dans l'état  $S_i$ ,

$$\delta_t(i) = \max_{q_1 \dots q_{t-1}} p(q_1 q_2 \dots q_t / \lambda) \quad (2.14)$$

Par induction on a :

$$\delta_{t+1}(j) = [\max_i \delta_t(i).a_{ij}] . b_j(o_{t+1}) \quad (2.15)$$

En conservant pour chaque  $t$  et chaque  $i$  l'état ayant amené au maximum  $\delta_t(j)$  et  $\phi_t(j)$ , on obtient l'algorithme de Viterbi.

### Algorithme de décodage de Viterbi

**Etape 1 :** Initialisation :

$$\delta_1(i) = \pi_i . b_i(o_1)$$

$$\Phi_1(i) = 0$$

**Etape 2 :** Récursion :

$$\delta_t(j) = \max_{1 \leq i \leq N} (\delta_{t-1}(i).a_{ij}) b_j(o_t) \quad 2 \leq t \leq T$$

$$\Phi_t(j) = \arg \max_{1 \leq i \leq N} (\delta_{t-1}(i).a_{ij}) \quad 1 \leq j \leq N$$

**Etape 3 :** Fin :

$$\hat{p} = \max_{1 \leq i \leq N} \delta_T(i)$$

$$\hat{q}_T = \arg \max_{1 \leq i \leq N} \delta_T(i)$$

**Etape 4 :** Meilleure séquence : (backtracking)

$$\hat{q}_t = \Phi_{t+1}(\hat{q}_{t+1}), \quad t = T - 1, T - 2, \dots, 1$$

Il convient de noter que l'algorithme de Viterbi est semblable (exceptée l'étape de backtracking) dans l'exécution au calcul Forward. La différence

principale concerne les états précédents d'excédent de maximisation dans qui est employée au lieu de la procédure d'addition d'induction de la procédure de Forward.

### 2.5.3. Solution du 3<sup>ème</sup> problème (Apprentissage)

On a vu précédemment les deux algorithmes classiques. Ces deux algorithmes supposent que l'on dispose d'un HMM construit et paramétré de manière à modéliser de façon satisfaisante les séquences que l'on souhaite traiter. La question qui nous préoccupe dans ce problème est celle de la construction d'un tel HMM.

On peut distinguer dans le problème de l'apprentissage d'un HMM deux cas distincts, suivant que la structure, le nombre d'états du HMM et les transitions autorisées - est connue ou ne l'est pas. Lorsque la structure est connue, le problème se réduit à un problème d'entraînement consistant à estimer les paramètres numériques - les distributions de probabilité de transition et de génération - de manière à expliquer au mieux les séquences d'apprentissage. Pour certaines applications, on ne dispose pas de connaissances suffisantes pour inférer naturellement la structure du HMM. L'apprentissage devient alors encore plus difficile. Il ne suffit plus de paramétrer une structure mais il faut également déduire cette structure, à partir des exemples fournis [4].

#### 2.5.3.1. Apprentissage à partir d'une structure connue

Nous nous intéressons dans cette section à l'apprentissage (ou entraînement) d'un HMM à partir d'une structure connue. Nous disposons pour cela d'une séquence d'apprentissage supposée représentative de la séquence que l'on souhaite modéliser. Une approche possible est, suivant le principe du maximum de vraisemblance, de chercher les paramètres  $\lambda = (\pi, A, B)$  du HMM qui maximisent la probabilité de génération de la séquence d'apprentissage. Soit  $O$  une séquence d'apprentissage. Notre but est alors de trouver les paramètres  $\lambda = (\pi, A, B)$  qui maximisent  $P(O/\lambda)$ . Une approche alternative au principe du maximum de vraisemblance est de chercher à maximiser la probabilité de génération de la séquence d'apprentissage suivant leur chemin de Viterbi. Le but est alors de trouver les paramètres  $\lambda = (\pi, A, B)$  qui maximisent  $P(O/\lambda, V)$ , avec  $V$

le chemin de Viterbi de la séquence  $O$  dans  $\lambda$ . Les paramètres ne sont donc pas estimés en maximisant la vraie probabilité de génération de la séquence d'apprentissage mais la probabilité de génération suivant le chemin le plus probable. Cette approche peut sembler moins rigoureuse que l'approche basée sur le principe du maximum de vraisemblance, mais elle est très utilisée en pratique et possède quelques arguments en sa faveur. D'abord, on peut se dire que si le HMM recherché est destiné à être appliqué à des problèmes de segmentation via la recherche des chemins de Viterbi, il semble judicieux de l'entraîner de cette manière. Ensuite, on remarque dans la pratique que la probabilité de génération d'une séquence de symboles suivant son chemin de Viterbi est en général beaucoup plus élevée que suivant n'importe quel autre chemin. Cette observation a conduit à l'hypothèse, appelée hypothèse de Viterbi, que tous les chemins, excepté le chemin de Viterbi, ont une probabilité nulle ou négligeable d'engendrer la séquence. Sous cette hypothèse, la probabilité de génération d'une séquence  $O$  par un HMM  $\lambda$  peut être approximée par la probabilité de génération de  $O$  suivant son chemin de Viterbi dans  $\lambda$ . On a alors :

$$P(O/\lambda) \approx P(O/\lambda, V) \quad (2.16)$$

Et maximiser  $P(O/\lambda, V)$  revient à maximiser une approximation de  $P(O/\lambda)$ . Le dernier argument en faveur de cette méthode d'entraînement est que l'on peut, dans un cas spécifique (entraînement de Baum-Welch), trouver la solution optimale au sens de la maximisation de  $P(O/\lambda)$  alors que l'on ne connaît aucune méthode optimale pour la maximisation de  $P(O/\lambda, V)$ . Abe et Warmuth ont étudié la question pour savoir s'il existe un algorithme d'entraînement efficace d'un automate probabiliste, dans le cadre d'une adaptation naturelle du paradigme de PAC-learning introduit par Valiant. Leur modèle est inspiré du modèle d'apprentissage non supervisé de Laird. La question est de savoir, pour un ensemble d'exemples de taille réduite, s'il existe un algorithme qui converge, avec une forte probabilité, vers une solution optimale. A notre connaissance, pour les HMMs, la question reste ouverte, mais nous pouvons penser que le problème est aussi difficile. Suivant la méthode d'estimation des paramètres choisis, il existe

deux heuristiques permettant l'entraînement des HMMs. Ces deux algorithmes, très similaires, sont tous deux issus d'une méthode générale servant à l'estimation des paramètres d'une grande famille de modèles probabilistes et appelée algorithme d'Expectation-Maximization (EM). Pour des raisons de simplicité, nous commencerons par exposer l'algorithme adapté à la maximisation de  $P(O/\lambda, V)$ .

### 2.5.3.2. Entraînement de Viterbi

Il existe un cas où il est possible de paramétrer la structure du HMM de manière à maximiser d'une façon optimale  $P(O/\lambda, V)$ . Ce cas est celui rencontré lorsque l'on connaît le chemin de Viterbi de la séquence d'apprentissage dans la structure. En effet, on peut alors associer à chaque état, chaque transition et chaque symbole attaché aux états du HMM le nombre de fois où ils sont utilisés pour générer la séquence d'apprentissage.

Soit  $n_s$ ,  $n_{s,s'}$  et  $n_s^o$ , respectivement le nombre de fois où l'état  $s$  est utilisé, le nombre de fois où la transition  $s \rightarrow s'$  est utilisée et le nombre de fois où le symbole  $o$  est généré par l'état  $s$  dans le chemin de Viterbi. Alors la  $P(O/\lambda, V)$  peut être réécrite :

$$P(O/\lambda, V) = \prod_{s \in S} \left( \prod_{o \in \Sigma} P(o/s)^{n_s^o} \prod_{s' \in S} P(s \rightarrow s')^{n_{s \rightarrow s'}} \right) \quad (2.17)$$

Maximiser cette formule revient à maximiser indépendamment chacun de ses sous-produits. On peut donc estimer les paramètres de A en maximisant l'expression :

$$\prod_{s' \in S} P(s \rightarrow s')^{n_{s \rightarrow s'}} \quad (2.18)$$

Les paramètres de B en maximisant :

$$\prod_{o \in \Sigma} P(o/s)^{n_s^o} \quad (2.19)$$

Les estimateurs de A et B sont donc respectivement :

$$\overline{P}(s \rightarrow s') = \frac{n_{s \rightarrow s'}}{n_s} \quad (2.20)$$

$$\overline{P}(o / s) = \frac{n_s^o}{n_s} \quad (2.21)$$

Cette méthode d'estimation n'est possible que dans le cas favorable où le chemin de Viterbi est connu. Lorsque ce n'est pas le cas, le problème est évidemment plus difficile. L'algorithme d'entraînement de Viterbi est un algorithme de reestimation itératif. Il consiste, à partir d'un paramétrage initial du HMM, à calculer le chemin de Viterbi de la séquence d'apprentissage à l'aide de l'algorithme de Viterbi. Le chemin de Viterbi est utilisé pour calculer, le nombre de fois où chaque transition, chaque état et chaque symbole attaché aux états sont utilisés. On réestime alors à l'aide des formules des estimateurs les paramètres du HMM, on paramètre la structure à l'aide de ces estimations et on réitère la procédure jusqu'à obtention de la stabilité.

#### Algorithme d'entraînement de Viterbi

##### Etape 1 : Données :

la séquence d'apprentissage.

$H \rightarrow$  la structure de l'HMM.

Choisir un paramétrage initial  $\lambda = (\pi, A, B)$  de  $H$  ;

##### Etape 2 : Répéter

Calculer le chemin de Viterbi de  $O$  suivant  $\lambda$  ;

Calculer les variables  $n_s$ ,  $n_{s,s'}$  et  $n_s^o$  de  $O$  ;

Ré estimer les paramètres  $\lambda$  de  $H$  en utilisant les formules des estimateurs;

Jusqu'à la stabilité du paramétrage;

On peut montrer que la valeur de  $P(O/\lambda, V)$  augmente à chaque itération et que l'algorithme converge vers un optimum local. Malheureusement, il existe généralement un grand nombre d'optima locaux et le paramétrage obtenu par ce processus dépend fortement du paramétrage initial choisi.

#### 2.5.3.3. Entraînement de Baum-Welch

L'algorithme d'entraînement de Baum-Welch est un algorithme qui cherche à estimer les paramètres  $\lambda = (\pi, A, B)$  du HMM en maximisant  $P(O/\lambda)$ . C'est un algorithme de réestimation itératif qui fonctionne sur le même principe que

l'algorithme d'entraînement de Viterbi. Dans ce dernier, on compte le nombre de fois où chaque état, transition et symbole attaché aux états sont utilisés dans les chemins de Viterbi. Ici on veut maximiser la probabilité de génération réelle, et non celle du chemin le plus probable. On va donc associer aux états, transitions et symboles le nombre de fois où ils sont utilisés pour la séquence et le chemin susceptible de générer la séquence, par la probabilité du chemin. Ces comptes pondérés sont alors utilisés pour réestimer les paramètres du modèle de la même manière que pour l'algorithme d'entraînement de Viterbi .

De manière plus formelle, considérons la séquence d'observations  $O = o_1, \dots, o_T$ . Notre but est de trouver les paramètres  $\lambda = (\pi, A, B)$  qui maximisent la probabilité  $P(O/\lambda)$  de générer  $O$  suivant  $\lambda$  [7].

Afin de décrire l'algorithme pour la réestimation des paramètres du HMM, nous définissons d'abord  $\xi_t(i, j)$ , la probabilité d'arriver à l'état  $S_i$  à l'instant  $t$ , et à l'état  $S_j$  à l'instant  $t+1$ , suivant le modèle  $\lambda$  et la séquence d'observations  $O$  :

$$\xi_t(i, j) = p(q_t = S_i, q_{t+1} = S_j / O, \lambda) \tag{2.22}$$

L'ordre des événements menant aux conditions exigées par  $\xi_t(i, j)$  est illustré dans la figure 2.5.

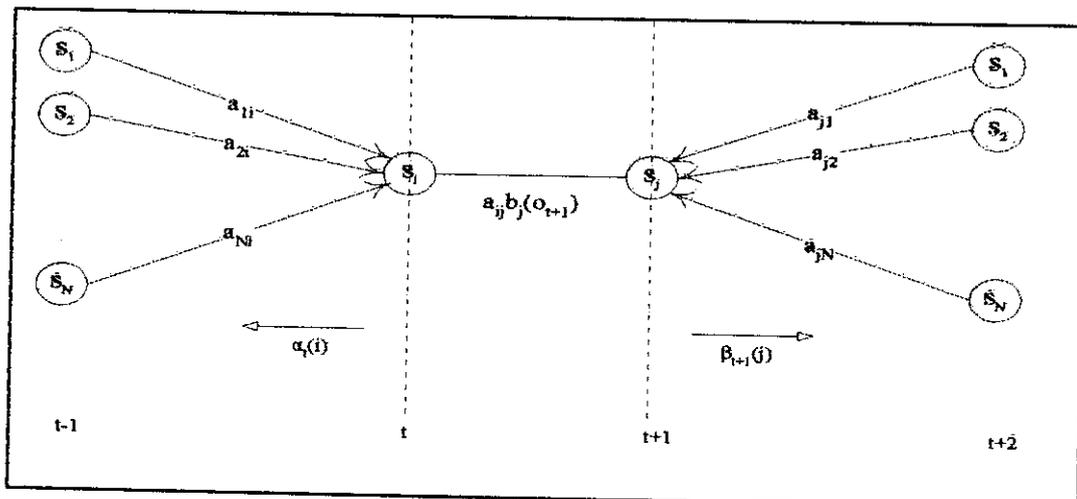


Figure 2. 5 : Illustration de l'ordre des opérations requises pour le calcul de l'événement commun que le système est dans d'état  $S_i$ , à l'instant  $t$  et à l'état  $S_j$  à l'instant  $t+1$  [7].

Il devrait être clair, de la définition des variables Forward et Backward, que nous pouvons écrire  $\xi_t(i, j)$  sous la forme :

$$\begin{aligned} \zeta_t(i, j) &= \frac{p(q_t = S_i, q_{t+1} = S_j, O / \lambda)}{p(O / \lambda)} \\ &= \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^N \sum_{j=1}^N \alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)} \end{aligned} \quad (2.23)$$

Considérons également la variable  $\gamma_t(i)$  :

$$\gamma_t(i) = p(q_t = S_i, O / \lambda) \quad (2.24)$$

Cette équation exprime la probabilité de générer O suivant  $\lambda$  toute en se trouvant sur l'état  $S_i$  à l'instant t, exprimée en fonction de  $\xi_t(i, j)$  :

$$\gamma_t(i) = \sum_{j=1}^N \zeta_t(i, j) \quad (2.25)$$

Si nous additionnons  $\gamma_t(i)$  en fonction de t, nous obtenons une quantité qui peut être interprétée comme le nombre de fois que l'état  $S_i$  a été visitée, ou d'une manière équivalente, le nombre prévu de transition fait à partir de l'état  $S_i$  (si nous excluons le cas de l'instant  $t=T$  de l'addition). De même, l'addition de  $\xi_t(i, j)$  en fonction de  $\xi$  (de  $t=1$  à  $T-1$ ) peut être interprétée comme le nombre prévu de transitions de l'état  $S_i$  vers l'état  $S_j$ .

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{NET}_{S_i \rightarrow S_j} \quad (2.26)$$

Le Nombre Espéré de Transitions depuis  $S_i$ , sachant l'observation O et le modèle  $\lambda$ .

$$\sum_{t=1}^{T-1} \zeta_t(i, j) = \text{NET.depuis } S_i \quad (2.27)$$

Le Nombre Espéré de Transitions depuis  $S_i$  vers  $S_j$ , sachant l'observation O et le modèle  $\lambda$ .

En utilisant les formules ci-dessus, nous pouvons donner une méthode pour la réestimation des paramètres d'un HMM. Un ensemble de formules raisonnables de réestimation pour A, B et  $\pi$  sont :

$$\begin{aligned}\bar{\pi}_i &= \text{Nbr Espéré de fois où tps 1 on est en } S_i \\ &= \gamma_1(i)\end{aligned}\quad (2.28)$$

$$\begin{aligned}\bar{a}_{ij} &= \frac{\text{NETS } S_i \rightarrow S_j}{\text{NETdepuis } S_i} \\ &= \frac{\sum_{t=1}^{T-1} \zeta_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)}\end{aligned}\quad (2.29)$$

$$\begin{aligned}\bar{b}_j(k) &= \frac{\text{NE de fois où on est en } S_j \text{ et on OBS } V_k}{\text{NE de fois où on est ds } S_j} \\ &= \frac{\sum_{t=1; o_t=v_k}^T \gamma_t(j)}{\sum_{t=1}^T \gamma_t(i)}\end{aligned}\quad (2.30)$$

Si nous définissons le modèle courant comme  $\lambda = (\pi, A, B)$  et nous l'employons pour calculer les parties droite de (2.23) et (2.24), et nous définissons le modèle réestimé comme  $\lambda' = (\pi', A', B')$  déterminé à partir des côtés gauches de (2.23) et (2.25), alors il a été prouvé que le modèle  $\lambda'$  est plus probable que le modèle  $\lambda$  dans le sens où  $P(O/\lambda') \geq P(O/\lambda)$ , c'est-à-dire, que nous avons trouvé un nouveau modèle  $\lambda'$  dans lequel la séquence d'observation sera mieux produite.

Si nous employons itérativement le  $\lambda'$  au lieu du  $\lambda$  et répétons le calcul de réestimation, nous pouvons alors améliorer la probabilité de O étant observé du modèle jusqu'à ce qu'un certain point limiteur soit atteint. Le résultat final de cette procédure de réestimation est appelé une évaluation de maximum de vraisemblance du HMM. Il devrait être précisé que l'algorithme de Baum-Welch converge vers un maximum local seulement, et que dans la plupart des problèmes d'intérêt, la surface d'optimisation est très complexe et elle a beaucoup de maximums locaux [7].

Les formules de réestimation de  $\pi_i$  et  $a_{ij}$  peuvent être dérivées directement en maximisant les fonctions auxiliaires de Baum :

$$A(\lambda, \lambda') = \sum_{q \in Q} p(O, q / \lambda') \log p(O, q / \lambda) \quad (2.31)$$

En fonction de  $\lambda'$ . Il a été prouvé par Baum et ses collègues que la maximisation de  $A(\lambda, \lambda')$  mène à la probabilité accrue  $P(O/\lambda') \geq P(O/\lambda)$  - Finalement la fonction convergent vers un point critique.

Posons,  $q = q_1 q_2 \dots q_T$  et  $O = o_1, o_2, \dots, o_T$ .

$$P(O, q / \lambda) = \pi_{q_1} b_{q_1}(o_1) \prod_{t=2}^T a_{q_{t-1} q_t} b_{q_t}(o_t) \quad (2.32)$$

On peut décomposer notre fonction auxiliaire en 3 termes indépendants d'où :

$$\begin{aligned} A(\lambda, \lambda') = & \sum_{q \in Q} (\log[\pi_{q_1}] p(O, q / \lambda) + \log[b_{q_1}(o_1)] p(O, q / \lambda)) + \\ & \sum_{q \in Q} (\sum_{t=2}^T \log a_{q_{t-1} q_t}) p(O, q / \lambda) + \\ & \sum_{q \in Q} (\sum_{t=2}^T \log b_{q_t}(o_t)) p(O, q / \lambda) \end{aligned} \quad (2.33)$$

Pour connaître l'estimée des  $\pi_i$  (les probabilités initiales), il suffit de dériver le premier terme (par rapport à chaque  $\pi_i$ ) et à résoudre à 0 (de même pour les autres paramètres), maximiser (sur  $\pi_i$ ) le premier terme est équivalent à maximiser seulement :

$$\sum_{q \in Q} \log \pi_{q_1} p(O, q / \lambda') \quad (2.34)$$

Cette formule revient à maximiser :

$$\sum_{i=1}^N \log \pi_i p(O, q_1 = i / \lambda') \quad (2.35)$$

Nous tenons compte de la contrainte :

$$\sum_{j=1}^N \pi_j = 1 \quad (2.36)$$

On peut intégrer cette contrainte dans le terme à maximiser en introduisant un multiplicateur de Lagrange (ici appelé  $\mu$ ) :

$$\frac{\delta}{\delta \pi_i} \left( \sum_{i=1}^N \log \pi_i p(O, i / \lambda') - \mu \left( \sum_{j=1}^N \pi_j - 1 \right) \right) = 0 \quad (2.37)$$

$$\text{Soit : } \pi_i = \frac{p(O, q_1 = i / \lambda')}{\mu} \quad \forall i \in [1, N]$$

$$\text{Où : } \sum_{i=1}^N \pi_i = 1 = \sum_{i=1}^N \frac{p(O, q_1 = i / \lambda')}{\mu} \Rightarrow \mu = \sum_{i=1}^N p(O, q_1 = i / \lambda')$$

$$\text{D'où : } \pi_i = \frac{p(O, q_1 = i / \lambda')}{\sum_{i=1}^N p(O, q_1 = i / \lambda')} = \gamma_1(i) \quad (2.38)$$

On peut alors proposer une version complète de l'algorithme de Baum-Welch. Comme pour l'algorithme d'entraînement de Viterbi, on peut prouver que la probabilité de génération des séquences  $P(O/\lambda)$  augmente à chaque itération et que l'algorithme converge vers un optimum local. Néanmoins, dans ce cas, l'optimum local n'est jamais atteint et il est nécessaire de définir un critère d'arrêt - par exemple en stoppant la procédure lorsque l'augmentation de  $P(O/\lambda)$  est inférieure à un certain seuil, où lorsque le nombre d'itérations effectuées est jugé suffisamment important [4].

**Algorithme d'entraînement de BaumWelch****Etape 1 : Données :**

O → la séquence d'apprentissage.

H → la structure de l'HMM.

Choisir un paramétrage initial  $\lambda = (\pi, A, B)$  de H ;

**Etape 2 : Répéter**

Calculer  $\alpha_t(i)$  à l'aide de l'algorithme Forward;

Calculer  $\beta_t(i)$  à l'aide de l'algorithme Backward;

Calculer  $\xi_t(i, j)$ ;

Calculer  $\gamma_t(i)$ ;

Re-estimer les paramètres  $\lambda$  de H en utilisant les formules (2.23), (2.24) et (2.25);

Jusqu'au critère d'arrêt;

**Notes sur la procédure de réestimation**

Un aspect important de la procédure de réestimation est que les contraintes stochastiques des paramètres du HMM, sont automatiquement satisfaits à chaque itération [08] :

- $\sum_{i=1}^N \bar{\pi}_i = 1$
- $\sum_{j=1}^N \bar{a}_{ij} = 1 \quad 1 \leq i \leq N$
- $\sum_{k=1}^M \bar{b}_j(k) = 1 \quad 1 \leq j \leq N$

En conclusion, nous notons que puisque le problème entier peut être vu comme problème d'optimisation, des techniques standard de gradient peuvent être employées pour trouver les valeurs optimales des paramètres du modèle [4]. De telles procédures ont été essayées et montrées pour rendre la solution comparable à celles des procédures standard de réestimation.

**2.5. Topologie des HMMs**

Nous avons seulement considéré le cas général des HMMs. Nous allons intéressé aux différentes types d'interconnexions entres états (figure 2.6.(a), (b) et (c)).

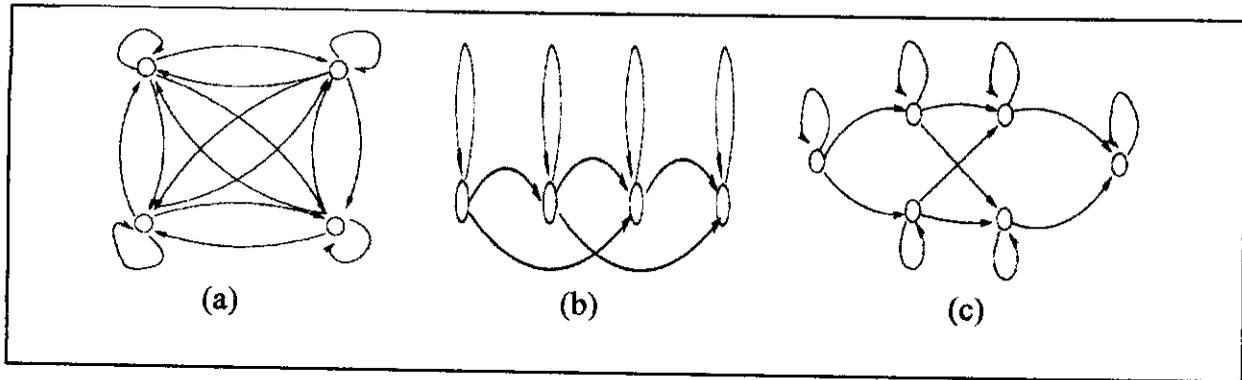


Figure 2.6 : Les trois modèles du HMM [7].

- (a) : Modèle ergodique à 4 états.
- (b) : Modèle gauche-droite à 4 états.
- (c) : Modèle gauche-droite avec chemins parallèles

Commençons par un cas spécial de HMM, appelé le modèle ergodique qui est entièrement relié et dans lequel chaque état du modèle pourrait être atteint par n'importe quel autre état du modèle (Proprement parler, un modèle ergodique a la propriété que chaque état peut être atteint de chaque autre état dans un nombre fini d'étapes), pour un modèle composé de  $N=4$  états, ce type de modèle a la propriété que chaque coefficient  $a_{ij}$  est positif. Par conséquent, pour l'exemple de (a) nous avons :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \quad (2.39)$$

Pour certaines applications, d'autres types de HMMs se sont avérés pour expliquer les propriétés observées du signal le mieux modèle par rapport au modèle ergodique standard. (a). Ce modèle s'appelle un modèle gauche-droite parce que la séquence fondamentale d'états associée avec le modèle, a la propriété que chaque fois que le temps augmente l'indice d'état augmente (ou reste le même), c'est-à-dire les états procèdent de gauche à droite.

Le type gauche-droite du HMM est bien utilisé pour modéliser les signaux dont les propriétés changent pendant des périodes de temps finies (par exemple,

la parole). La propriété fondamentale de ce type du HMM est que les coefficients de transition d'état ont la propriété :

$$a_{ij} = 0, \quad j < i \quad (2.40)$$

C'est-à-dire, on ne permet aucune transition aux états dont les indices sont inférieurs à l'indice de l'état actuel. En outre, les probabilités d'état initial ont la propriété :

$$\pi_i = \begin{cases} 0, & i \neq 1 \\ 1, & i = 1 \end{cases} \quad (2.41)$$

Puisque la séquence d'état doit commencer dans l'état 1 et finir dans l'état N. Souvent, avec les modèles gauche-droite, des contraintes additionnelles sont placées sur les coefficients de transitions d'états pour s'assurer que les grands changements des indices d'état ne se produisent pas, par conséquent une contrainte de la forme :

$$a_{ij} = 0, \quad j > i + \Delta$$

Cette contrainte est souvent employée, en particulier, pour l'exemple du (b). La valeur de  $\Delta$  est 2, c'est-à-dire, aucun saut de plus de 2 états n'est permis, d'où la forme de matrice de transition d'état pour lequel :

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{bmatrix} \quad (2.42)$$

Il devrait être clair que, pour le dernier état dans un modèle gauche-droite, les coefficients de transitions d'état s sont indiqués comme suit :

$$a_{N\bar{N}} = 1 \quad (2.43)$$

$$a_{Ni} = 1, \quad i < N \quad (2.44)$$

Bien que nous ayons dichotomisé les HMMs en modèles ergodique et gauche-droite, il y a beaucoup de variations possibles et de combinaisons

possibles. A titre d'exemple, (c) Montre une interconnexion parallèle de deux HMMs de type gauche droite.

Ce modèle est un modèle gauche-droite (il obéit à toutes les contraintes de  $a_{ij}$ ), cependant, on peut voir qu'il a certaine flexibilité non actuelle dans un modèle gauche-droite strict (c'est-à-dire, sans chemins parallèles).

Il est à remarquer que l'imposition des contraintes du modèle gauche-droite, n'ont essentiellement aucun effet sur la procédure de réestimation. C'est le cas puisque n'importe quel paramètre d'un HMM à la valeur zéro au commencement, demeurera à zéro dans toute la procédure de la réestimation.

## 2.6. Conclusion

Nous avons présenté dans ce chapitre une vue d'ensemble des HMMs, de leurs applications et des algorithmes classiques utilisés dans la littérature : les algorithmes de calcul de la probabilité de génération d'une séquence par un HMM, l'algorithme de recherche du chemin optimal, ainsi que les algorithmes d'entraînement et quelques pistes pour l'apprentissage de la structure.

Remarquons pour conclure, que les HMMs sont des modèles de référence pour la reconnaissance automatique de texte et de la parole.

La manipulation des HMMs permet de :

- Calculer la probabilité d'une séquence ;
- Déterminer le chemin de Viterbi ;
- Apprendre d'un HMM (Entraînement de Viterbi ou de Baum-welch).

# Chapitre 3 :

## Application des HMM à la RAP

### 3.1. Introduction

L'utilisation des HMM dans un système de RAP implique nécessairement l'ajustement de quelques paramètres et l'optimisation de certains critères.

Le but est de mettre en place les algorithmes nécessaires permettant l'élaboration d'un système de reconnaissance des phonèmes de l'AS en utilisant les HMMs. Nous exposons en détail notre système, de l'élaboration du corpus jusqu'à l'étape de décision (reconnaissance).

### 3.2. Présentation du système de RAP

Un système de RAP consiste, de façon générale, à transcrire le signal de la parole en symbole.

Notre système de reconnaissance comporte les modules illustrés dans le schéma suivant (Figure 3.1).

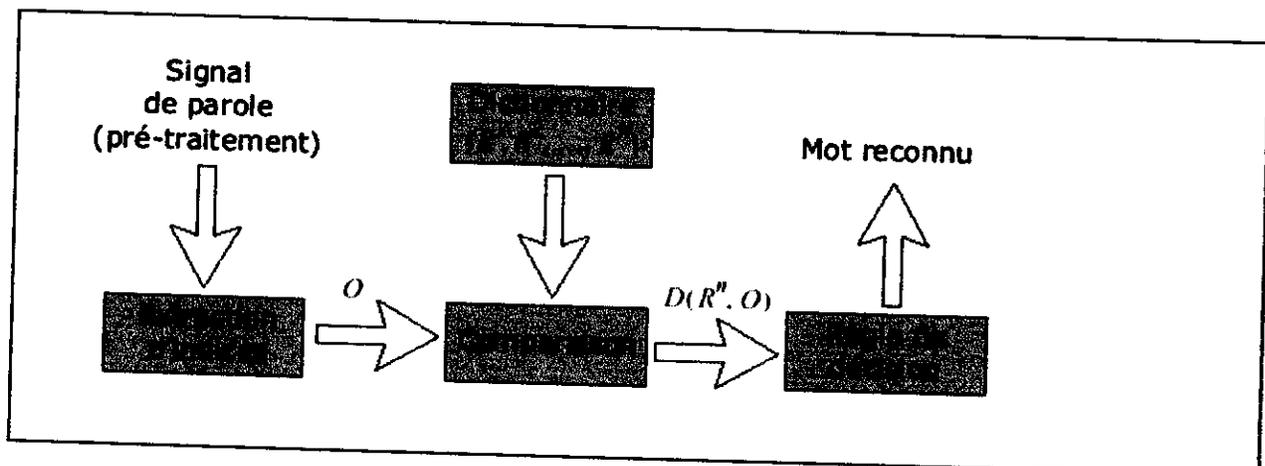


Figure 3.1 : Système de reconnaissance automatique de la parole

#### 3.2.1. Phase d'apprentissage

Constitution du dictionnaire de références Acoustiques  $\{ R^1, R^2, \dots, R^N \}$  ; chaque mot du vocabulaire est représenté par une ou plusieurs références.

#### 3.2.2. Phase de reconnaissance

La décision du système s'effectue à partir des dissemblances  $D$  entre les références du dictionnaire  $R^n$  et l'image du mot à identifier  $O$ .

### 3.3. Acquisition des données

L'acquisition des données consiste à enregistrer les phrases du corpus choisis. Les enregistrements ont été effectués par un locuteur Jordanien.

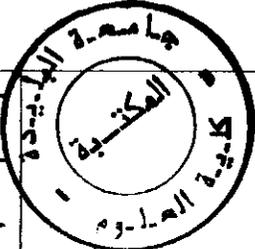
### 3.4. Elaboration du corpus

La première étape à effectuer avant d'entamer les traitements, c'est l'élaboration du corpus d'apprentissage et de test. Le choix de ce dernier n'est pas arbitraire, afin de tester les performances de notre système de reconnaissance. En ce qui concerne notre travail nous avons pris un corpus constitué de phrases porteuses contenant tous les phonèmes de l'AS pris dans différentes positions (initiale, médiane et finale). Nous justifions le choix de ce type de corpus par le fait qu'il est préférable de reconnaître les phonèmes dans des contextes réels pour prendre en en considération les effets de la coarticulation (Tab.3.1).

Tab.3.1 : Corpus utilisé.

-وَحاولت.	-أَسعدُ زَوْجِين.
- إني أشعرُ هزةً قَويةً.	-جِلسِ يَستمعُ إلى الراديو.
-أرضِ المَطار.	-و يَضافُ إليه اليَصل.
-وكثيرٌ من الصِناعاتِ الخفيفةِ، وأنشئت.	-في ما جاء في برنامجِ المرأةِ في ذلك.
-على تحسِينِ وسائلِ النقلِ والمواصلاتِ	-ظهورِ الإسلامِ. سَيَما.
وإنشاءِ المَطاراتِ.	-فاذا بِصوتِ جَميلٍ.
-في صَحبتهِ.	-يوضَعُ اللَّحمُ في القَدْرِ ثم يَغطى.
-ونظرةِ الحِيبِ تشع.	-ولم يَستطعُ أن يَصبِرَ فقامَ إلى أهلهِ يقولُ
- على الغِربةِ، كلِّ هذا.	لِهم.
-و تنظيمِ توزيَعها، فتلجأُ إلى بناءِ السُدودِ.	-لا يَد لي من الزَواجِ بِهذهِ المرأةِ.
-أهم رِوادِ النهضةِ الحديثَةِ.	-أن يَعرِفَ طَريقها إلى المَعدةِ أو لا.
-نذِكرُ في مَقدمتهم رِفاعَةَ رافعِ الطِيطاوي.	-وانتَظرِ سَاعةً ثم سَاعةً وسَاعةً حتى تَعبُ
-يطرسُ البِستانيِ وولدِ.	من الانتِظارِ.
-التي يَستَكنها الفِلاحون.	-من المَطحِبِ والعَرَقِ يَسيلُ من وِجْهها وقالت

-تسعين.	له أسفة
- تكوينه	- عملت لك بيضا مقلبا خوفا.
-مجموع.	فوجده قد احترق و تحجر
- رأينا.	-أسرعت الزوجة إلى الخروج كأنها على
-عمر الولايات.	موعد.
-نصيب كبير, الدين.	-تقع الأقطار العربية عند ملتقى ثلاث قارات
-تونس.	أوروبا و آسيا وأفريقيا وتمتد سواحلها.
-فإننا نلاحظ.	-والبحر الأبيض المتوسط وبحر العرب
-أن تملك قلبه.	و على المحيطين الأطلسي والهندي.
- فإذا ملكتها.	-وتمتد جذور هذه الحضارة بعيدا في التاريخ
-قل هو الله أحد, وظلت.	كما نرى في آثار.
-وكانت الغاية.	-العراق والفينيقيين في لبنان والرومان في
-من سور القرآن تدخّل في نفسها الطمأنينة	سوريا والأردن.
فلم تتذكر.	-فمع أن الأتراك والعرب قد عاشوا.
-والعلوم المختلفة.	-لمذهب ديني.
-تشجيع.	-والأرز في مصر والعراق والقمح في سوريا
-مراكز.	و ينتج التبغ في الجزائر والتمور.
-في شبه.	-تشكل الزراعة موردا أساسيا في أكثر
-والتاريخ كذلك لا يصلح وحده أن يكون	الأقطار العربية, وتقوم على أساس.
الأساس الوحيد للقومية.	-تستغل مياه النيل أحسن استغلال.
-نهر الدانوب.	-فراش الموت.
-هذا الشعور.	-ضعيف, أبيض.
-خاصة للبنات.	-المتفوقين للدراسة على حسابها.
-الوزارات, غير. صاح, الرشيد	-الروح التي, أن يكون جوار.
-جامع الزيتونة تونس.	-نسبة المسيحيين.
-قطعة واحدة, من الأمم.	-حيث مي زيادة.
-شعور, وحدة المركز.	-خارج, روح التعاون.

	<p>- المشهور المسؤول. - وأربعة عشر. - ولم يمض في ظلها. - شوق. - الذكور من ذريتهم. - دخول. - اللذبة. - منذ ملك. - خلال. - المصري ليقاتلوا.</p>	<p>- وما يجب ذكره موضوعات صحيفة. - ومن أهم الصحف. - حكام العرب الكاظمين. - مختلف العصور. - الشعراء البارزون و المترجمون. - عاش أبناؤها محاطين. - نشوء حزب البعث العربي الاشتراكي، ويضم. - إيذاء إبراهيم، قال له أبوه. - فتنت. - يعقوب.</p>
--	---	--

### 3.5. Segmentation

Une fois le corpus des phrases a été enregistré, nous passons à la phase de segmentation.

La phase de segmentation joue un rôle très important dans le système de reconnaissance vocale et nécessite un intérêt particulier de notre part.

La *segmentation* du signal de parole vise à extraire du continuum acoustique des unités sur lesquelles portera la "décision", c'est-à-dire le décodage proprement dit.

Le problème est donc double :

- Choisir les unités de décision;
- Opérer une segmentation correcte.

Le choix des unités est vaste: phones, **phonèmes** (la plus petite unité présente dans la parole et susceptible par sa présence de changer la signification d'un mot), diphones, syllabes, mots, etc.

De soudains changements dans le spectre du signal ou dans son amplitude sont souvent le signe de frontières de segments. Néanmoins, ces frontières ne sont pas des indices fiables à cause du phénomène de coarticulation.

la segmentation du corpus phonémique a été effectuée manuellement (ce qui justifie le temps, relativement long, alloué à cette opération) en utilisant le logiciel PRAAT.

Après chaque segmentation nous effectuons des tests par écoute afin de s'assurer que la segmentation a été bien faite.

### 3.6. Description du logiciel PRAAT

*Praat* est un programme informatique qui permet de mener des analyses phonétiques, de faire de la synthèse de la parole et de manipuler des données (analyses statistiques, construction de grammaires, etc.).

La fenêtre d'analyse affiche une barre de menus, le signal acoustique, une fenêtre d'analyse (fréquence fondamentale, courbe d'intensité, spectrogramme, détection automatique des formants). A gauche et à droite des fenêtres du signal et de l'analyse, on peut voir les valeurs de fréquence et d'intensité tandis qu'en haut et en bas de ces fenêtres, on peut voir les valeurs de temps (Figure 3.2)

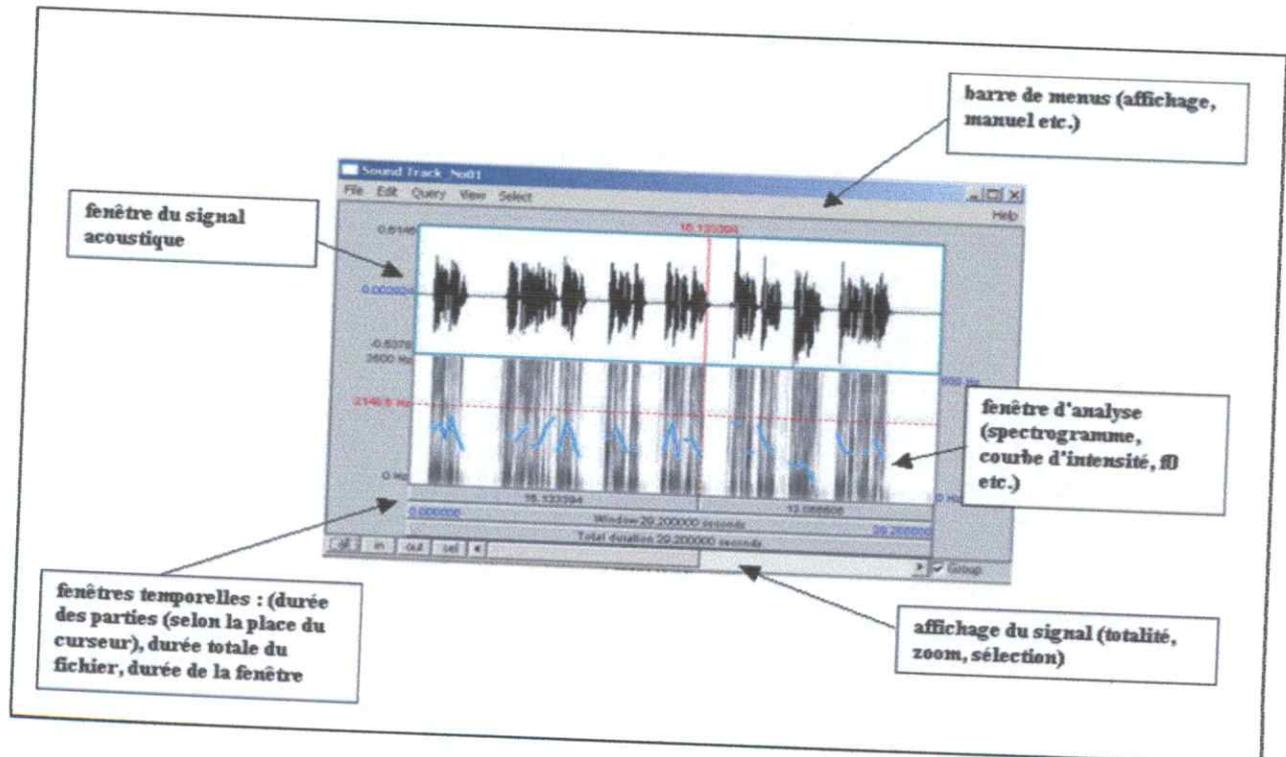


Figure 3.2 : Différentes fonctionnalités du PRAAT.

### 3.7. Extraction des formants

Chaque phrase prononcée est visualisée au moyen d'un éditeur de signaux pourvu d'un sonographe (PRAAT) qui nous donne la représentation temps, fréquence et énergie (sonagramme). De bonnes connaissances phonétiques sont nécessaires pour effectuer un étiquetage phonétique c'est-à-dire une opération qui consiste à identifier et à reconnaître sur le sonagramme chaque phonème du mot porteur.

On peut visualiser les formants qui apparaissent comme des points rouge, il suffit de pointer avec le curseur sur le point rouge pour avoir la valeur du formant dans l'axe des fréquences à gauche (Figure 3.3)[10].

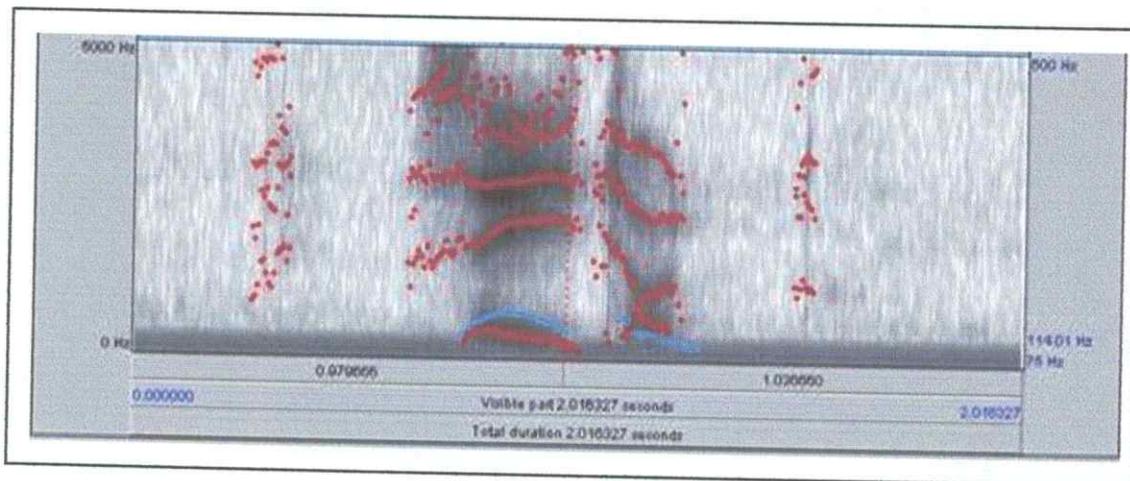


Figure 3.3 : Sonagramme et formants du mot زيت.

### 3.8. Phonèmes étudiés

Le système phonétique de l'AS soit constitué de 28 consonnes et 6 voyelles (3 brèves et 3 longues), nous nous sommes étudiés toutes ces consonnes et voyelles.

Comme nous l'avons cité auparavant, la variation du signal de la parole dépend d'un ensemble considérable de facteurs phonologiques, syntaxiques, etc. En ce qui concerne notre étude, nous nous sommes intéressés et limités à un paramètre jugé dans la variation des phonèmes arabes tels que :

- la position de la consonne dans le mot (initial, médiane ou finale) ;

- la position de la voyelle dans le mot (initial, médiane ou finale) ;
- la classe de la consonne (fricative, occlusive, ...).

### 3.9. Durées moyennes

A partir des données mesurées, un calcul des durées moyennes ( $D_{moy}$ ) et des écart-types a été réalisé pour quelques consonnes et pour les voyelles brèves dans différents contextes [11] afin de prendre en considération ces résultats (tab 3.2) dans l'étape de la segmentation.

Tab 3.2 : Les durées moyennes (en ms) et les écart-types de quelques consonnes [11].

Phonème	transcription des arabisants	$D_{moy}$	Ecart-type
[ش]	[ʃ]	134.70	9.58
[ط]	[t]	115.47	7.45
[ض]	[d]	100.89	6.78
[ظ]	[d]	98.60	8.29
[س]	[s]	131.95	10.07
[ت]	[t]	112.57	9.73
[د]	[d]	98.02	8.25
[ذ]	[d]	98.47	9.36
[ز]	[z]	100.67	7.13
[ث]	[t]	124.58	7.57
[غ]	[g]	76.40	12.24
[ء]	[ʔ]	66.50	13.02
[ح]	[h]	120.96	4.84
[خ]	[h]	118.33	10.36
[ق]	[q]	98.84	16.75
[ج]	[g]	92.11	14.83
[ ]	[a]	97	8.39
[ ]	[u]	84	8.15
[ ]	[i]	80	9.55

### 3.10. Procédure d'apprentissage du modèle

Rappelons qu'un HMM est paramétré par :  $A, B, \pi, N, M$

On regroupe ces paramètres dans une structure de données, ce qui donne la structure suivante :

```
{
  N : entier ; \ Le nombre d'états
  M : entier ; \ nombre de symboles (formants)
   $\pi = a_0$  : Tableau [N] élément de type réel ; \ le vecteur de distribution
  initiale de probabilités de transitions ( $a_0$ )
  A : Tableau [N] [N] élément de type réel ; \ La matrice de distribution
  de probabilités de transitions d'états
  B : Tableau [N] [M] élément de type réel ; \ La matrice de distribution
  de probabilités de générations des symboles par états
}
```

Un phonème est émis par une suite d'état constituant un certain parcours entre l'état initial et final. L'algorithme de Forward-Backward permet de calculer la probabilité  $P(O/\lambda)$  de production de la séquence d'observation  $O$  sachant le modèle  $\lambda(A, B, \pi)$ . L'algorithme de Baum Welch permet de construire un modèle markovien associé à un phonème, à l'aide de l'ensemble d'élocution du phonème selon sa position dans le mot, ce qui permet de ré estimer les nouvelles valeurs  $A, B, \pi$ .

#### 3.10.1 Génération des observations dans un HMM

##### Etape 1 :

Poser  $t=1$  Choisir un état initial  $S_t=q_i$  avec une distribution  $a_{0i}$  ou  $1 \leq i \leq N$   
Poser  $i=1$

##### Etape 2 :

Choix de l'observation  $O_t=v_k$  avec une probabilité  $b_j(k)$  ou  $1 \leq k \leq M ; 1 \leq j \leq N$ .

##### Etape 3 :

Transition au nouvel  $S_{t+1}=q_j$  avec une probabilité  $a_{ij}$  avec  $1 \leq j \leq N$ .

##### Etape 4 : $t=t+1$ .

Si  $t \leq T$  ; aller a l'étape 2.  
si non fin.

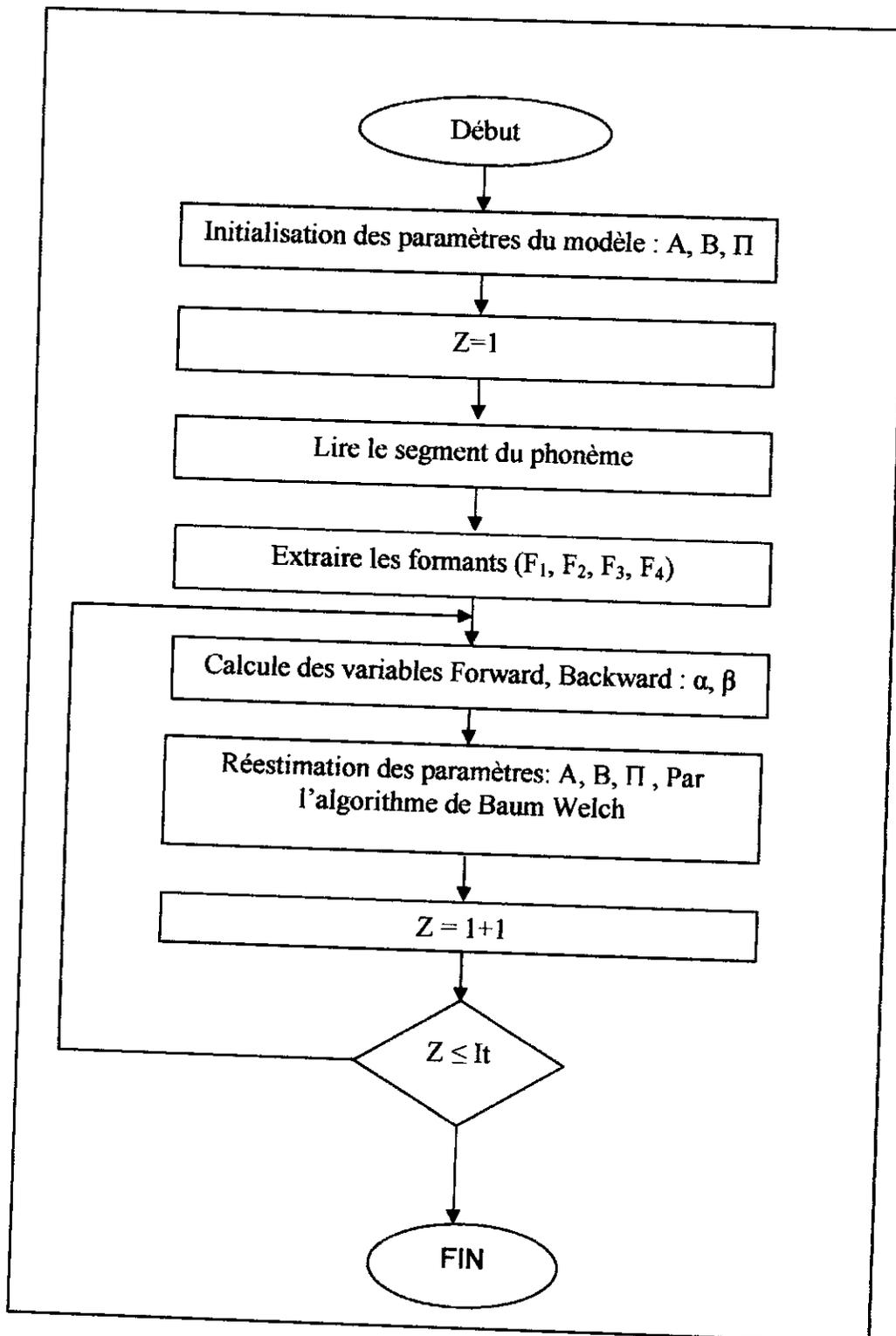
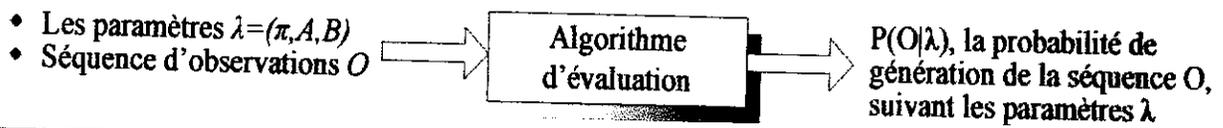


Figure 3.4 : Organigramme de la procédure d'apprentissage.

It : Nombre d'itérations.

### 3.10.2. Algorithmes d'évaluation (solution au 1<sup>er</sup> problème)



#### 3.10.2.1. Algorithme de Forward

Pour réaliser le calcul Forward il faut développer ses trois principales étapes :

##### Procédure Forward

##### Var :

$i, j, t$  : entier;  
Som, Probat : réel;  
alpha : Tableau [T][N] élément de type réel ;

Début // la séquence d'observation O, donnée.

//Etape1 :l'initialisation.

pour  $i \leftarrow 1$  à N  
alpha[1,i]  $\leftarrow a_0[i] * B[i, O[1]]$ ;

//Etape 2 : l'induction.

pour  $t \leftarrow 1$  à T-1

faire

pour  $j \leftarrow 1$  à N

faire

S  $\leftarrow 0$ ;

pour  $i \leftarrow 1$  à N

Som  $\leftarrow$  Som + alpha[t,i] \* A[i,j];

alpha[t+1,j]  $\leftarrow$  S\*B[j, O[t+1]];

fait;

fait;

//La Terminaison.

Probat  $\leftarrow 0$ ;

pour  $i \leftarrow 1$  à N

Probat  $\leftarrow$  Probat + alpha[T,i];

Fin; // le résultat = Probat.

### 3.10.2.2. Algorithme de Backward

Pour réaliser le calcul Backward, il faut également développer ses trois principales étapes :

#### Procédure Backward

##### Var :

i, j, t : entier;  
Som, Probat : réel;  
beta : Tableau [T][N] élément de type réel ;

Début // la séquence d'observation O, donnée.

//Etape1 :l'initialisation.

pour i←1 à N  
  beta[T,i] ←1;

//Etape2 :l'induction.

pour t←T-1 à 1

  faire

    pour i←1 à N

      faire

        Somme←0;

        pour j←1 à N

          Som←Som + A[i,j] \* B[j,O[t+1]] \* beta[t+1,j];

        beta[t,i] ←Som;

      fait;

    fait;

//La terminaison.

Probat←0;

pour i←1 à N

  Probat←Probat + pi[i] \* beta[1,i] \* B[i,O[1]];

Fin; // le résultat = Probat.

### 3.10.3. Algorithme de Viterbi (solution du 2<sup>ème</sup> problème)

Il faut développer les quatre principales étapes de l'algorithme :

- l'étape d'initialisation ;
- l'étape de l'induction ;
- l'étape de terminaison ;
- l'étape de rétro propagation (Backtracking).

**Procédure Viterbi****Var :**

$i, j, t, \text{IndMax}$  : entier;  $\text{MaxP}$  : réel;  
 $\text{delta}$  : Tableau  $[T][N]$  élément de type réel ;  
 $\text{psi}$  : Tableau  $[T][N]$  élément de type entier ;

**Début** // la séquence d'observation  $O$ , donnée.

//Etape1 : Initialisation.

pour  $i \leftarrow 1$  à  $N$

$\text{delta}[1, i] \leftarrow \text{pi}[i] * B[i, O[1]]$ ;

//Etape2 : Induction.

pour  $t \leftarrow 2$  à  $T$

faire

pour  $j \leftarrow 1$  à  $N$

faire

$\text{MaxP} \leftarrow 0; \text{IndMax} \leftarrow -1$ ;

pour  $i \leftarrow 1$  à  $N$

faire

Si  $\text{MaxP} < \text{delta}[t-1, i] * A[i, j]$

alors

$\text{MaxP} \leftarrow \text{delta}[t-1, i] * A[i, j]$ ;

$\text{IndMax} \leftarrow i$  ;

fsi;

fait;

$\text{delta}[t, j] \leftarrow \text{MaxP} * B[j, O[t]]$ ;

$\text{Psi}[t, j] \leftarrow \text{IndMax}$  ;

fait;

fait;

// Etape3 : Terminaison.

$\text{MaxP} \leftarrow 0; V[T] \leftarrow -1$  ;

pour  $i \leftarrow 1$  à  $N$

faire

Si  $\text{MaxP} < \text{delta}[T, i]$

alors

$\text{MaxP} \leftarrow \text{delta}[T, i]$ ;

$V[T] \leftarrow i$  ;

fsi;

fait ;

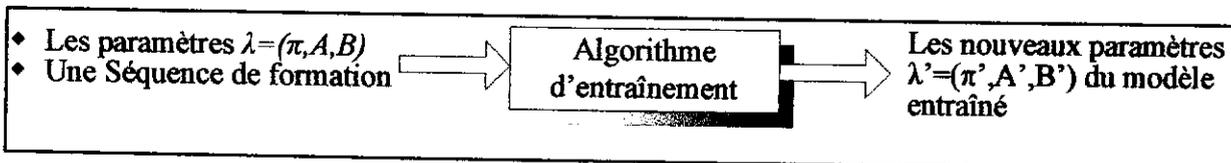
// la rétro propagation (Backtracking).

pour  $t \leftarrow T-1$  à  $1$

$V[t] \leftarrow \text{psi}[t+1, V[t+1]]$  ;

**Fin**; // le résultat =  $V[T]$ , le chemin de Viterbi.

### 3.10.4. Algorithmes d'apprentissage (résolution au 3<sup>ème</sup> problème)



#### 3.10.4.1. Algorithme de Baum Welch

Avant de présenter l'algorithme de Baum Welch, on expose d'abord les deux procédures suivantes :

##### Procédure CalculerGamma

###### Var

i, t : entier;

**Début** // la séquence d'observation O, donnée.

Forward;  
Backward; // le résultat = P.

pour t ← 1 à T  
  pour i ← 1 à N  
    gamma[t,i] ← (alpha[t,i] \* beta[t,i]) / P;

**Fin;**

##### Procédure Calculer Psi

###### Var

i, j, t : entier;

**Début** // la séquence d'observation O, donnée.

Forward;  
Backward; // le résultat = P.

pour t ← 1 à T-1  
  pour i ← 1 à N  
    pour j ← 1 à N  
      xi[t,i,j] ← (alpha[t,i] \* beta[t+1,j] \* A[i,j] \* B[j,O[t+1]]) / P;

**Fin;**

Enfin voilà la version complète de l'algorithme de Baum Welch

**Procédure BaumWelch**

**Var**

it, i, j, k, t: entier;  
 NbrEspTransSiVersSj, NbrEspTransDepuisSi, NbrEspSiAvecOk: réel;  
 Probat: réel;  
 gamma : Tableau [T] [N] élément de type réel ;  
 Psi : Tableau [T] [N] [N] élément de type réel ;

**Début**

```

it ← 1.
Tanque(it < Nbre Iteration)
  faire
    Forward;
    Backward;
    CalculerGamma ;
    CalculerPsi ;

    pour i ← 1 à N
      a0[i] ← gamma[1,i];
    pour i ← 1 à N
      faire
        NbrEspTransDepuisSi ← 0;
        pour t ← 1 à T-1
          NbrEspTransDepuisSi ← NbrEspTransDepuisSi + gamma[t,i];
        pour j ← 1 à N
          faire
            NbrEspTransSiVersSj ← 0;
            pour t ← 1 à T-1
              NbrEspTransSiVersSj ← NbrEspTransSiVersSj + xi[t,i,j];
            A[i,j] ← NbrEspTransSiVersSj / NbrEspTransDepuisSi;
          fait:
        pour k ← 1 à M
          faire
            NbrEspSiAvecOk ← 0;
            pour t ← 1 à T
              si O[t]=k alors
                NbrEspSiAvecOk ← NbrEspSiAvecOk + gamma[t,i];
            B[i,k] ← NbrEspSiAvecOk / NbrEspTransDepuisSi;
          fait:
        fait:
      fait:
    it ← it + 1;
  fait:

```

**Fin;**

## 3.10.4.2. Entraînement de Viterbi

Procédure Entraînement De ViterbiVar

i, j, k, t, Ni, Nij, Nio : entier;  
 Stab : booléen ← faux; // la stabilité du paramétrage.  
 V : Tableau [T] [N] élément de type réel ;

Début

Viterbi; // le résultat = V (chemin de Viterbi)

pour i ← 1 à N  
   si V[1]=i alors pi[i] ← 1  
   sinon pi[i] ← 0;

Tantque non Stab

faire

  Stab ← vrai;

pour i ← 1 à N

faire

    Ni ← 0;

pour t ← 1 à T si (V[t]=i) alors Ni ← Ni+1;

si Ni ≠ 0

alors

pour j ← 1 à N

faire

        Nij ← 0;

Pour t ← 1 à T-1

si (V[t]=i) et (V[t+1]=j) alors Nij ← Nij+1;

        Stab = (Stab) et (A[i,j]=Nij/Ni);

        A[i,j] ← Nij/Ni;

fait;

pour k ← 1 à N

faire

        Nio ← 0;

pour t ← 1 à T

si (V[t]=i) et (O[t]=k) alors Nio ← Nio+1;

        Stab = (Stab) et (B[i,k]=Nio/Ni);

        B[i,k] ← Nio/Ni;

fait;

fsi;

fait;

fait;

Fin;

### 3.11. Procédure de reconnaissance

La procédure de reconnaissance consiste à calculer par l'algorithme Forward-Backward, la probabilité  $P(O/\lambda)$  de production de la séquence d'observation  $O$  sachant le modèle  $\lambda = (A, B, \pi)$  et pour reconnaître le phonème il faut décider au profit du modèle donnant le maximum de probabilité  $P(O/\lambda)$ .

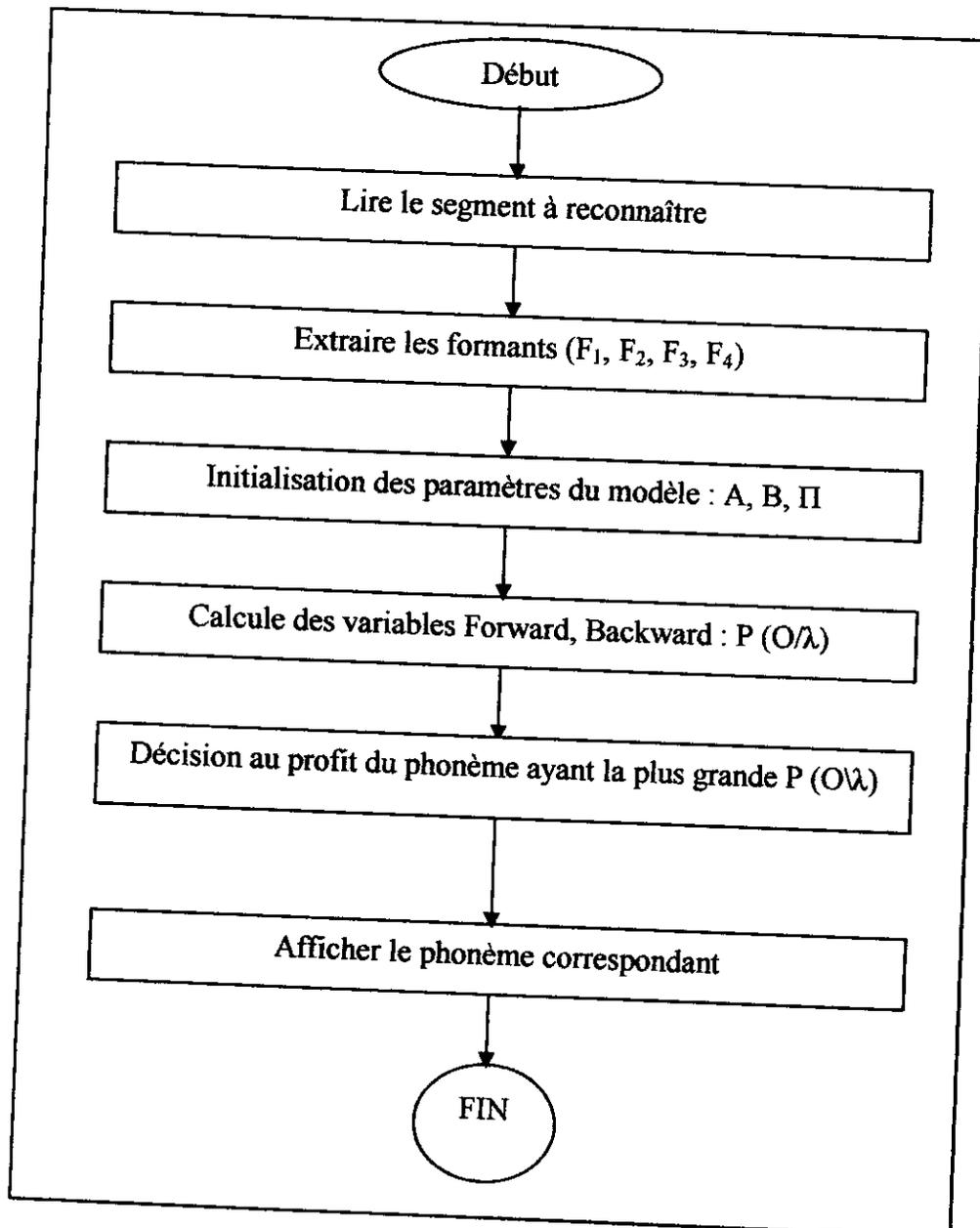


Figure 3.5 : Organigramme de la procédure de reconnaissance phonémique.

### 3.12. Conclusion

Nous avons présenté dans ce chapitre un système de reconnaissance phonémique concernant les phonèmes de l'AS en utilisant le modèle de HMM et nous avons vu l'entraînement des paramètres de différents algorithmes.

# Chapitre 4 :

## Généralités sur L'AS et la RAP

## 4.1. Introduction

Dans ce chapitre nous réalisons un système de reconnaissance des phonèmes basé sur les HMMs, en utilisant les différents algorithmes présentés dans le chapitre précédent.

La reconnaissance est appliquée aux phonèmes oraux de l'Arabe Standard, ceci nécessite un corpus constitué de phrases contenant les vingt neuf phonèmes à reconnaître, après l'enregistrement de ce dernier nous avons fait une segmentation afin d'extraire les formants qui sont utilisés comme valeurs d'entrée.

## 4.2. Procédure d'apprentissage

Un phonème peut être produit en suivant tous les parcours allant de l'état initial et aboutissant en l'état final. La vraisemblance du phonème est calculée par l'algorithme Forward-Backward et donc donnée par :

$$P(O / \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (4.1)$$

$$P(O / \lambda) = \sum_{i=1}^N \pi_i \cdot \beta_1(i) \cdot b_i(o_1) \quad (4.2)$$

Et les formules de réestimation des paramètres de notre modèle sont calculées à l'aide de l'algorithme de Baum Welch décrit dans le chapitre précédent.

Rappelons que  $a_{ij}$  représente la probabilité de transition de l'état  $i$  vers l'état  $j$  et que  $b_j(k)$  représente la probabilité d'émission par l'état  $i$  de l'observation quantifiée du symbole  $k$ . vue le modèle adopté, la matrice des probabilités d'émission est de dimension  $N \times M$  nous prenons pour notre cas  $4 \times 4$  ('N=4' pour le nombre d'états émetteurs et 'M=4' pour le nombre de formants), la matrice des probabilités de transition est de dimension  $4 \times 4$  ('4' étant le nombre d'états).

Dans ce qui suit, on représentera les paramètres de chaque modèle HMM sous forme de tableau récapitulatif donnant :

- la matrice A ;
- la matrice B ;
- la vraisemblance P (O/λ) ;
- la matrice A réestimée ;
- la matrice B réestimée .

#### 4.2.1. Problème d'apprentissage

Rappelons, que l'algorithme d'apprentissage reçoit en entrée :

- les paramètres du modèle.
- une séquence de formation.

Et donne comme résultat les paramètres du modèle après l'entraînement.

Pour résoudre ce problème, on a le choix entre deux solutions : l'entraînement de *Viterbi* et l'entraînement de *BaumWelch* et dans notre application nous avons choisi l'entraînement de Baum Welch.

### 4.3. Trace d'exécution des algorithmes de l'apprentissage

#### 4.3.1. La trace d'exécution de l'algorithmes Forward

Rappelons que le calcul des variables  $\alpha_i$  se fait en deux étapes, l'initialisation et l'induction :

$$\alpha_1(i) = \pi_i b_i(o_1) \quad (4.3)$$

$$\alpha_{t+1}(j) = \left( \sum_{i=1}^N \alpha_t(i) \cdot a_{ij} \right) \cdot b_j(o_{t+1}) \quad (4.4)$$

Or, connaissant  $\alpha_T(i)$ , c'est-à-dire la probabilité d'avoir généré la séquence O et d'être arriver sur l'état i, pour tous  $i \in [1..N]$ . Le calcul de P (O/λ) est immédiat :

$$p(O / \lambda) = \sum_{i=1}^N \alpha_T(i) \quad (4.5)$$

#### 4.3.2. La trace d'exécution de l'algorithmes Backward

Rappelons que le calcul des variables  $\beta_i$  se fait en deux étapes, l'initialisation et l'induction :

$$\beta_T(i) = 1 \quad (4.6)$$

$$\beta_t(i) = \sum_{j=1}^N a_{ij} \cdot b_j(o_{t+1}) \cdot \beta_{t+1}(j) \quad (4.7)$$

Et, connaissant  $\beta_1(i)$ , c'est-à-dire la probabilité de générer la séquence  $O$  en partant de l'état  $i$  et d'arriver sur le dernier état « correspond à  $t=T$  » pour chaque état  $i$ . Le calcul  $P(O/\lambda)$  peut alors être réalisé par la formule :

$$p(O/\lambda) = \sum_{i=1}^N \pi_i \cdot \beta_1(i) \cdot b_i(o_1) \quad (4.8)$$

#### 4.3.3. La trace d'exécution de entraînement de BaumWelch

Nous avons opté pour un critère d'arrêt de l'algorithme d'entraînement de BaumWelch, qui est un nombre d'itération ( $It$ ) fixé empiriquement à 40 pour avoir des résultats exploitables.

On donne deux exemples ayant citer les traces d'exécution des algorithmes et les résultats pour deux phonèmes (  $\text{ت}$  et  $\text{د}$  ) dans les différentes positions (initiale, médiane et finale)

#### 4.4. Modèle du Phonème (MP)

On donne deux exemples ayant citer les traces d'exécution des algorithmes et les résultats pour deux phonèmes (  $\text{ت}$  et  $\text{د}$  ) dans les différentes positions (initiale, médiane et finale)

##### 4.4.1. MP : $\text{ت}$ en Pi

La matrice A

0.360347779079196	0.0603207496170826	0.359672042526354	0.219659428777367
0	0.153870761356366	0.343463425037321	0.502665813606313
0	0	0.707067543584925	0.292932456415075
0	0	0	1

**La matrice B**

0.072487195355037	0.202202699722688	0.323435512210452	0.401874653225494
0.0728819030503679	0.204494389705685	0.329241079270789	0.39338265839959
0.0734365992513416	0.207001434513591	0.334040062423232	0.385521911130404
0.071523533608642	0.205413890463258	0.328675842422405	0.394386757737102

**4.4.1.1. La trace d'exécution de l'algorithme Forward**

0.072487195355037	0	0	0
0.00528165580963686	0.000894148030207008	0.00539686224543286	0.00327070183391375
0.000615573121083951	0.000150192101088417	0.00201182883240607	0.00212365385954447
8.91439991589247E-5	2.36981600947159E-5	0.000653648084743316	0.00115306726059998

$P(O/\lambda) = 0.001919557456858460$ .

**4.4.1.2. La trace d'exécution de l'algorithme Backward**

0.02648133225739	0.0266765095293522	0.0267746187746525	0.0266268588602543
0.128774225711823	0.12950511276722	0.129640817642212	0.129625409841537
0.393835991621017	0.391187518835068	0.388118714094162	0.394386768341064
1	1	1	1

$P(O/\lambda) = 0.001919557456858460$ .

**4.4.1.3. Problème d'apprentissage**

La trace d'exécution de l'entraînement de BaumWelch le résultat après le reestimation est :

$P(O/\lambda) = 0.000365539222099$ .

**4.4.2. MP :  $\omega$  en PM**

**La matrice A**

0.195181576616475	0.215766164747564	0.325987599645704	0.263064658990257
0	0.401713573016648	0.31017840681091	0.288108020172442
0	0	0.549693449499839	0.450306550500161
0	0	0	1

**La matrice B**

0.0745348862906289	0.174958413097531	0.343085509203003	0.40742116534149
0.044692210264648	0.263324909652343	0.311785221622032	0.380197654613839
0.0430422474936443	0.256671103123035	0.310975232672471	0.389311489260333
0.0552280194093346	0.235696082900096	0.312079523520815	0.396994361968618

**4.4.2.1. La trace d'exécution de l'algorithme Forward**

0.0745348862906289	0	0	0
0.00254526640888594	0.00423481925556859	0.00623645295365279	0.00462144885114237
0.000170441144975099	0.000701631094489127	0.00173257117835742	0.00290839999688245
1.35536682537673E-5	0.000121142441830275	0.000477130639583283	0.00156239939288798

$P(O/\lambda) = 0.002174226101487870.$

**4.4.2.2. La trace d'exécution de l'algorithme Backward**

0.0291705839335918	0.0311224199831486	0.0304940138012171	0.0292015317827463
0.124811731278896	0.122159950435162	0.122930899262428	0.123893804848194
0.392900794744492	0.387863844633102	0.392771124839783	0.396994352340698
1	1	1	1

$P(O/\lambda) = 0.00217422610148787.$

**4.4.2.3. Problème d'apprentissage**

La trace d'exécution de l'entraînement de BaumWelch le résultat après le reestimation est ;

$P(O/\lambda) = 0.000814698994839.$

**4.4.3. MP :  $\hat{c}$  en PF**

La matrice A

0.323804662242641	0.452385003686688	0.169814531223413	0.0539958028472577
0	0.27885979620792	0.330194763317426	0.390945440474655
0	0	0.443004727718438	0.556995272281562
0	0	0	1

**La matrice B**

0.0770575704647931	0.181205541217614	0.337521391030597	0.404215556274906
0.0794920237188776	0.177731281920997	0.339889012162417	0.402887735747465
0.0808004718659748	0.173869315895447	0.331915301829044	0.413414940691448
0.0866802559494163	0.179544528605673	0.33584029510627	0.397934905273881

**4.4.3.1. La trace d'exécution de l'algorithme Forward**

0.0770575704647931	0	0	0
0.00452136828690808	0.00619565726644385	0.00227516609956716	0.000747046250167259
0.000494144861545428	0.0012824416665132	0.0012684066783695	0.00157193471697693
6.46770800223271E-5	0.000234144310083311	0.000442055772049458	0.00111679544203195

$P(O/\lambda) = 0.001857672585174440.$

**4.4.3.2. La trace d'exécution de l'algorithme Backward**

0.0241075940430164	0.0237594060599804	0.0236831698566675	0.0239947941154242
0.136462077498436	0.134943082928658	0.133959010243416	0.133642584085464
0.404837936162949	0.404427468776703	0.404792636632919	0.397934913635254
1	1	1	1

$P(O/\lambda) = 0.00185767259201139.$

**4.4.3.3. Problème d'apprentissage**

La trace d'exécution de l'entraînement de BaumWelch Le résultat après le reestimation est :

$P(O/\lambda) = 0.001213699249283.$

**4.4.4. MP :  $\lambda$  en PI**

**La matrice A**

0.0205157356208981	0.120200506433776	0.384011162213839	0.475272595731487
0	0.683832335329341	0.0353293413173653	0.280838323353293
0	0	0.930640913081651	0.0693590869183494
0	0	0	1

**La matrice B**

0.105116813662886	0.195115102828416	0.321270749600058	0.378497295120781
0.102399468120104	0.196498209848644	0.315437754540144	0.385664614523646
0.0983499244871318	0.19799958798259	0.313034429649271	0.390616042562202
0.0951793686391224	0.200089689518614	0.311630331270718	0.393100655555655

**4.4.4.1. La trace d'exécution de l'algorithme Forward**

0.105116813662886	0	0	0
0.00042077523275341	0.00248277339883622	0.00799245726550721	0.00999630898821281
2.773374060907E-6	0.000551504470623016	0.00240642239655915	0.00356751267678766
2.15356668091231E-8	0.000145576803168797	0.000882817424625541	0.00152940591582455

$P(O/\lambda) = 0.00255782157182693.$

**4.4.4.2. La trace d'exécution de l'algorithme Backward**

0.0243331361562014	0.024187158793211	0.0242436490952969	0.0245114043354988
0.12248345464468	0.12240369617939	0.122342139482498	0.122502088546753
0.390953123569489	0.387927860021591	0.390788376331329	0.393100649118423
1	1	1	1

$P(O/\lambda) = 0.00255782157182693$

**4.4.4.3. Problème d'apprentissage**

La trace d'exécution de l'entraînement de BaumWelch le résultat après le reestimation est :

$P(O/\lambda) = 0.000550483180796.$

**4.4.5. MP : → en PM**

**La matrice A**

0.0139900344959755	0.161172863165964	0.278523061198416	0.546314041139645
0	0.323558235959291	0.0846588767433095	0.591782887297399
0	0	0.266773934030571	0.733226065969429
0	0	0	1

**La matrice B**

0.158650693044077	0.214852340716282	0.258198581496561	0.368298444904936
0.156808338140616	0.215289095008398	0.256887798339996	0.371014788094685
0.157675988034706	0.210501294714933	0.253082237806524	0.378740489025799
0.151918879958145	0.21024657713456	0.258874312218264	0.378960260129231

**4.4.5.1. La trace d'exécution de l'algorithme Forward**

0.158650693044077	0	0	0
0.000476870929713485	0.00550498229811731	0.00930160525350394	0.0182227228667169
1.72255653994915E-6	0.000477308065432226	0.000779566938177564	0.00739375528437405
8.8754862647488E-9	5.74013989800262E-5	9.42519474188061E-5	0.00312595143404352

$P(O/\lambda) = 0.003277613548561930$

**4.4.5.2. La trace d'exécution de l'algorithme Backward**

0.0206593088805676	0.0207325145602226	0.0205993968993425	0.0206258352845907
0.0972500815987587	0.0974571853876114	0.0975135564804077	0.0981030762195587
0.377469301223755	0.376370817422867	0.37890163064003	0.378960251808167
1	1	1	1

$P(O/\lambda) = 0.003277613548561930$

**4.4.5.3. Problème d'apprentissage :**

La trace d'exécution de l'entraînement de BaumWelch le résultat après le reestimation est :

$P(O/\lambda) = 0.000264841876809$

**4.4.6. MP : en PF**

La matrice A

0.367427257451136	0.396391615813213	0.0562041213407624	0.179977005394888
0	0.282111609440482	0.027645026101022	0.690243364458496
0	0	0.469250210614996	0.530749789385004
0	0	0	1

**La matrice B**

0.105998616520392	0.240844701583199	0.258421196324183	0.39473548535752
0.119922431343302	0.218898601849918	0.270111178543131	0.391067832224703
0.124699786389787	0.200823890101725	0.279180151015316	0.395296183632293
0.122177097966121	0.200213551678231	0.280343067926518	0.397266231349686

**4.4.6.1. La trace d'exécution de l'algorithme Forward**

0.105998616520392	0	0	0
0.00938012583834741	0.0091974544276421	0.00119642019494995	0.00381953670779882
0.000890652248113876	0.00170518807576043	0.000374907123811293	0.00350182842878702
0.000129177153188843	0.000326189849445957	0.00010796469408805	0.00200146778861713

$P(O/\lambda) = 0.002564799506217240$

**4.4.6.2. La trace d'exécution de l'algorithme Backward**

0.0241965390741825	0.0228044912219048	0.0222980063408613	0.0222979504615068
0.10599410533905	0.110066808760166	0.11103294044733	0.111370839178562
0.393768638372421	0.395463138818741	0.396341770887375	0.397266238927841
1	1	1	1

$P(O/\lambda) = 0.002564799506217240.$

**4.4.6.3. Problème d'apprentissage**

La trace d'exécution de l'entraînement de BaumWelch le résultat après le reestimation est :

$P(O/\lambda) = 0.000602564143361.$

**4.5. Procédure de reconnaissance**

En phase de reconnaissance, les performances de notre système sont évaluées à l'aide d'un ensemble d'exemple de phonèmes.

La procédure de reconnaissance consiste à calculer par l'algorithme Forward-Backward, la probabilité  $P(O/\lambda)$  de production de la séquence

d'observation  $O$  sachant le modèle  $\lambda = (A, B, \pi)$  et pour reconnaître le phonème il faut décider au profit du modèle entraîné donnant le maximum de vraisemblance de la probabilité  $P(O/\lambda)$ .

#### 4.5.1. Effet de la taille du corpus d'apprentissage

Il est assez intuitif de penser que le fait d'augmenter la taille du corpus d'apprentissage permet d'améliorer le taux de reconnaissance. En effet, ceci permet de capturer les statistiques de chaque phonème du vocabulaire, rendant plus meilleures les performances de reconnaissance de nos modèles HMM. C'est pour cela que nous avons spéculé sur la taille du corpus d'apprentissage lors de nos expériences pour voir son effet sur le Taux de Reconnaissance (TR).

#### 4.6. Choix du langage de programmation

Dans toute branche de l'ingénierie, les outils communément disponibles jouent un rôle considérable. Parmi ces outils, le langage de programmation qui occupe une place sans doute importante dans le domaine d'informatique. D'une importance capitale pendant les phases de réalisation et la maintenance du logiciel, son choix devient par ce fait très délicat.

Pour notre projet, le langage de programmation choisi est le *Visuel Basic* (sous *Windows*), ce choix est motivé par les critères suivants :

##### ***Pour quoi le Visuel Basic ?***

- le langage *Visuel Basic* est souple, modulaire et puissant ;
- programmation à base Orienté Objet ;
- permet la compilation séparée ;
- permet l'exécution rapide surtout dans le cas des calculs compliqués et itératives le temps de réponse est réduit dans le cas de la recherche dans les grandes base des données.

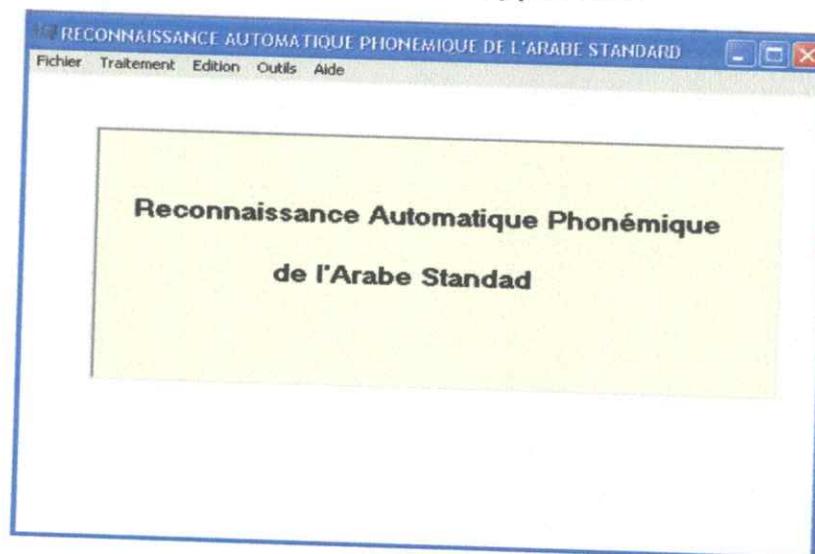
##### ***Pour quoi sous Windows ?***

- offre une interface graphique qui permet à l'utilisateur un usage facile de n'importe quel programme *Windows*.

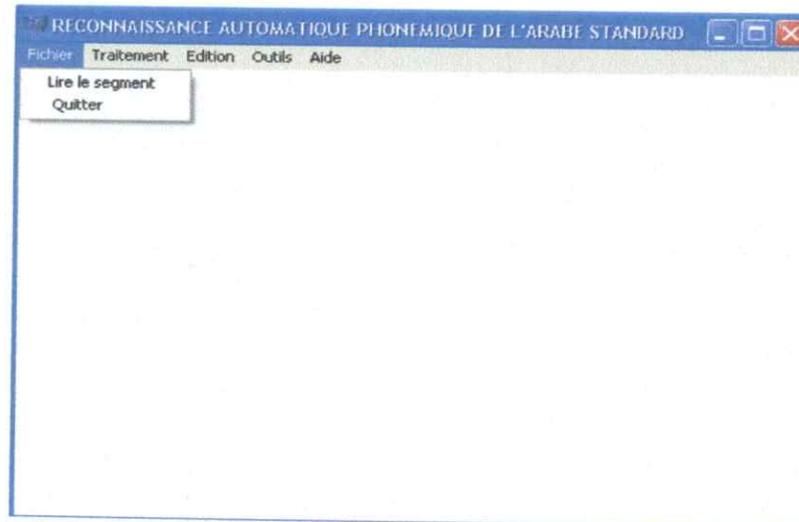
- Windows propose un ensemble d'objets d'interfaces utilisateur composé de fenêtres, de menus et d'icônes, ce qui rend les applications Windows plus conviviales (se dit d'un matériel facilement utilisable par un public non spécialisé), et faciles à comprendre et à utiliser.
- L'utilisation de logiciel de segmentation PRAAT exige la plateforme Windows dont le choix est plus convenable pour l'extraction des formants et l'exactitude des résultats de la segmentation.

#### 4.7. Description du logiciel

C'est la fenêtre principale de notre application, Le menu principal permet d'accéder aux différentes fonctions de notre application



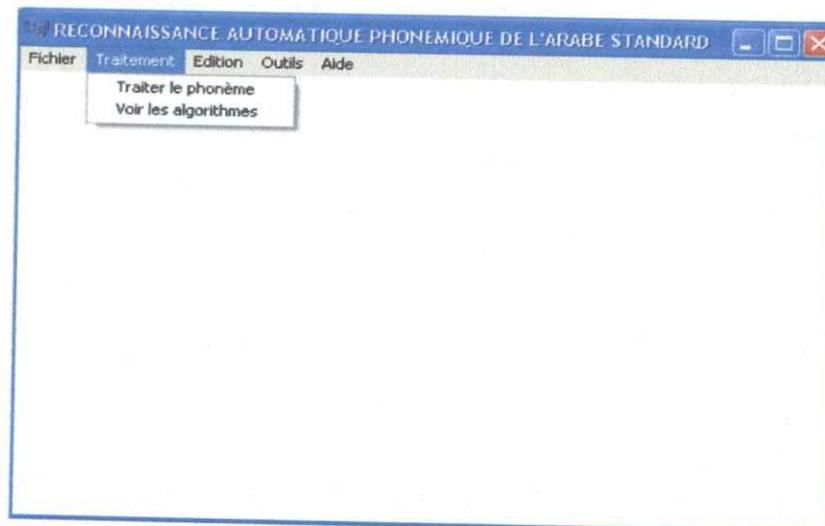
Dans ce qui suit nous allons détaillé chaque sous menu



#### 4.7.1. Menu Fichier

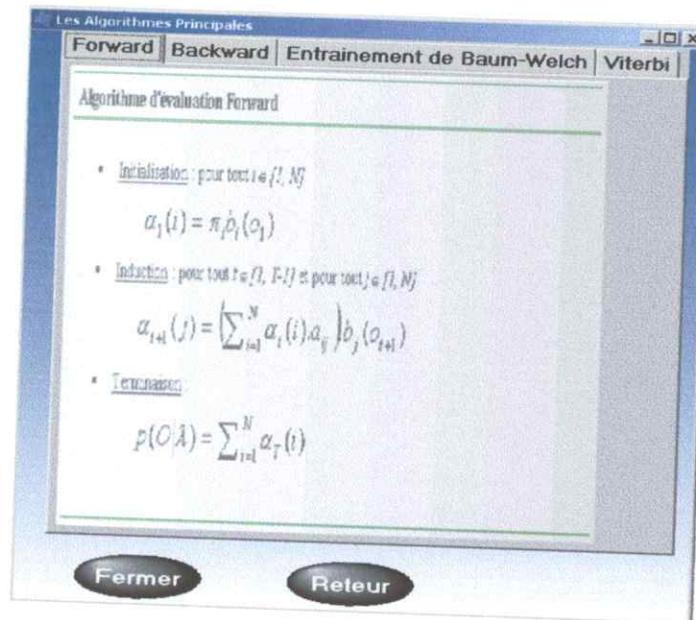
Le menu Fichier comporte les actions suivantes :

- *Lire les Segments*: pour lire les fichiers générés par **PRAAT** qui contiennent les valeurs des formants.
- *Quitter* : Quitter l'application.

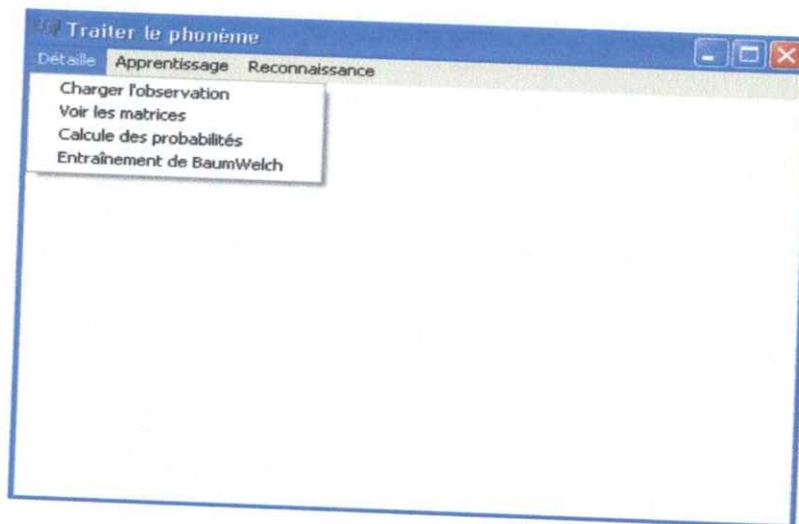


#### 4.7.2. Menu Traitement

- *Voir les Algorithmes*: Pour voir ou imprimer les algorithmes des HMMs.



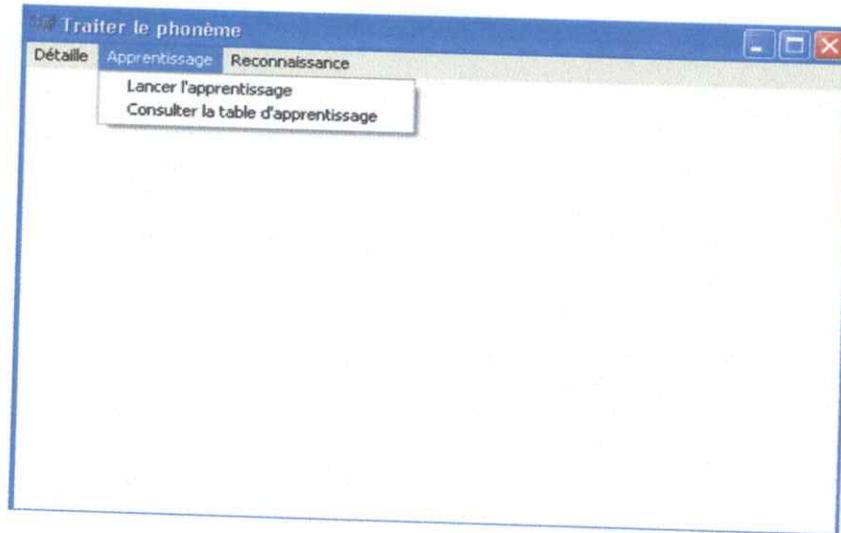
- *Traiter le phonème* : pour mieux voir les traitement sur le phonème choisi



Le menu traiter le phonème comporte les actions principales pour la manipulation et la visualisation des paramètres du HMM :

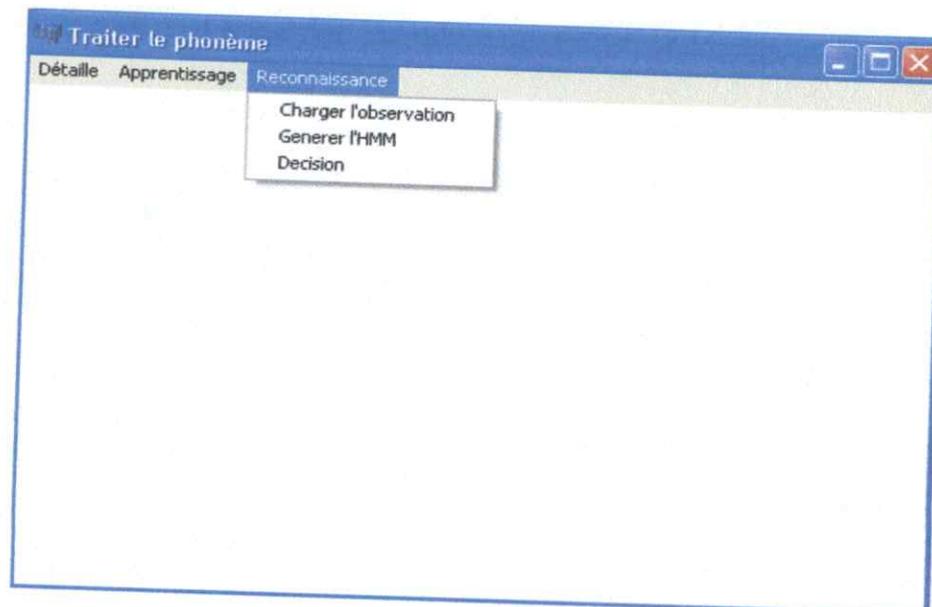
- *Charger l'observation* : Charger le fichier qui contient les formants du phonème ;
- *Voir les Matrices* : pour visualiser les différentes matrices du HMM.
- *Calculer les Probabilités* : permet de calculer les probabilités avec l'Algorithmes de Forward et de Backward ;

- *Apprentissage* : Lancer l'apprentissage avec l'algorithme de Baum Welch.
- *Quitter* : Quitter la fenêtre.



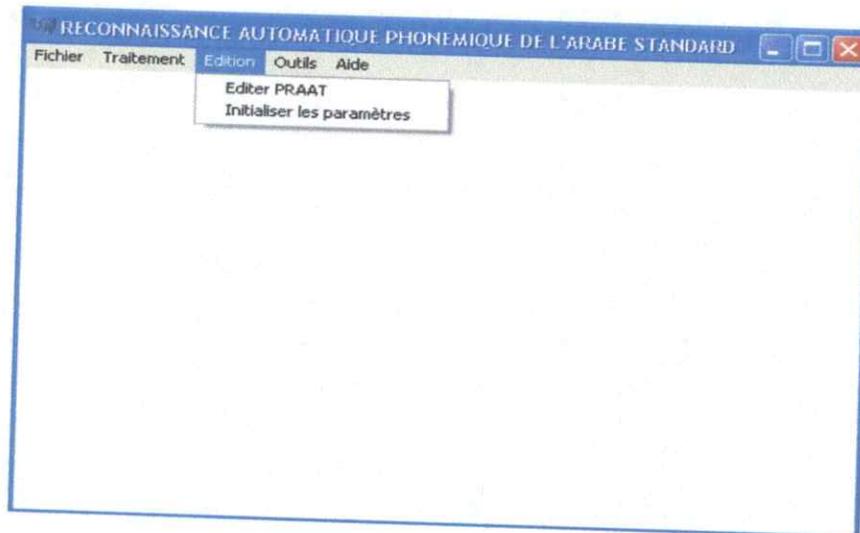
#### 4.7.2.3. Menu apprentissage

- *Lancer l'apprentissage par Baum Welch* : pour éditer les 3 autres commandes : charger l'observation, générer HMM correspondant et consulter les résultats de l'Apprentissage pour soit les sauvegardés ou bien les annulés ;



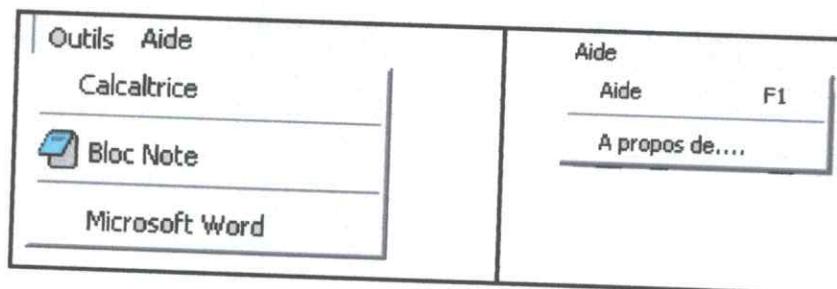
### 4.7.2.3. Menu Reconnaissance

- *Charger l'observation* : charger l'observation à reconnaître.
- *Générer le HMM correspondant* : générer le HMM correspondant à l'observation après on vient l'étape de décision.



### 4.7.3. Menu Edition

- *Editer PRAAT* : pour éditer **PRAAT** de nouveau.
- *Initialiser les Paramètres* : Initialise les paramètres de notre système.



### 4.7.4. Menu Outils

- *Calculatrice ; Bloc note et Microsoft word*

### 4.7.5. Menu Aide

- *Aide ; A propos.*

#### **4.8. Conclusion**

Nous avons présenté dans ce chapitre un système de reconnaissance des consonnes orales de l'Arabe Standard en utilisant les HMMs.

Les phonèmes à reconnaître sont pris dans des phrases enregistrées et situées dans des positions différentes.

D'après les résultats obtenus, nous avons constaté que la taille du corpus influe sur le taux de reconnaissance car le nombre d'observation augmente.

# Conclusion générale

## **Conclusions générales et perspectives**

Cette étude s'inscrit dans le cadre de Reconnaissance Automatique Phonémique de l'Arabe Standard

Nous avons utilisé un corpus constitué de 76 phrases et 15 mots contenant les vent huit phonèmes de l'AS dans les différentes positions (initiale, médiane et finale). Puis nous avons analysé ce corpus afin d'extraire les formants qui sont considérés parmi les paramètres pertinents du signal vocal.

Cette analyse est faite à l'aide du logiciel PRAAT.

La phase de segmentation est très difficile à cause de la variabilité du signal vocal (intra-locuteur, interlocuteur et contextuelle), cela nécessite de bonnes connaissances en phonétique et en phonologie.

Pour cela, nous avons utilisé les modèles de Markov Cachés et nous avons utilisé les algorithmes de Forward-Backward et Baum-Welch pour résoudre les trois problèmes des HMM (évaluation, décodage et apprentissage).

Enfin, il faut mentionner que le développement d'un système à base des HMM est une tâche très délicate et qui nécessite beaucoup d'expériences. En effet de nombreux problèmes se posent concernant l'implémentation des algorithmes de l'HMM, l'extraction des paramètres, etc.

Nous suggérons des perspectives à notre travail telles que :

- élaboration du corpus : corpus de taille en mode multilocuteurs. Le choix de ce type de corpus (parole continue) par le fait qu'il est préférable d'étudier les segments dans un continuum vocal pour pouvoir prendre en considération les effets de coarticulation existants entre les phonèmes ;
- prendre en considération des autres paramètres comme valeurs d'entrée (durée, intensité, pitch, ...), également les voyelles longues ;
- création d'une base de données contenant le maximum des données a fin de faciliter la manipulation aux utilisateurs de cette base;
- utilisation des autres logiciels d'analyse (Speech Analyzer, CLAN...).

# Références Bibliographiques



## Références Bibliographiques

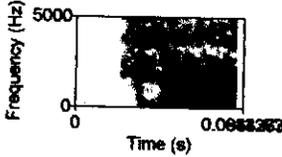
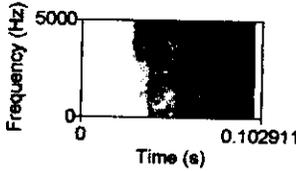
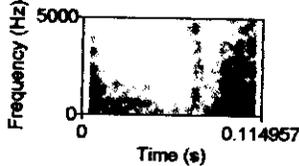
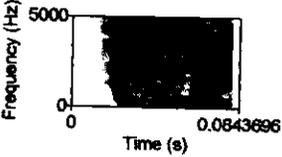
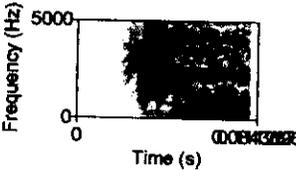
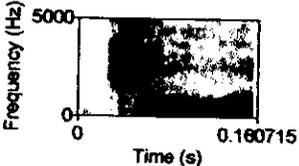
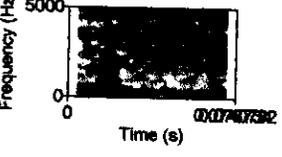
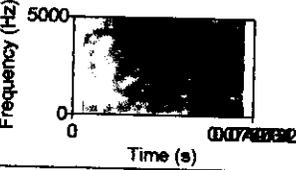
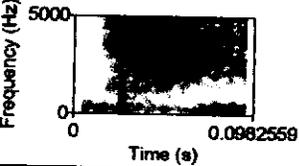
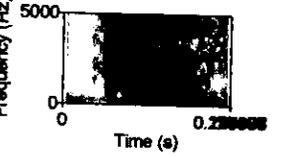
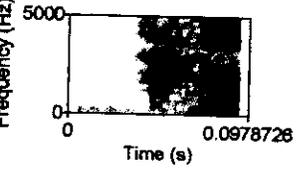
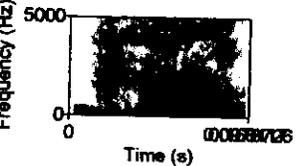
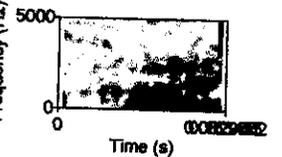
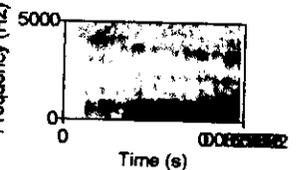
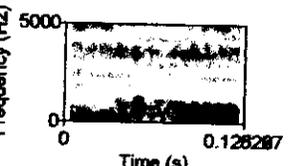
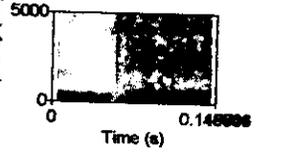
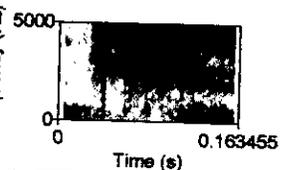
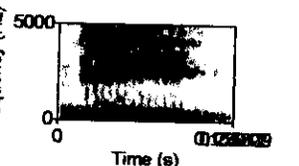
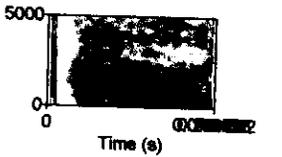
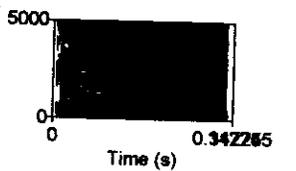
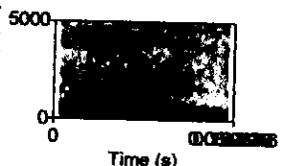
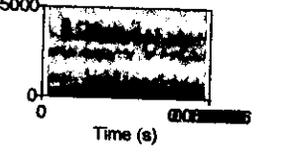
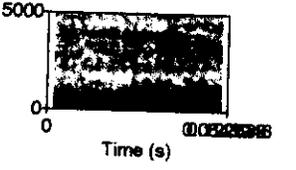
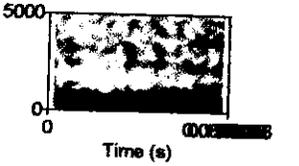
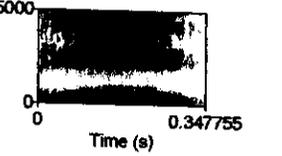
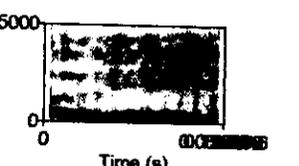
- [1] <http://www.unil.ch/ling/page13431.html>.
- [2] T. Dutoit, Introduction au Traitement Automatique de la Parole, thèse de doctorat, pp 18, Faculté polytechnique de Mons, 2000.
- [3] Calliope, La parole et son traitement automatique, Edition Masson, Paris 1989.
- [4] D. Rebai & A. Boudjettou, Reconnaissance Automatique des fricatives de l'Arabe Standard par les modèles de Markov cachés (HMM), PFE, Université de Blida, Département d'informatique, 2005.
- [5] R.Boite & M.Kunt, Traitement de la parole, Presses polytechniques Romand, 1987.
- [6] K. Kabache, Application des Réseaux de Neurones à la Reconnaissance Automatique des Phonèmes Spécifiques en Arabe Standard, Mémoire de Magister, CRSTDLA, Bouzaréah, Alger, Algérie 2006.
- [7] L. R. Rabiner, A tutorial on Hidden Markov Models and selected applications in Speech Recognition. Proceedings of the IEEE, 77(2): 257-285, 1989.
- [8] L. Bréhélin & O. Gascuel, Modèles de Markov cachés et apprentissage de Séquences, Editions Cépaduès, H. Prade, R. Jeansoulin et C. Garbay éditeurs, P 407-421, 2000.
- [9] P. Langlais, Introduction aux Modèles de Markov Cachés (HMM), thèse de doctorat, Université de Montréal, Octobre 2003.
- [10] H. Kechih & A. Hadj Hamdi, Reconnaissance Automatique des Occlusives de l'Arabe Standard par les réseaux de Neurones Artificiels, PFE, Université de Blida, Département d'informatique, 2005
- [11] G. Droua-Hamdani, Prédiction de la durée segmentale des phonèmes de l'Arabe Standard, Mémoire de Magister, CRSTDLA, Bouzaréah, Alger, Algérie, 18/02/2003.

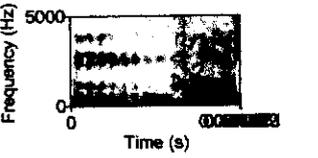
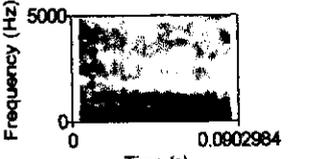
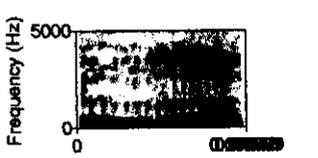
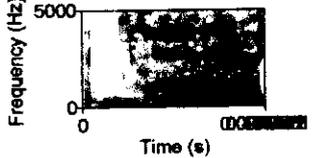
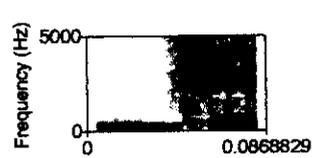
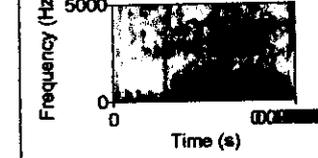
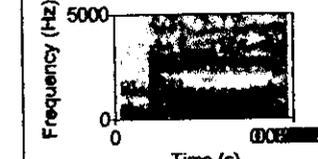
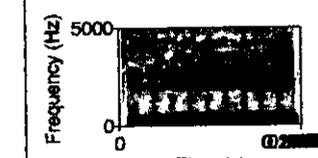
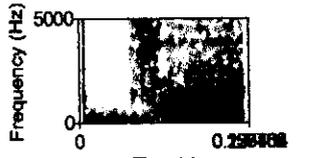
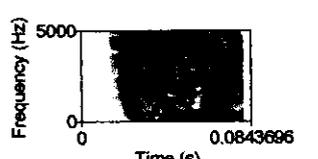
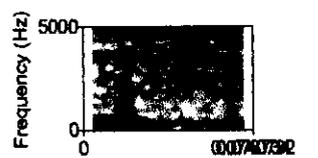
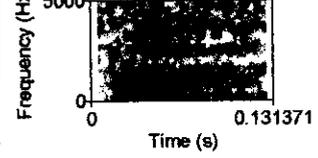
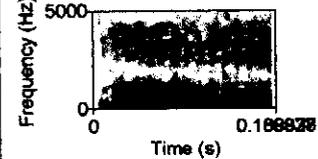
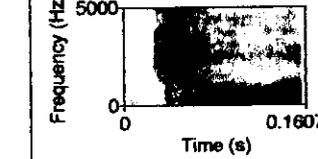
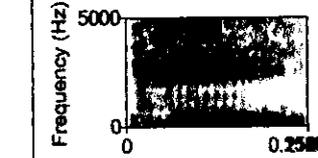
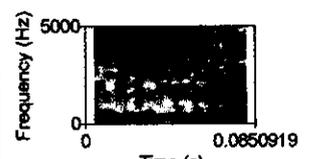
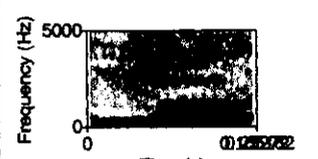
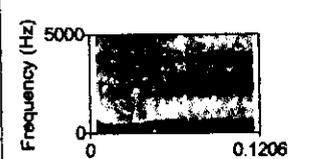
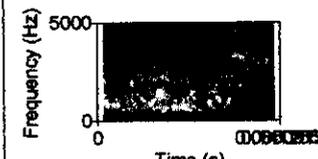
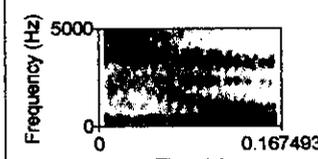
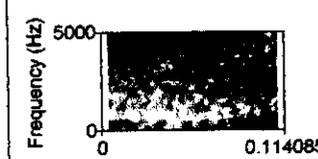
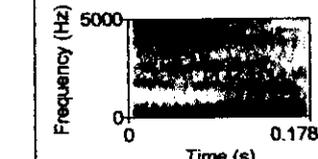
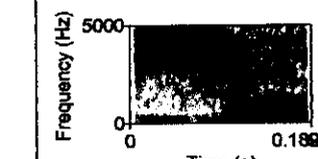
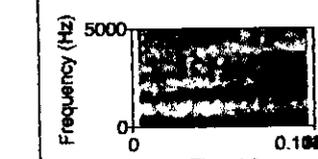
# Annexes

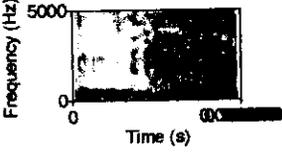
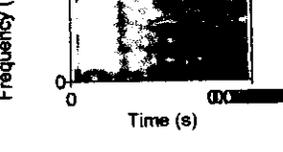
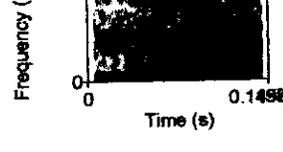
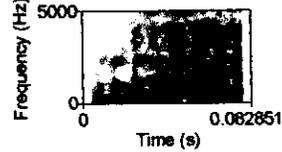
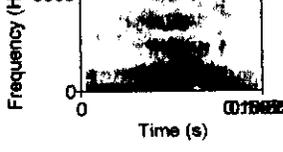
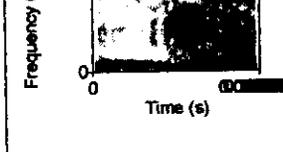
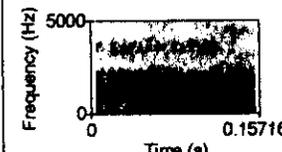
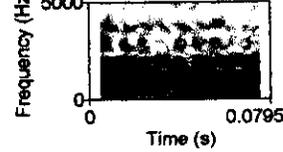
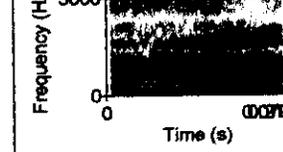
## **Annexe A :**

Nous présentons un tableau qui figure les Sonagrammes des 28 phonèmes de l'AS étudiés dans les différentes positions (initiale, médiane, finale) suivant l'ordre des phonèmes dans Tab 1.1 (TOP), nous nous sommes limités dans notre étude aux 3 voyelles brèves.

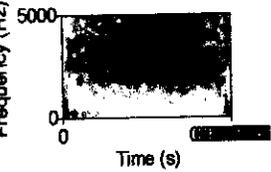
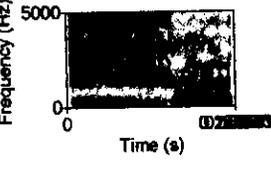
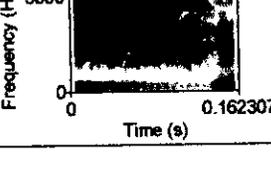
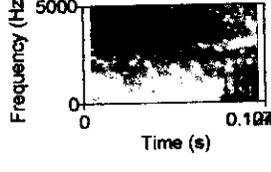
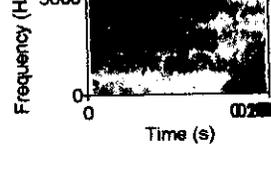
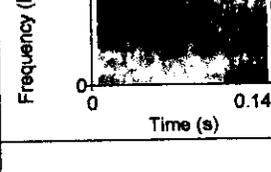
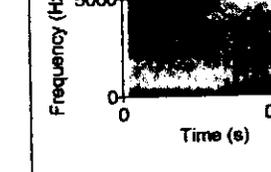
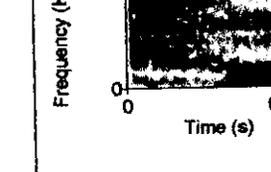
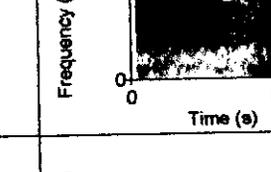
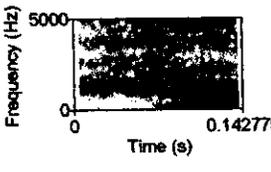
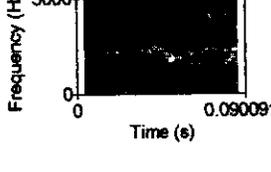
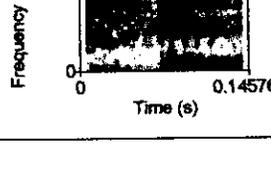
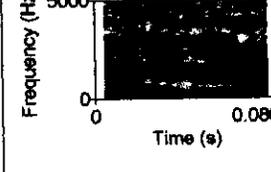
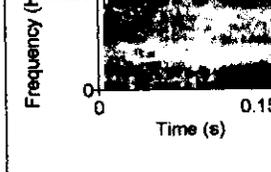
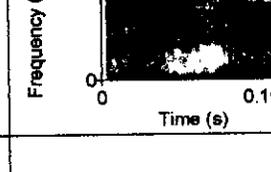
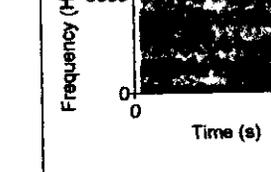
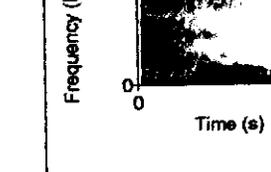
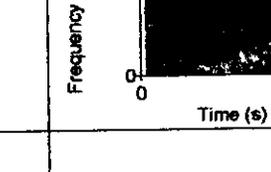
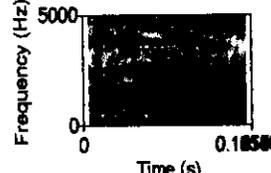
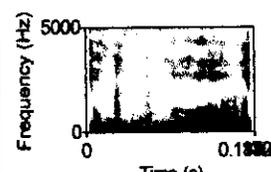
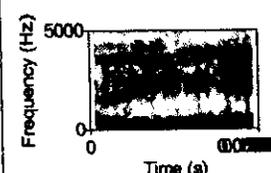
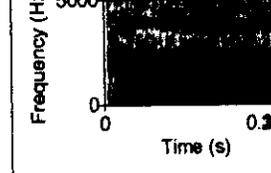
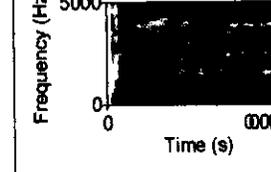
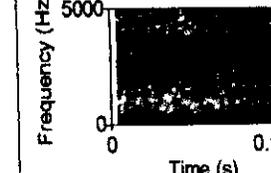
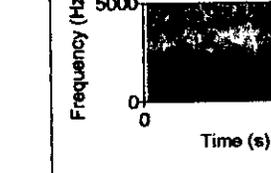
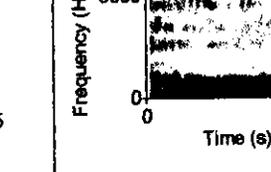
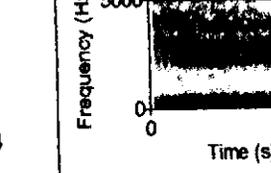
Phonème	Voyelles brèves	Position		
		Initiale	Médiane	Finale
[ç] [b]	[a]			
	[u]			
	[i]			
[ç] [d]	[a]			
	[u]			
	[i]			
[ç] [q]	[a]			
	[u]			
	[i]			

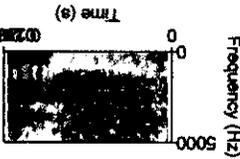
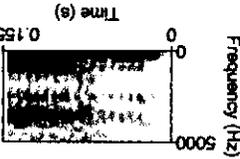
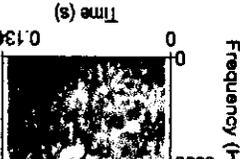
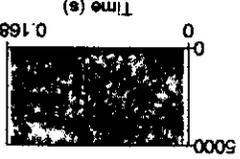
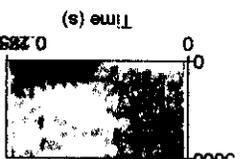
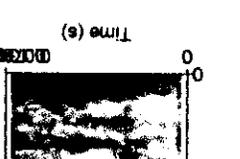
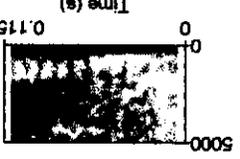
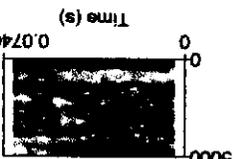
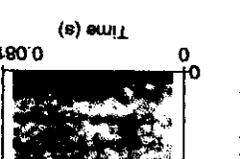
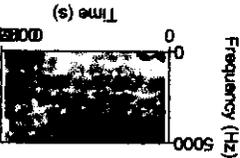
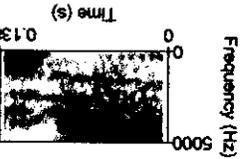
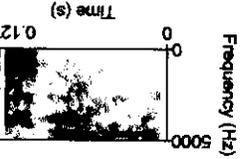
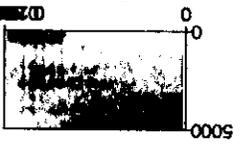
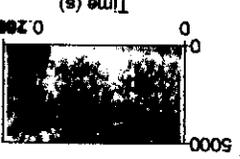
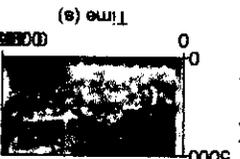
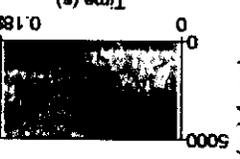
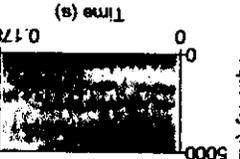
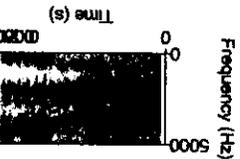
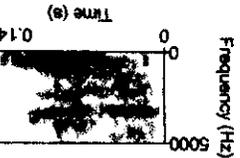
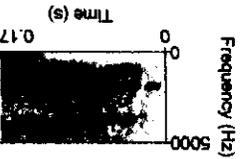
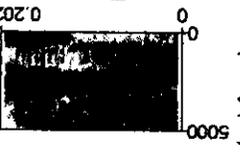
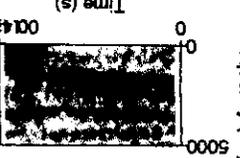
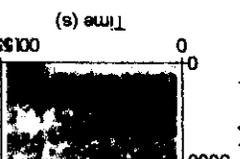
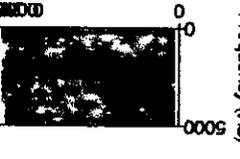
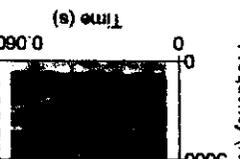
[t]	[a]	 <p>Frequency (Hz) 5000 0 Time (s) 0.0888263</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.102911</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.114957</p>
	[u]	 <p>Frequency (Hz) 5000 0 Time (s) 0.0843696</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0843696</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.160715</p>
	[i]	 <p>Frequency (Hz) 5000 0 Time (s) 0.0748392</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0748392</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0982559</p>
[d] [k]	[a]	 <p>Frequency (Hz) 5000 0 Time (s) 0.226666</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0978726</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0888726</p>
	[u]	 <p>Frequency (Hz) 5000 0 Time (s) 0.0826692</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0826692</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.126287</p>
	[i]	 <p>Frequency (Hz) 5000 0 Time (s) 0.146666</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.163455</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.128909</p>
[p] [j]	[a]	 <p>Frequency (Hz) 5000 0 Time (s) 0.0888263</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.342265</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0888263</p>
	[u]	 <p>Frequency (Hz) 5000 0 Time (s) 0.0843696</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0843696</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0843696</p>
	[i]	 <p>Frequency (Hz) 5000 0 Time (s) 0.347755</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0843946</p>	 <p>Frequency (Hz) 5000 0 Time (s) 0.0843946</p>

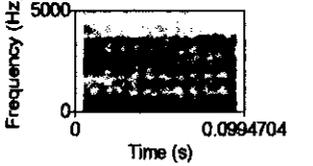
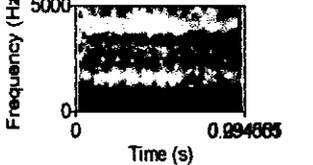
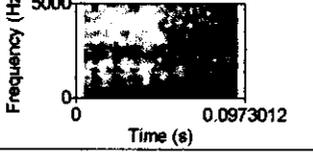
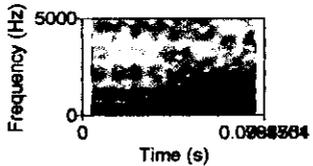
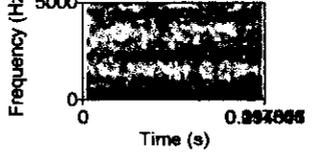
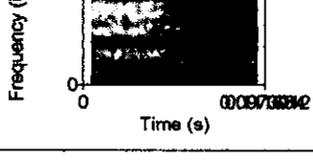
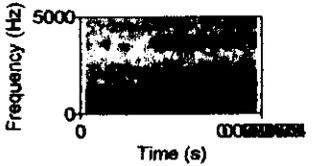
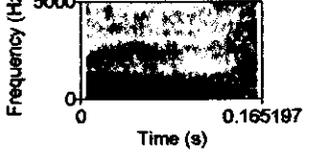
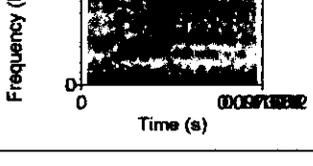
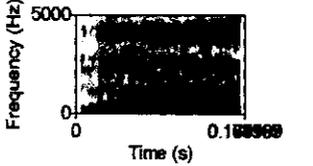
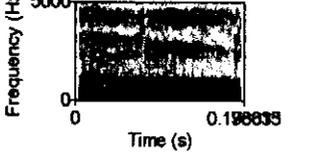
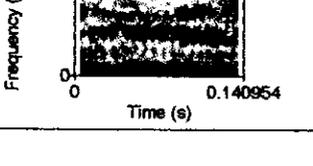
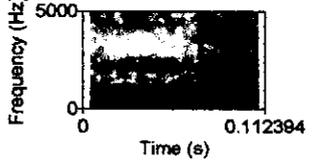
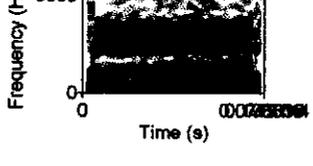
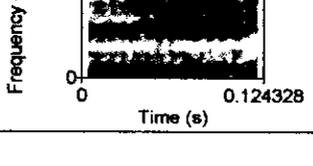
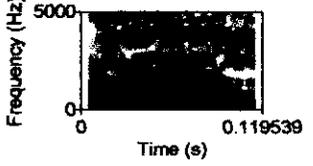
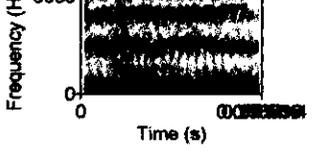
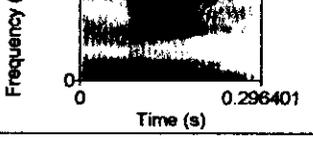
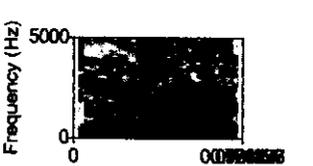
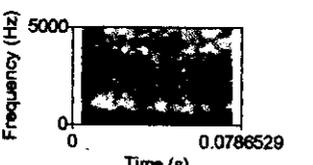
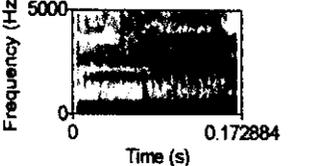
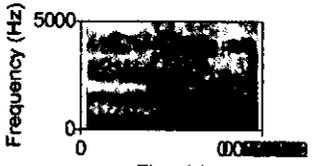
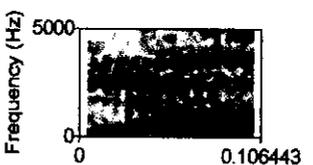
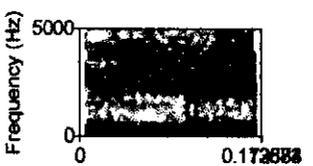
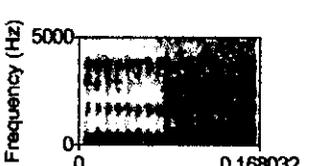
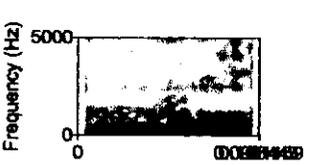
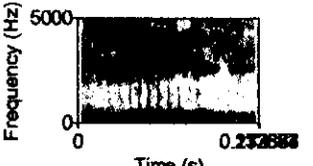
<p>[ض]</p> <p>[د]</p>	<p>[a]</p> <p>[u]</p> <p>[i]</p>	  	  	  
<p>[ط]</p> <p>[ت]</p>	<p>[a]</p> <p>[u]</p> <p>[i]</p>	  	  	  
<p>[ج]</p> <p>[ز]</p>	<p>[a]</p> <p>[u]</p> <p>[i]</p>	  	  	  

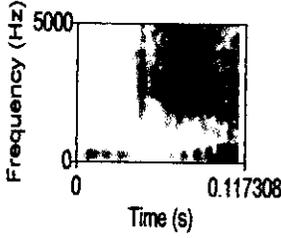
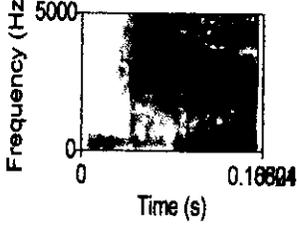
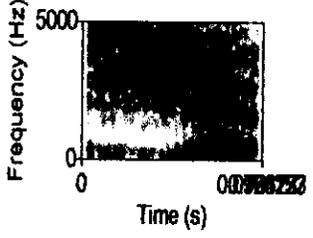
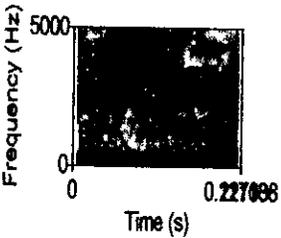
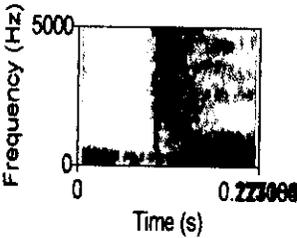
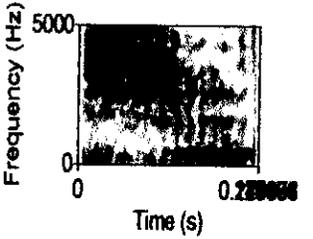
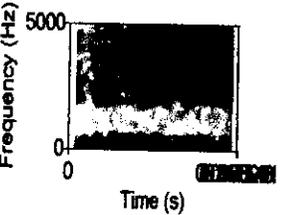
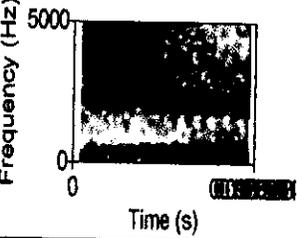
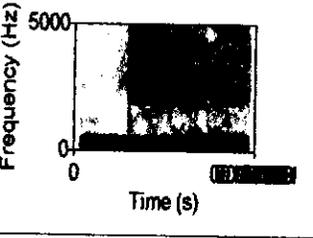
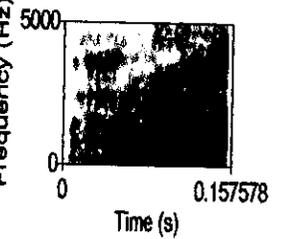
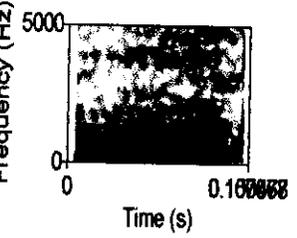
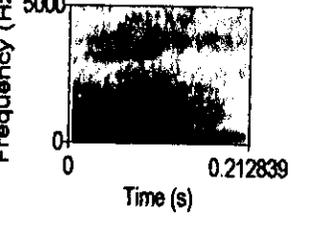
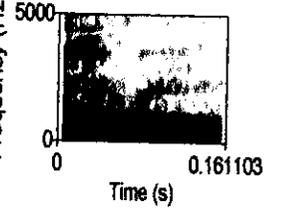
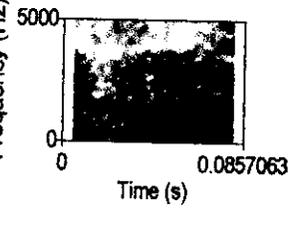
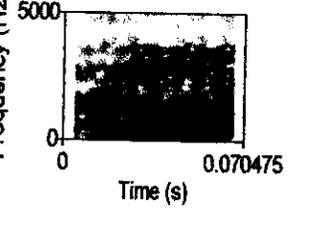
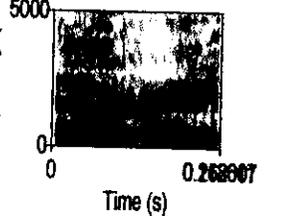
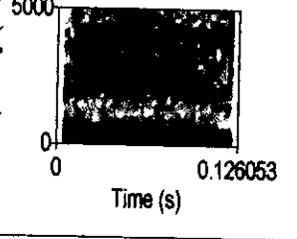
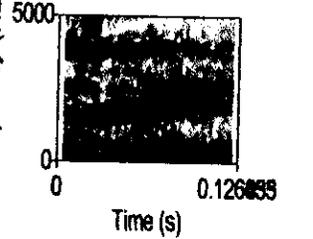
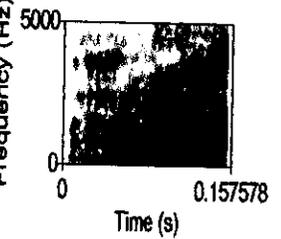
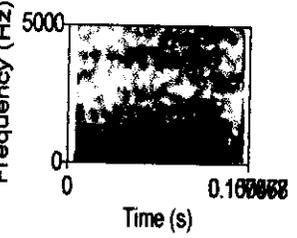
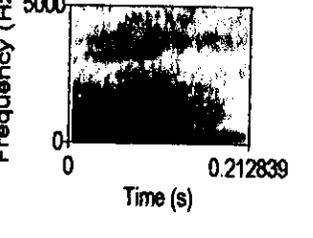
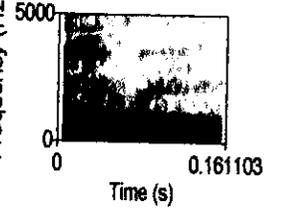
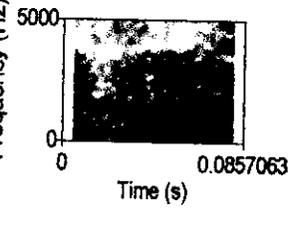
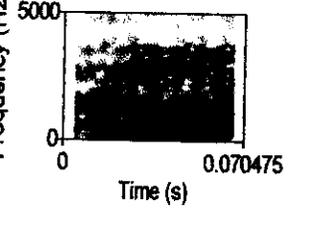
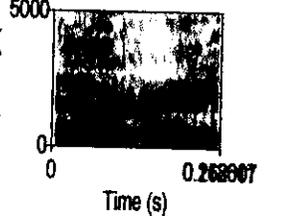
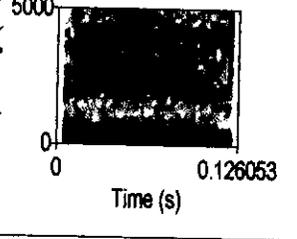
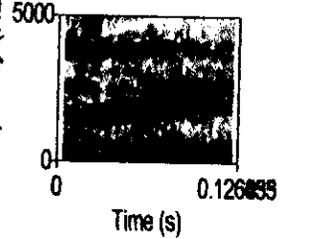
<p>[ɔ] [d]</p>	<p>[a] [u] [i]</p>			
<p>[ɛ] [ɔ]</p>	<p>[a] [u] [i]</p>			
<p>[ɛ] [ɛ]</p>	<p>[a] [u] [i]</p>			

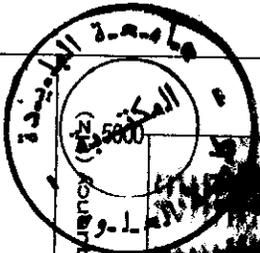
[س] [s]	[a]			
	[u]			
	[i]			
[ت] [t]	[a]			
	[u]			
	[i]			
[د] [d]	[a]			
	[u]			
	[i]			

<p>[ش] [س]</p>	<p>[a] [u] [i]</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.162307</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.225203</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.162307</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.107886</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.225203</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.149241</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.129898</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.225203</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.129898</p>
<p>[خ] [ه]</p>	<p>[a] [u] [i]</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.142775</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.0900916</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.145767</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.0863878</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.158752</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.156011</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.189888</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.228788</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.158817</p>
<p>[و] [ه]</p>	<p>[a] [u] [i]</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.185886</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.188226</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.000000</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.282385</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.000000</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.138224</p>	 <p>Frequency (Hz) 5000 0 0 Time (s) 0.228822</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.000000</p>  <p>Frequency (Hz) 5000 0 0 Time (s) 0.188224</p>

<p>[d]</p>	<p>[i]</p> <p>[u]</p> <p>[a]</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>
<p>[s]</p> <p>[ɜ]</p>	<p>[i]</p> <p>[u]</p> <p>[a]</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>
<p>[h]</p> <p>[ɛ]</p>	<p>[i]</p> <p>[u]</p> <p>[a]</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>	 <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>  <p>Frequency (Hz)</p> <p>Time (s)</p>

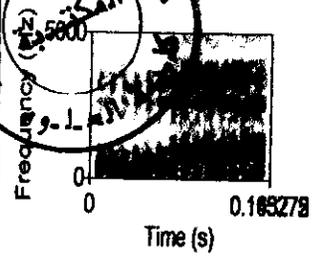
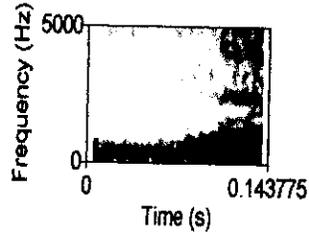
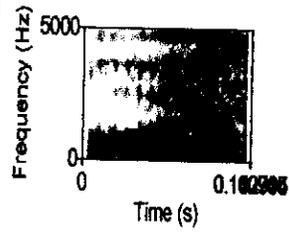
<p>[r] [m]</p>	<p>[a] [u] [i]</p>	  	  	  
<p>[ʊ] [n]</p>	<p>[a] [u] [i]</p>	  	  	  
<p>[ʊ] [l]</p>	<p>[a] [u] [i]</p>	  	  	  

[ɔ] [ə]	[a]	 <p>Frequency (Hz) 5000 Time (s) 0.117308</p>	 <p>Frequency (Hz) 5000 Time (s) 0.166824</p>	 <p>Frequency (Hz) 5000 Time (s) 0.2288233</p>
	[u]	 <p>Frequency (Hz) 5000 Time (s) 0.227068</p>	 <p>Frequency (Hz) 5000 Time (s) 0.227068</p>	 <p>Frequency (Hz) 5000 Time (s) 0.2288233</p>
	[i]	 <p>Frequency (Hz) 5000 Time (s) 0.2288233</p>	 <p>Frequency (Hz) 5000 Time (s) 0.2288233</p>	 <p>Frequency (Hz) 5000 Time (s) 0.2288233</p>
	[a]	 <p>Frequency (Hz) 5000 Time (s) 0.157578</p>	 <p>Frequency (Hz) 5000 Time (s) 0.166824</p>	 <p>Frequency (Hz) 5000 Time (s) 0.212839</p>
	[u]	 <p>Frequency (Hz) 5000 Time (s) 0.161103</p>	 <p>Frequency (Hz) 5000 Time (s) 0.0857063</p>	 <p>Frequency (Hz) 5000 Time (s) 0.070475</p>
	[i]	 <p>Frequency (Hz) 5000 Time (s) 0.268067</p>	 <p>Frequency (Hz) 5000 Time (s) 0.126053</p>	 <p>Frequency (Hz) 5000 Time (s) 0.126053</p>
[ɔ] [r]	[a]	 <p>Frequency (Hz) 5000 Time (s) 0.157578</p>	 <p>Frequency (Hz) 5000 Time (s) 0.166824</p>	 <p>Frequency (Hz) 5000 Time (s) 0.212839</p>
	[u]	 <p>Frequency (Hz) 5000 Time (s) 0.161103</p>	 <p>Frequency (Hz) 5000 Time (s) 0.0857063</p>	 <p>Frequency (Hz) 5000 Time (s) 0.070475</p>
	[i]	 <p>Frequency (Hz) 5000 Time (s) 0.268067</p>	 <p>Frequency (Hz) 5000 Time (s) 0.126053</p>	 <p>Frequency (Hz) 5000 Time (s) 0.126053</p>

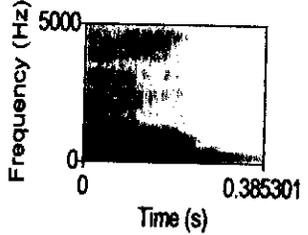
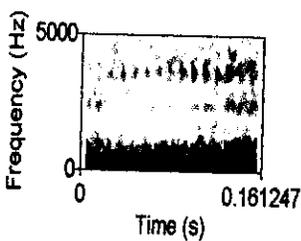
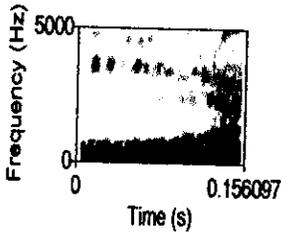


[g]  
[w]

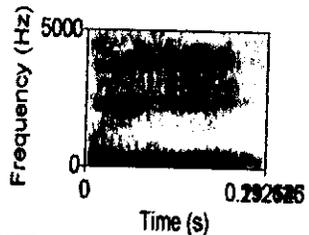
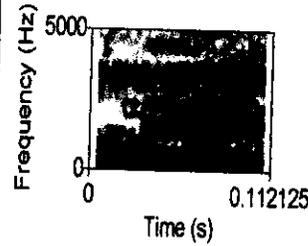
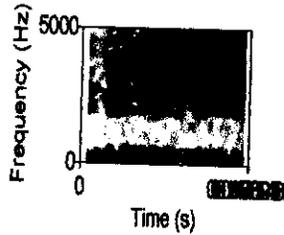
[a]



[u]

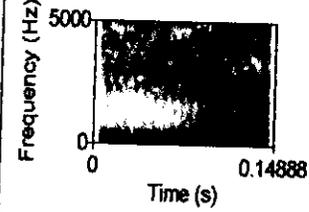
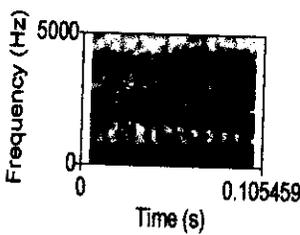
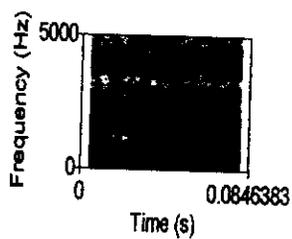


[i]

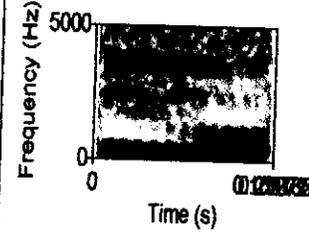
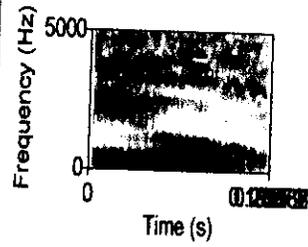
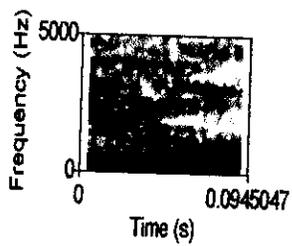


[ʃ]  
[y]

[a]



[u]



[i]

