الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي و البحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique

Mémoire de Projet de Fin d'Études

présenté par

FERRAH Dalila

pour l'obtention du diplôme de Master II en Electronique spécialité Signaux en Ingénierie des Systèmes et Informatique Industriel (SISII)

Thème

Reconnaissance Automatique De La Parole Arabe Par Les HMMs A L'aide De HTK Toolkit

Proposé par : **M^r^ BENSELAMA Zoubir** & **M^r^ BENCHERIF M^ed^ . Abd**

Année Universitaire 2013-2014

Remerciements

Je remercie tout d'abord ALLAH notre créateur qui ma guidé dans le chemin et de m'avoir donné la force et la patience pour mener à bien ce travail.

En préambule à ce mémoire, je souhaite adresser mes remerciements les plus sincères aux personnes qui m'ont apporté leur aide et qui ont contribué à l'élaboration de ce mémoire ainsi qu'à la réussite de cette formidable année universitaire.

Je tiens à remercier sincèrement Monsieur **BENSELAMA Zoubir** d'abord en tant que promoteur de ce mémoire ensuite pour m'avoir donné plus de confiance en moi, pour sa générosité et la grande patience dont il a fait preuve tout le long de mon travail malgré ses nombreuses charges académiques et professionnelles.

Mes remerciements s'adressent également à Monsieur **BENCHERIF Mohamed Abd** en tant que responsable de ce mémoire et son suivi dans la partie programmation; qui m'a appris à être rigoureux dans mes travaux afin d'éviter les obstacles qui pouvaient se présenter, qui s'est toujours montré attentif et disponible tout au long de la réalisation de ce mémoire, ainsi que pour l'inspiration, l'aide et le temps qu'il a bien voulu me consacrer sans quoi ce mémoire n'aurait jamais eu autant de succès.

Je remercie madame **AKAK.A** , pour son encouragement.

Enfin, j'exprime toute ma gratitude à tous les consultants et internautes rencontrés lors des recherches effectuées et qui ont accepté de répondre à mes questions avec gentillesse.

Dédicaces

À la mémoire de mon père.

À ma mère et à mes sœurs et frères ; le petit ange Oussama, Razim et Ibtissem. Vous vous êtes dépensés pour moi sans compter. En reconnaissances de tous les sacrifices consentis par tous et par chacun pour me permettre d'atteindre cette étape de ma vie.

À mes oncles, tantes, cousins et cousines affectueuses reconnaissances.

À mes enseignants de l'école primaire jusqu'à l'université dont les conseils précieux m'ont guidé ; qu'ils trouvent ici l'expression de ma reconnaissance.

À mes amies : ARKAM Meriem , M'BARKI Amina, HADDAD Affaf, RAF3I Hanen , ZAAROUR Nassrine , KATMIR FZ ,et à toute la promo mastère électronique (SISII) 2013 / 2014 et à leurs familles.

Je vous remercie pour votre patience et pour m'avoir aidé à avancer. Vous êtes tous pour moi comme une seconde famille.

Merci d'être toujours près de moi dans mes joies et mes peines

À tous mes camarades de département d'électronique.

FERRAH Dalila

ملخص:

التعرف التلقائي على الكلام (HTK) يظهر مجموعة هامة من التطبيقات في الطبيعة الصعبة و المتنوعة المتعلقة بالملايين من الناس في جميع أنحاء العلم . و يمكن أن نتوقع بأن الكلام اصبح جزءا من الوسائط المتعددة بين الإنسان و النظام الآلي وثانيا بسبب الوعي المتزايد لهذه التكنولوجيا التي لا تزال غير واضحة حتى الان لإغلبية الناس.

و بالنظر الى أهمية التعرف التلقائي على الكلام () و ضعت العديد من البرامج نذكر على سبيل المثال البرنامج الاكثر استعمالا (HTK) القا ئم على نماذج ماركوف المخفية .

في هذا العمل قمنا بإنشاء نظام التعرف عاى الكلام العربي بواسطة نماذج ماركوف المخفية باستخدام HTK

كلمات المفاتيح:

Résumé :

La reconnaissance automatique de la parole (RAP) donne aujourd'hui lieu à un ensemble important d'applications de nature et de difficulté très variées, concernant quotidiennement des millions de personnes à travers le monde. On peut prévoir que la parole fera de plus en plus partie des interfaces multimédia entre un utilisateur et un système automatique, d'une part grâce à l'amélioration de la robustesse des systèmes de reconnaissance automatique de la parole et, d'autre part, du fait de la sensibilisation croissante du grand public à cette technologie encore peu connue. Vue l'importance de la RAP, plusieurs logiciels ont été développés, parmi les plus connus, on trouve le HTK (Hidden Markov Model Toolkit) qui est basé sur les Modèles de Markov Cachés.

Dans ce mémoire, nous nous sommes intéressés à la réalisation d'un système de reconnaissance automatique de la parole Arabe par les HMMs à l'aide de l'outil HTK.

Mots clés : Reconnaissance automatique de la parole, Modèles de Markov Cachés, l'outils Htk

Abstract :

The automatic speech recognition (RAP) now gives rise to an important set of applications of nature and difficulty varied on a daily basis millions of people around the world. It is anticipated that the speech will increasingly part of multimedia interfaces between a user and an automatic system, firstly by improving the robustness of automatic recognition speech systems, secondly, because of the growing public awareness of this technology is still little known. The significance of the RAP, several software packages have been developed, the most known are the HTK (Hidden Markov Model Toolkit) which is based on Hidden Markov Models.

In this work, we are interested in the creation of a system of Arabic speech recognition based on HMM using HTK.

Keywords : speech recognition, acoustic model, Hidden Markov Models, HTK.

Listes des acronymes et abréviations

Table des matières

Les titres **liste des figures** et **liste des tableaux** ne figurent pas dans la table des matières.

Liste des figures

Liste des tableaux

Introduction générale

La parole constitue sans aucun doute l'un des moyens les plus utilisés pour la communication entre les êtres humains. Ceux-ci ont très rapidement cherché à l'intégrer dan les interfaces Homme Machine. Cela est rendu réalisable grâce aux efforts consentis de nombreuses équipes de recherche à travers le monde entier.

La parole comme un moyen de dialogue homme-machine efficace, a donné naissance à plusieurs travaux de recherche dans le domaine de la Reconnaissance Automatique de la Parole (RAP). Un système de RAP est un système qui a la capacité de détecter à partir du signal vocal la parole et de l'analyser dans le but de transcrire ce signal en une chaîne de mots ou phonèmes représentant ce que la personne a prononcé.

La reconnaissance de parole présente de nouvelles difficultés car on ne connait pas le nombre de mots qui composent une phrase, ni les frontières de chaque mots. Pour ces raisons, au cours de ce mémoire, notre travail consiste à construire un système de reconnaissance de la parole Arabe à l'aide du logiciel particulier HTK toolkit .

Cette étude est basée sur une approche probabiliste, où on utilise les modèles de Markov cachés (HMM) pour modéliser les vecteurs des paramètres acoustique (MFCC), et donner la décision après avoir calculer le maximum de vraisemblance entre le modèle acoustique et le modèle linguistique, qui va introduire une erreur de reconnaissance. Pour sa part, le HTK va nous donner un rapport des différents taux de reconnaissance, de substitutions, de suppressions et d'insertion.

Ce mémoire est constitué de quatre chapitres :

- Le premier chapitre représente une vue générale sur la parole. Lors de ce chapitre, on présente en général tout ce qui concerne la parole, la production , ces caractéristiques et ainsi l'approche globale sur la langue Arabe Standard (AS) .
- Le deuxième chapitre aborde la reconnaissance automatique de la parole.
- Dans le troisième chapitre, nous avons décrit les Modèles de Markov Cachés et une présentation de l'outils HTK.
- Le quatrième chapitre illustre notre système de reconnaissance de la parole Arabe ainsi que les résultats obtenus et leurs interprétations.

Enfin, une conclusion générale et des perceptions sont données pour ouvrir la voie à des travaux futurs.

Chapitre 1 Généralités sur la parole

1.1 Introduction

La parole est un moyen essentiel pour la communication entre les humains, c'est la capacité de communiquer par un système de sons articulés émis par les organes de la phonation.

Dans ce chapitre nous allons décrire de manière générale le signale de la parole et ses caractéristiques , l'appareil phonatoire qui représente l'organe principal de la production vocale puis une présentation de manière succincte de quelques notions sur la langue Arabe Standard (AS).

1.2 La parole

La parole correspond à une variation de la pression de l'air causée par le système articulatoire [1]. La parole peut être vue comme une suite de sons vocales produits soit par des vibrations des cordes vocales (source quasi périodique de voisement) , soit par une turbulence créée par l'air s'écoulant dans le conduit vocal, lors du relâchement d'une forte constriction de ce conduit (sources de bruit non voisées) .[2]

La parole est formée de phonèmes et de transitions entre ces phonèmes. Plusieurs types de phonèmes existent: les voyelles, les consonnes fricatives, les nasales et les liquides.[3]

1.3 Le mécanisme de la production de la parole

La production de la parole réside dans les fluctuations de la pression de l'air engendrée , puis émise par l'appareil phonatoire , ces fluctuations constituent le signal

vocal. Elles sont détectées par l'oreille qui procède à une certaine analyse et les résultats sont transmis au cerveau qui les interprète .[4]

La génération de la voix n'est pas réalisée par un système propre, mais elle est assurée conjointement par les organes de l'appareil phonatoire et de l'appareil respiratoire .[5]

Pour comprendre le fonctionnement de l'appareil phonatoire lors de l'émission des sons , nous devons tenir comte de ses différents composants et organes. Ce dernier peut être divisé en trois (03) parties effectuant respectivement les fonctions de: Soufflerie (S), Vibration (V) et Articulation (A) . Figure (1.1)

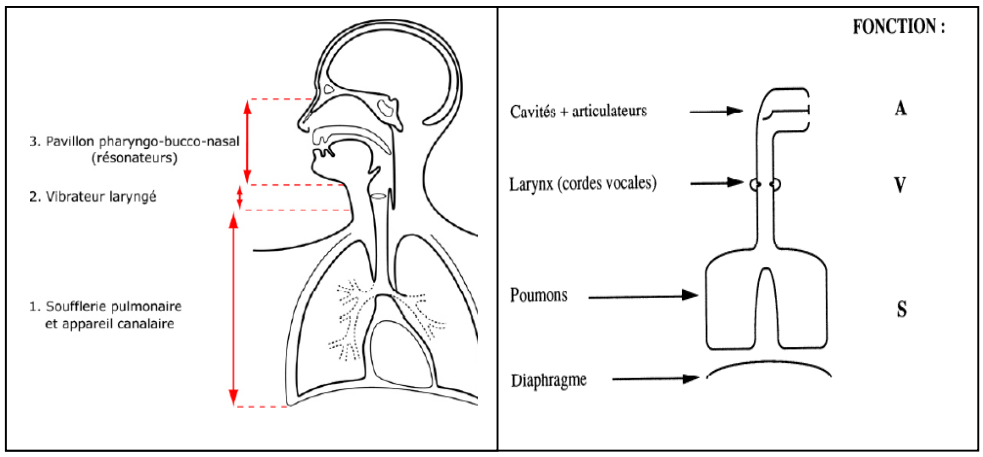


Figure 1.1 Schéma de l'appareil phonatoire.[6]

Les trois groupes d'organes assument les fonctions essentielles dans l'acte de **parole, ou phonation sont:**

- **L'appareil respiratoire** (diaphragme, poumons,trachées), soufflerie qui fournit l'energie et la qualité d'air nécessaire .

- **Le larynx** ,organe vibrant ou nait le son.

- **Le conduit vocal ,** formé des cavités résonantes supra-laryngées (pharynx,bouche,nez) ou s'effectue l'articulation proprement dite par les

changements de forme du tractus vocal. Ces changements résultent surtout des mouvements des lèvres , de la lague, du voile du palais et la machoire inférieure.

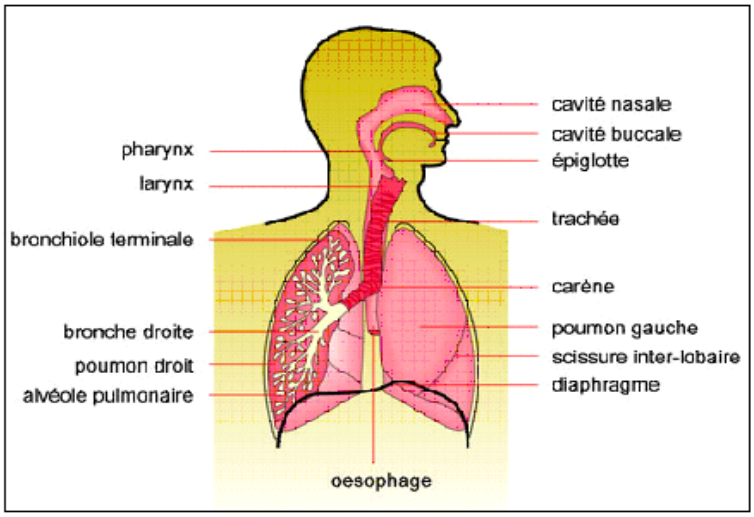


Figure 1.2 Le schéma général des organes de l'appareil phonatoire.[7]

1.3.1 Les poumons

Ils fournissent l'énergie nécessaire pour la production des sons. Cette énergie est assurée par le biais d'un mouvement cyclique de la respiration.

La respiration comprend deux (02) phases : l'inspiration et l'expiration , cette dernière assure l'opération de phonation et cela grâce à un flux d'air provenant des poumons. Ce flux s'appelle air *pulmonaire* (ou *pulmonique*) *égressif*.

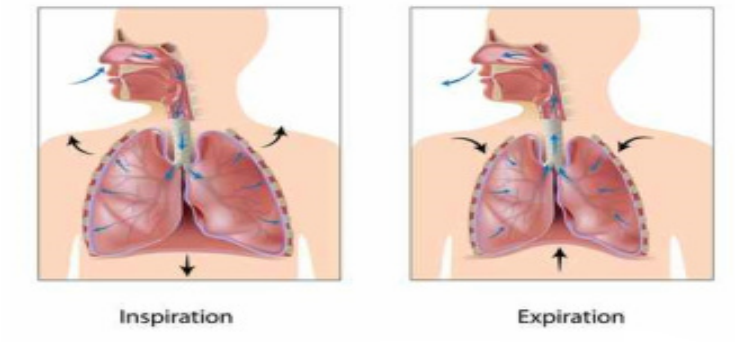


Figure 1.3 Inspiration et Expiration de l'air dans les poumons.[8

1.3.2 le larynx, organe vibrant

Il représente l'organe de la phonation, puisqu'il joue un rôle très important dans l'émission des sons vocaux .Le larynx est placé dans le cou à l'extrémité supérieure de l'arbre respiratoire (entre la trachée et le pharynx).

Le larynx n'est pas fixe dans le cou ; il se déplace de haut en bas quand on parle. Il s'élève pour les sons aigus et s'abaisse pour les sons graves.

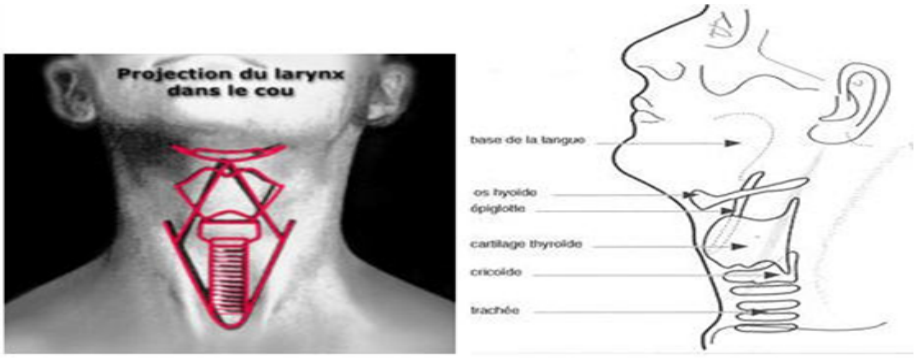
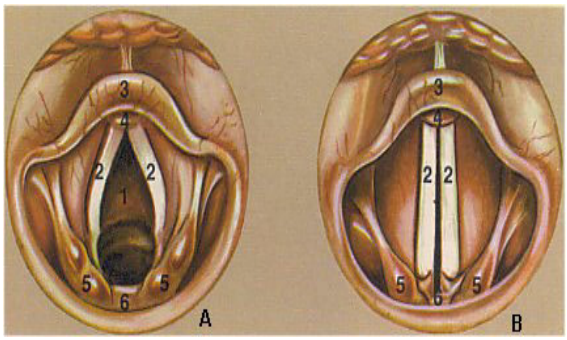


Figure 1.4 Le larynx. [9]

- **les cordes vocales**

Les cordes vocales sont en fait deux lèvres symétriques placées en travers du larynx. Ces lèvres peuvent fermer complètement le larynx et, en s'écartant progressivement, déterminer une ouverture triangulaire appelée glotte.



A: Glotte pendant la respiration.

B: Glotte pour la phonation

1:Glotte

2: Cordes vocales .

3: Epiglotte.

4, 6: les extrémités

5: Cartilages aryténoïdes

Figure 1.5 Les cordes vocales. [9]

Les cordes vocales ont trois (03) positions fondamentales :

- Soit, elles sont ***écartées*** : la glotte est ouverte et l'air circule librement, c'est *respiration* . Lors d'une inspiration profonde, l'écartement est maximal, lors de la respiration normale, l'écartement est moyen.
- Soit, elles sont ***accolées*** : la glotte est alors fermée et l'air ne passe pas. C'est *l'apnée*.
- Soit, les cordes sont ***rapprochées*** : la glotte est variable. C'est *la phonation* ou *le voisement*.

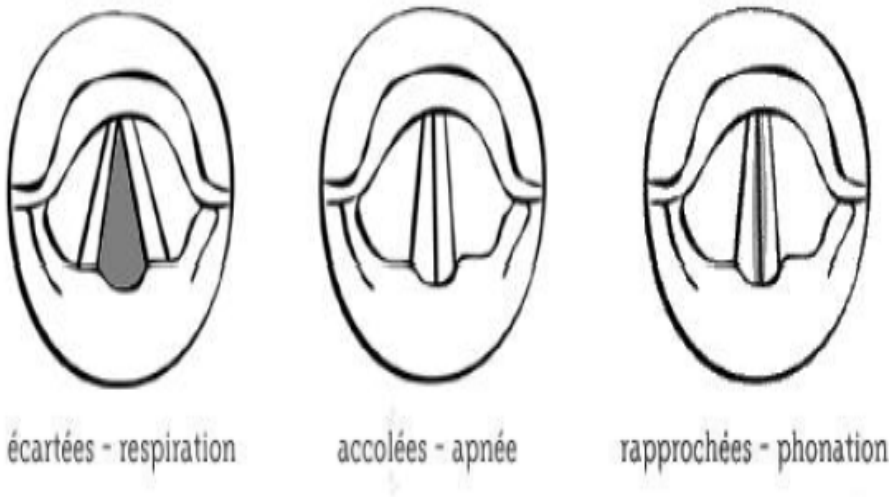


Figure 1.6 Les positions fondamentales des cordes vocales .[6]

1.3.3 La cavité résonnante

La majorité des sons du langage sont le fait du passage d'une colonne d'air venant des poumons, qui traverse un ou plusieurs résonateurs de l'appareil phonatoire.

Les résonateurs principaux sont : le pharynx ,la cavité buccale , la cavité labiale et les fosses nasales.

a Le pharynx (ou cavité pharyngale):

Est un conduit musculo-membraneux situé entre la bouche et l'œsophage d'une part et entre les fosses nasales et le larynx d'autre part. La paroi du pharynx est constituée de muscles constricteurs. Effet d'une constriction : modification du diamètre du pharynx. La racine de la langue peut également reculer ou avancer et donc agir sur le volume de cette première cavité supra glottique.

b Les fosses nasales (ou cavités nasales):

Sont deux cavités cunéiformes séparées par une cloison verticale médiane et sont recouvertes de muqueuses. Une résonance nasale est très caractéristique (nasillement). L'air passe par le nez lorsque le voile du palais (prolongement musculaire du palais osseux) est rabaissé : passage oro-nasal ouvert.

c La bouche (ou cavité buccale):

Est séparée des fosses nasales par une cloison appelée le palais. Dans cette cavité se situent des articulateurs, certains fixes (passifs), d'autres mobiles (actifs).

d La cavité labiale: Est une cavité que l'on crée lorsqu'on projette en avant les lèvres.

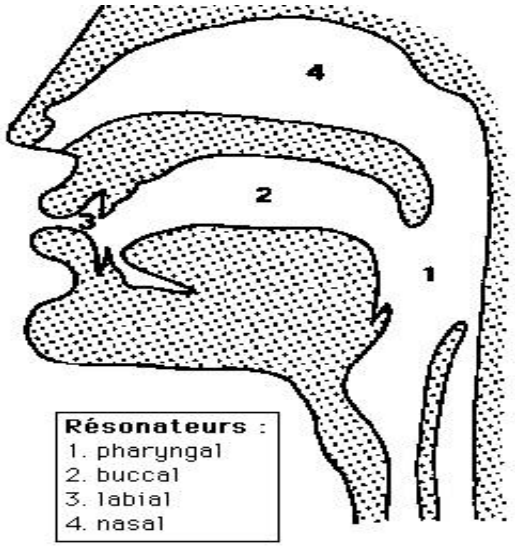


Figure 1.7 la cavité résonnante.[6]

1.4 Représentation schématique de l'appareil phonatoire humain

Pour comprendre le principe utilisé en production de la parole , il faut donner une représentation schématique de l'appareil phonatoire illustrée dans la figure (1.8).

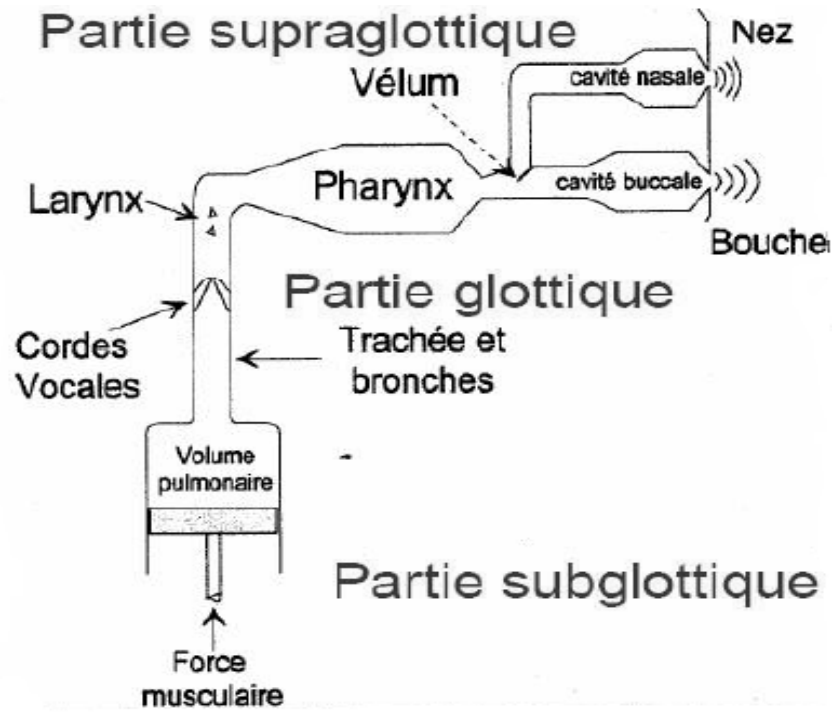


Figure 1.8 Schéma représentatif de l'appareil phonatoire humain.[6]

1.4.1 Partie sub-glottique ou appareil respiratoire (diaphragme poumons, trachée) qui fournit l'énergie nécessaire à la phonation en insufflant l'air vers la partie glottique.

1.4.2 Partie glottique ou larynx (ensemble de cartilages, ligaments et muscles) contenant les cordes vocales (replis tendus horizontalement qui, sous l'effet des muscles, jouent un rôle de valve vis-à-vis de l'air des poumons libérant ainsi un flux d'air vers la partie supra- glottique.

1.4.3 Partie supra- glottique ou conduit vocal, formé des cavités orales (pharyngienne et buccale), à géométrie variable, en fonction des éléments articulateurs (langue, mâchoire inférieure, lèvres) et des cavités nasales, à géométrie fixe, pouvant être couplées aux cavités orales par abaissement du voile du palais.

1.5 Notions de phonétique

En linguistique, un ***phonème*** est la plus petite unité distinctive c'est-à-dire permettant de distinguer des mots les uns des autres que l'on puisse isoler dans la chaîne parlée.[6]

La phonétique classe les phonèmes en ***voyelles***, ***consonnes*** et ***semi-voyelles*** (***semi-consonnes***). La distinction entre voyelles et consonnes s'effectue de la manière suivante :

- Si le passage de l'air se fait librement à partir de la glotte, on a affaire à une ***voyelle.***
- si le passage de l'air à partir de la glotte est obstrué, complètement ou partiellement, en un ou plusieurs endroits, on a affaire à une ***consonne*.**

Les semi-voyelles présentent la même articulation que les voyelles , mais se comportent dans la syllabe comme les consonnes : plus précisément , les consonnes et les semi-voyelles ne peuvent construire à elles seules une syllabe , les voyelles si (par exemple : le mot *abbaye* [a / be / i] comporte des voyelles alors que le mot abeille [a / bɛj] comporte une semi-voyelle notée [j].

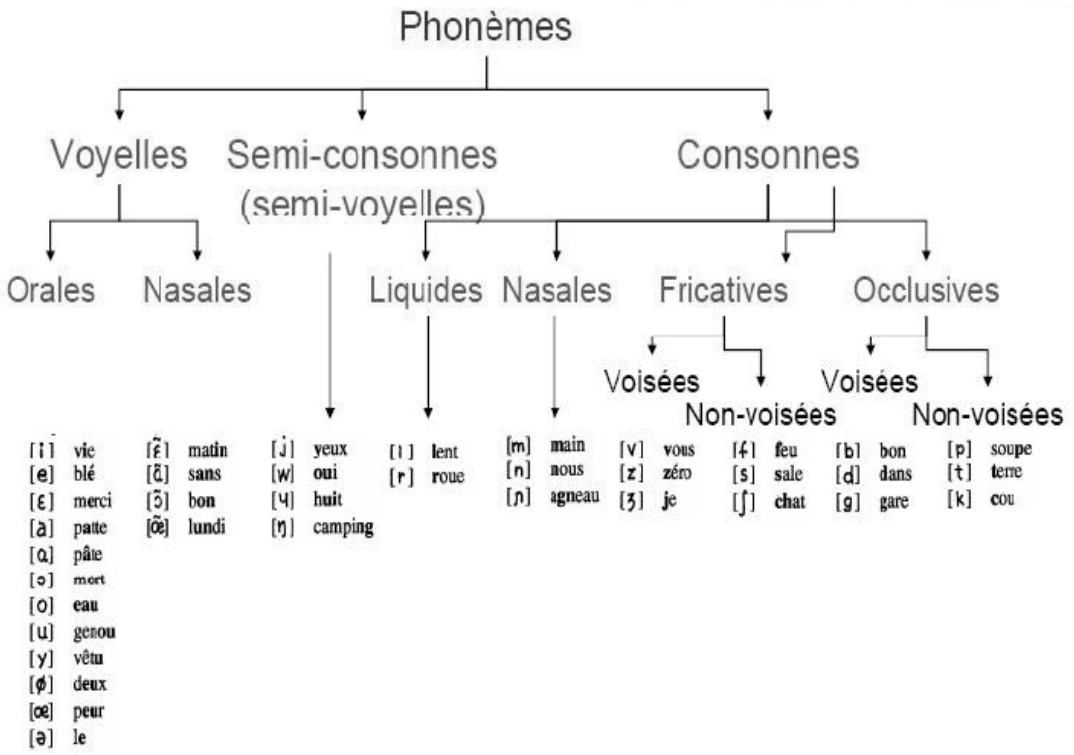


Figure 1.9 Les types de sons (phonèmes de la langue Française).[6]

1.5.1 Les voyelles

Les voyelles sont produits lorsque le conduit vocal est ouvert, est la formes des cavités (essentiellement la bouche) modifie le timbre , les voyelles sont prononcés soit orales ou nasales selon que la cavité nasale n'est pas ou est mise en parallèle à la cavité buccale, par exemples :

- ***Orales :*** idée , modèle, alarme, pate, corps, beau, élu, loup, peur, petite.
- ***Nasales :*** matin, temps, bon, brun.

1.5.2 Les consonnes

Les consonnes sont produit lorsque les rétrécissement apparait dans l'appareil phonatoire et cordes vocales peuvent vibrer ou laisser passer librement l'air (sons voisés et non voisés), ainsi qu'ils sont fricatives si le rétrécissement est partiel ou occlusives (plosives) si une occlusion totale apparait dans l'appareil phonatoire,

causant une augmentation de la pression et un relâchement brutal de celle-ci lors de l'ouverture , et voila quelques exemples:

- ***Fricatives non voisées:*** chanter , soupe, facile.
- ***Fricatives voisés:*** jouer, zéro, vélo.
- ***Occlusives non voisées:*** papa, tapis, carte.
- ***Occlusives voisées:*** bébé, début, gauche.
- ***Liquides :*** lapin , rayon.
- ***Nasales:*** maman, nord, grogner.

1.6 Caractéristiques spécifiques du signal vocal

Quelques caractéristiques du signal vocale sont résumées dans ce que suit :

1.6.1 La non stationnarité

Le signal vocal n'est pas un signal stationnaire puisque le conduit vocal se déforme d'une façon continue et les paramètres du modèle sont variables dans le temps.[10]

1.6.2 La continuité

Le langage oral est une suite continue de sons sans séparation entre les mots . Les silences correspondent en général à des pauses de respiration dont l'occurrence est aléatoire. Il peut très bien y avoir des intervalles de silences au milieu d'un mot et aucun intervalle entre deux mots successifs. Par conséquent, il est très difficile de déterminer le début et la fin des mots composant la phrase .[2]

1.6.3 La variabilité

La parole présente une très grande variabilité qui résulte de plusieurs facteurs et ceci que ce soit pour un même ou plusieurs locuteurs. [11]

On distingue trois(03) sortes de variabilités lors de la production de la parole:

a Variabilité intra-locuteur

Elle concerne les différences de production du signal parole chez un même locuteur. Puisque un locuteur ne prononcera jamais deux fois de manière identique un même mot.

Plusieurs critères peuvent être responsables de ces différences :

- La fatigue.
- L'état émotionnel du sujet : une émotion telle que la peur affecte le timbre et le rythme de la voix.
- Les maladies affectant les organes de la voix.

b Variabilité interlocuteurs

Les différences morphologiques et culturelles font que les paramètres vocaux sont spécifiques à chaque locuteur . Cette variabilité concerne un ensemble d'individus ou chacun d'eux a ses propres caractéristiques , en prononçant la même phrase ,avec le même rythme , le même accent , ainsi que le même timbre.

Cette variabilité est aussi due principalement à la différence de l'âge, du sexe, de la physiologie et de l'origine géographique de chaque individu.

c Variabilité contextuelle

Est liée au phénomène de la coarticulation des sons entre eux tels que deus sens voisins peuvent s'influencer mutuellement. Cette variabilité est appelée aussi la variabilité due à l'environnement et cela puisque l'environnement peut diminuer le signal vocal généré sans que le locuteur ne modifie son mode d'élocution.

1.7 Etude acoustique de la parole

L'onde de la parole couvre quasiment toute l'étendue du spectre audible. En pratique, on peut se limiter à la bande 50 - 5000 Hz. Le signal de la parole est un signal très riche en informations , pour cela nous abordons le signal de la parole dans ce

paragraphe du point de vue acoustique en évaluant ses paramètres à savoir la fréquence fondamentale , l'intensité, la durée , l'intonation , la résonance, ...etc.

1.7.1 La fréquence fondamentale F_0 (ou pitch)

La fréquence fondamentale **F_0** est la fréquence de vibration des cordes vocales , elle varie d'une personne à une autre en fonction de la longueur et de la tension des cordes vocales de chaque personne.

La fréquence fondamentale permet de diviser l'ensemble des sons de la parole humaine en trois (03) grandes classes:[1]

- 70 à 250 Hz chez les hommes.
- 150 à 400 Hz chez les femmes.
- 200 à 600 Hz chez les enfants.

1.7.2 L'intensité ou l'énergie

Elle est résultante de la pression sou glottique. Généralement, elle exprime le volume sonore d'un phonème et dans le cas d'un voisement elle représente l'amplitude des vibrations des cordes vocales. Elle est exprimée pour un signal échantillonné X_n par:

$$E = \sum_{N=1}^{T} x_n^2 \qquad tel\ que : n = 1,..,T \tag{1.1}$$

A l'échelle perceptive , elle est exprimée en décibels (dB) par:

$$E_{dB} = 10 * log_{10}\left(\sum_{N=1}^{T} x_n^2\right) \tag{1.2}$$

1.7.3 La durée

La durée est le paramètre acoustique le plus délicat à évaluer, car il ne dépond d'aucun corrélat biologique , contrairement à F_0 et l'intensité (qui dépend respectivement de la tension des cordes vocales et de la pression sous glottique).

Pour calculer la durée d'un phonème , il faudrait se fixer deux événements qui délimitent ses repères initial et final. La durée représente généralement le temps de la prononciation d'un phonème.

Pour mesurer une durée quelconque, il faudrait au préalable désigner, d'une part, les unités à mesurer et d'autre part , leurs repères (les frontières) dans le signal parole. Elles peuvent concerner les phonèmes , distance entre voyelles, les pauses, ...etc.

Il existe deux types de durées:

- La durée observée, qui correspond à la mesure objective du temps de l'activation des organes de phonation.

- La durée perçue, est liée au mécanisme de la perception et elle est fréquemment utilisée dans le cas des occlusives puisqu'elles sont caractérisées par la durée de réalisation no continue.

1.7.4 L'intonation

Le terme de l'intensité a deux définitions possibles:

- Au sens strict, ce mot désigne les changements relatifs à la hauteur de la voix, que certains chercheurs confondent avec le mot mélodie.
- Le sens le plus étendu de ce terme fait aussi référence aux changement de la durée et de l'intensité. Dans ce dernier cas, il s'apparente au mot prosodie.

1.7.5 La résonnance

Un système vibratoire possède généralement une fréquence de vibration dite propre, correspondant à son mode d'oscillation libre. En présence d'une excitation extérieure , ce système entre en vibration à la fréquence imposée par l'extérieur. Mais l'amplitude dépend fortement de la fréquence; elle est maximale lorsque la fréquence imposée est égale à le fréquence propre du système.

1.8 Vue d'ensemble de la langue Arabe Standard (AS)

La langue Arabe Standard est la langue dans laquelle est écrit la Saint Coran et que l'on trouve aussi enseignée dans les écoles . Elle est la langue officielle de nombreux pays , et elle est également la langue employée dans la plupart des écrits et à l'oral dans les situations officielles ou formelles(discours religieux, politiques, journaux télévisés,...etc.).

L'alphabet de la langue Arabe se compose de 28 lettres qui sont toutes des consonnes ou [huruuf] plus la hamza et 6 voyelles ou [harakaat] (3 courtes et 3 longues).[12]

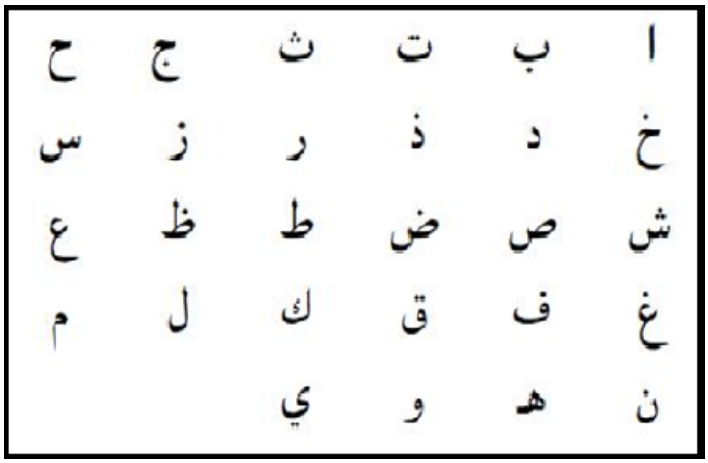


Figure 1.10 L'alphabet de la langue Arabe.[12]

1.8.1 Les consonnes de l'Arabe Standard

La langue Arabe s'écrit de droite à gauche , les lettres peuvent changer leur forme de présentation selon leurs positions (au début, au milieu ou à la fin) à l'intérieur des mots (Tableau I.1) . Toutes les lettres se lient entre elles sauf les sept (07) lettres suivantes **(ا و ر ز د لا ذ)** qui ne se joignent pas à gauche.

/ ت /[t] au début du mot		**/ ت / [t] au milieu du mot**		**/ ت / [t] à la fin du mot**	
تمر	تـ	كتب	ـتـ	حياة زيت مدرسة	ة ت ـة

Tableau 1.1 Exemples de variations de la lettre / ت / [t] dans les différentes positions (initiale, médiane et finale)

En réalité; on peut diviser les 28 consonnes en deux groupes :

- ***14 consonnes solaires*** : qui assimilent le /ل / de l'article, c'est -à-dire lors de la prononciation on élimine le son qui correspond à la lettre / ل /.
  Exemple : le mot/ الشمس / [a§§amsu] qui signifie le soleil , sera prononcé [a§§amsu] et pas [al§§amsu].
- ***14 consonnes lunaires*** : qui se prononcent / ل / de l'article.
  Exemple: le mot / القافلة/ sera prononcé [alqaafila] qui signifie la caravane.

Les consonnes solaires	**Les consonnes lunaires**
ت ث د ذ ر ز س ش ص ض ط ظ ل ن	**أ ب ج ح خ ع غ ف ق ك ـه م و ي**

Tableau 1.2 Classification des consonnes de la langue arabe.[12]

Suivant les organes de l'appareil phonatoire mis en jeu et leurs excitations; il est possible de faire une autre classification des consonnes tout en se basant sur le mode et lieu d'articulation, comme il est illustré dans la figue (1.11).

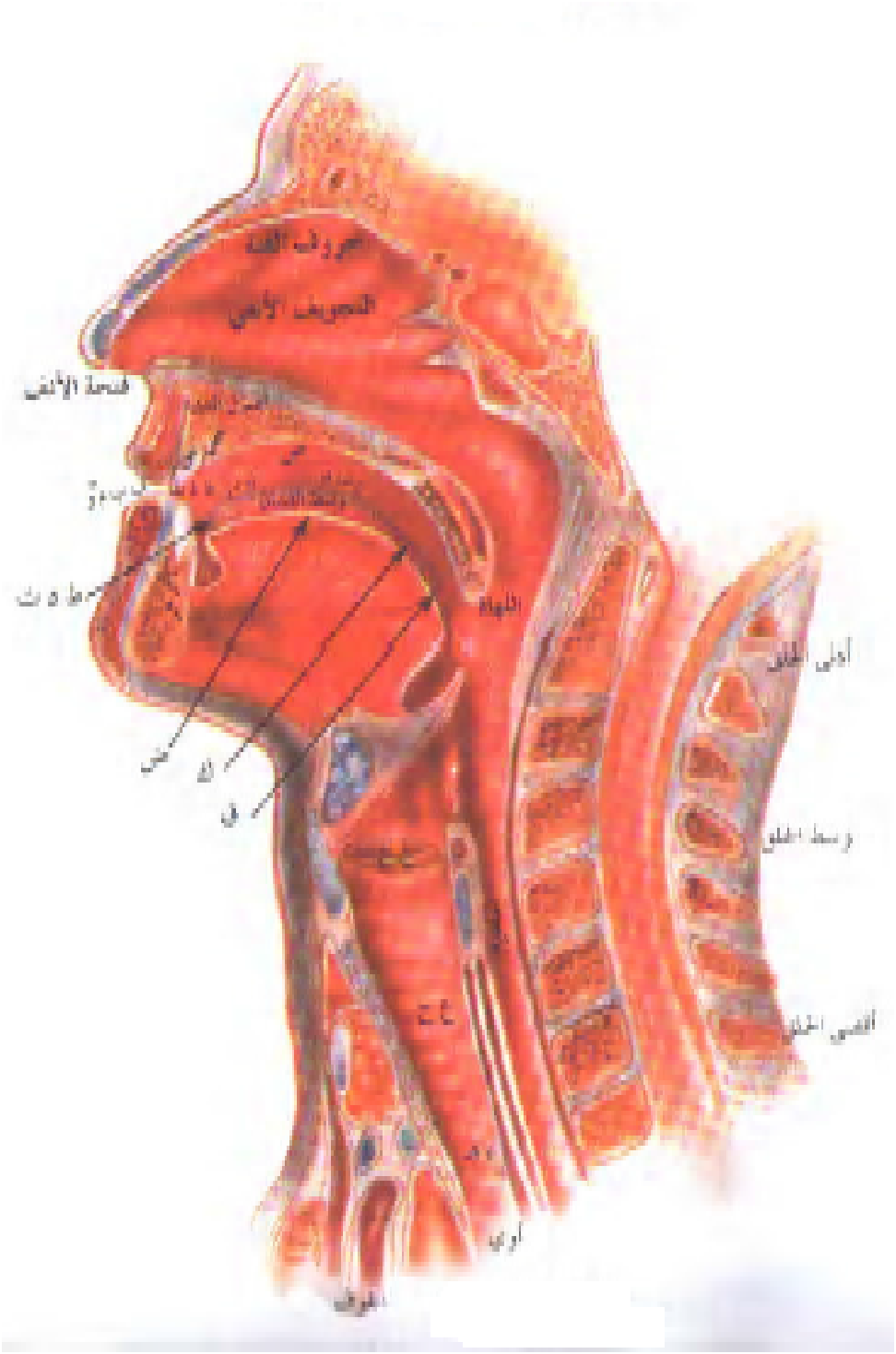


Figure 1.11 Les lieux d'articulation des 28 [huruuf] de l'Arabe Standard [13]

a Classification des consonnes selon le mode d'articulation

Suivant la classification des sons , on peut distinguer plusieurs types de consonnes:

- ***Voisées / non voisées (sonores / sourdes)***

L'air nécessaire pour la production des sons sort des poumons et passe par la trachée , en haut de la trachée se trouve une boite en cartilage qu'on appelle le larynx. Suspendues dans le larynx on trouve deux bandes de tissu élastique , qu'on appelle les codes vocales ou la glotte . Si les cordes vocales sont ouvertes , on entend un son non voisé ou sourd comme [p], si elles se rapprochent et vibrent , on a un son voisé comme [v] .

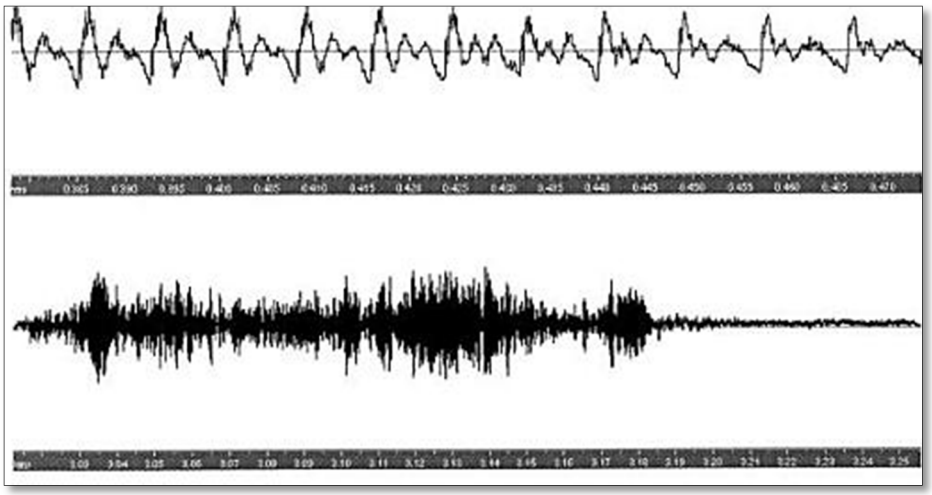


Figure 1.12 Exemples de son voisé (haut) et non-voisé (bas).[11]

✓ **Les sons voisés**

Dans le cas des phonèmes voisés le flux d'air ***p*** est un train d'impulsion de période ***N***. Ce flux d'air est modifié par les contributions glottales ***g***, rayonnement (lèvres) ***r*** et celle du conduit vocal ***v*** (figure I.14). Le signal de parole ***Y*** résultant est la convolution de ***P*** par les réponses impulsionnelles ***g*** , ***r***, ***v*** des trois parties du processus de production de la parole [15] comme le présente l'équation suivante:

$$Y = P * g * v * r \qquad (1.3)$$

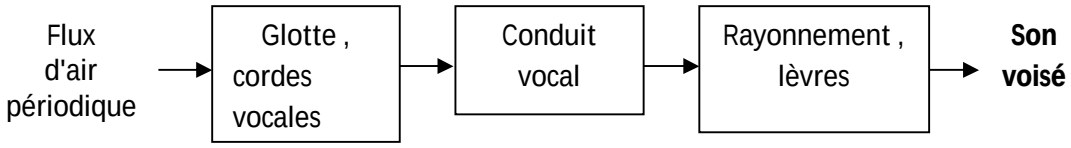


Figure 1.13 Processus de production de la parole dans le cas les phonèmes voisés.[14]

- ✓ **Les sons non voisés**

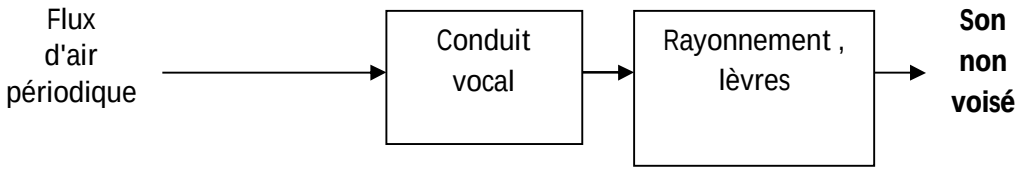


Figure 1.14 Processus de production de la parole dans le cas des phonèmes non voisés .[14]

Le son non voisé peut être considéré comme un bruit blanc qui résulte d'un écoulement turbulent de l'air à travers le conduit vocale(figure 1.14) . Sa forme d'onde ne présente aucune périodicité. Dans ce cas les cordes vocales ne vibres pas . Le flux d'air ***u*** est considéré comme un bruit blanc .[15]

$$Y = u * v * r \qquad (1.4)$$

- ***Occlusives ou plosives / constrictives ou fricatives***

Le premier type de consonnes est caractérisé par une fermeture complète (occlusion) en point du conduit vocal. La détente de cette occlusive s'accompagne d'un bruit explosif de la consonne occlusive [15].

Les sons du deuxième type sont générés par une constriction en un point de conduit vocale. Cette dernière est accompagnée par un passage continu de l'air.

- ***L'opposition nasale / orale***

Dans le premier cas le son est produit à travers un couplage entre les cavités pharyngo-buccale et nasale et dans le second l'air passe par la cavité buccale seulement (figure 1.15).

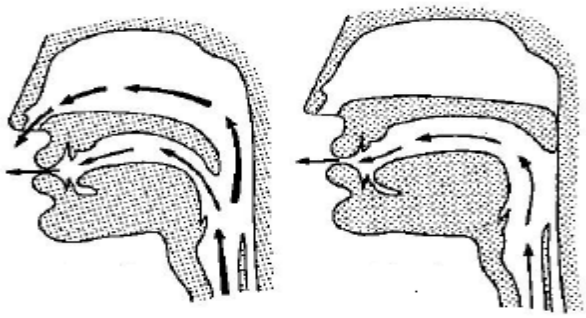


Figure 1.15 L'opposition nasale / orale.[6]

- ***Liquides***

L'articulation des liquides ressemble à une voyelle , la seule différence réside dans la fermeture partielle de conduit vocal (c'est le cas de / ل / [l]).

La classe des liquides est parmi les sons les plus difficiles à segmenter car elle influence les sons voisins progressivement et régressivement (phénomène de l'assimilation).

- ***Vibrantes***

Le passage de l'air dans une consonne vibrante est interrompu par des brèves occlusions successives.

b Classification des consonnes selon le lieu d'articulation

Les lieu d'articulation est la zone du conduit vocal qui participe à la formation du son. Il présente la position de la constriction totale (cas des occlusives) ou partielle (cas des fricatives) d'une zone spécifique du conduit vocal lors du passage de l'air provenant des poumons. Le lieu d'articulation peut être bilabiale, labiodentale, dentale, glottale,...etc.

1.8.2 Les voyelles de l'Arabe Standard

En Arabe Standard chaque consonne ou [harf] est suivie par une voyelle [harakatun] pour qu'elle puisse êtres produite. Cette voyelle corresponde au mouvement aéro-organique qui assure la réalisation de ce [harf].

Cependant , les voyelles ne sont utilisées que pour des textes sacrés et didactiques , les textes courants rencontrés dans les journaux et les livres n'en comportent habituellement pas [14].

- Les 6 voyelles peuvent être divisées en deux classes illustrées dans le tableau ci-après:

Les voyelles courtes	Les voyelles longues (almadd)
ـَ [a] Fathatun	ـَا [aa]
ـُ [u] Dammatun	ـُو [uu]
ـِ [i] Kasratun	ـِي [ii]

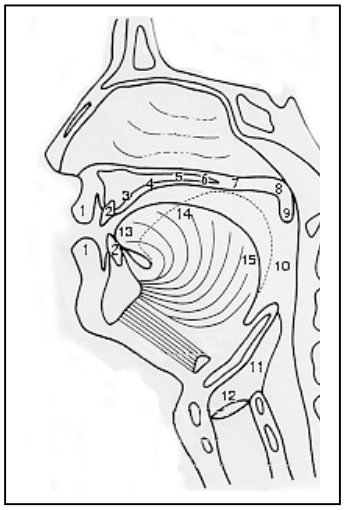
Tableau1.3 Classification des voyelles de l'arabe Standard

- Les voyelles sont ajoutées au dessus ou au dessous des consonnes (ـَ ـِ ـُ ـْ) . Elles sont nécessaires à la lecture et à la compréhension correcte d'un texte , le tableau (1.4) donne un exemple pour les mots كتب et مدرسة. L'absence des voyelles génère une certaine ambiguïté à deux niveaux :
  - ✓ Sens du mot.
  - ✓ Difficulté à identifier sa fonction dans la phrase.

Mot sans voyelles	1 ière Interprétation		2 ème Interprétation		3 ème Interprétation	
كتب	كَتَبَ	Il a écrit	كُتب	Il a été écrit	كتب	Des livres
مدرسة	مَدْرَسَةٌ	Ecole	مدرسة	Enseignante	مدرسة	enseignée

Tableau 1.4 Ambiguïté causée par l'absence de voyelles pour les mots كتب et مدرسة

Selon le contexte la durée d'une voyelle longue est environ double de celle voyelle courte, de plus les différentes voyelles courtes se diffèrent entre elles par leurs lieux d'articulation et le degré d'ouverture dub conduit vocal (ouvert, fermé, semi fermé) illustrés dans la figure (1.16).



Palatales (antérieures)	Centrales	Vélaires (postérieures)
[Kasratun]		[Dammatun]
	[fathatun]	

Figure 1.16 Les lieux d'articulation des voyelles courtes de l'Arabe .[14]

Si une consonne n'est liée a aucune voyelle , elle doit comporter un petit rend qu'on appelle [sukuun].

Exemple : دَرْسُ[darsun] qui signifie la leçon.

la chadda ou la gémination: le signe de la chadda peut être placé au-dessus de toutes les consonnes en position non initiale. La consonne qui la reçoit est alors analysée en une séquence de deux consonnes identiques (dédoublement de consonnes identiques)

En ce qui concerne la sémantique , la gémination peut changer totalement le sens des mots , exemple :

ضدَّرَ التلميذ[Haddara] qui signifie l'élève a préparé

ضدَرَ التلميذُ[Hadara] qui signifie l'élève est présent

L'Arabe classique standard a 34 phonèmes parmi lesquels 6 sont voyelles et 28 sont des consonnes [15]. Les phonèmes Arabes se distinguent par la présence de deux classes qui sont appelées pharyngales et emphatiques. Les syllabes permises dans la langue Arabe sont: [*CV*] , [*CVC*] ,[*CVCC*] .Ou le [*V*] désigne voyelle courte ou longue et le [*C*] représente une consonne [15].

1.9 Conclusion

Nous avons fait un bref tour d'horizon sur les caractéristiques de production de la parole, processus de sa génération, et aussi quelques caractéristiques de base de la langue Arabe Standard.

Les objectifs de ce chapitre sont de définir les notions que nous utiliserons dans notre travail. Cette partie théorique sera complétée dans le chapitre suivant par une étude approfondie des systèmes de reconnaissance automatique de la parole.

Chapitre 2 La Reconnaissance Automatique De La Parole

2.1 Introduction

La Reconnaissance Automatique de la Parole a pour but de permettre à un utilisateur de s'adresser oralement à une machine pour des taches diverse: commande , traduction ,... etc.

Nous abordons dans ce chapitre l'étude de l'étage frontal de tout système de reconnaissance automatique de la parole, suivi d'une brève historique ,domaines d'applications de la RAP et enfin la paramétrisation du signal vocal.

2.2 Définition

Un système de Reconnaissance Automatique de la Parole (RAP) est un système qui a la capacité de détecter la parole et de l'analyser dans le but de générer une chaîne de mots ou phonèmes représentant ce que la personne a prononcé. [1]

2.3 Historique de la reconnaissance de la parole

Le tableau (2.1) propose un historique succinct de l'évolution des systèmes de reconnaissance de la parole.

Année	Evénement
1952	Reconnaissance des dix chiffres, pour un locuteur, par un dispositif électronique câblé.
1960	Utilisation des méthodes numériques.
1965	Reconnaissance de phonèmes en parole continue.
1968	Reconnaissance de mots isolés par des systèmes simulés sur gros ordinateurs (jusqu'à 500 mots).
1969	Utilisation d'informations linguistiques.
1971	Lancement du projet ARPA aux États-Unis pour l'étude de systèmes de compréhension de la parole.
1972	Premier appareil commercialisé de reconnaissance de mots (VIP 100 de Threshold, 32 mots, monolocuteur).
1976	Fin du projet ARPA ; les systèmes opérationnels sont HARPY, HEARSAY I et II et HWIM.
1978	Commercialisation d'un système de reconnaissance à micro-processeur sur une carte de circuits imprimés (VRM d'Interstate, jusqu'à 100 mots, monolocuteur).
1981	- Utilisation de circuits intégrés VLSI (*Very Large Scale Integrated*) spécifiques du traitement de la parole (système VRC-100 d'interstate). - Système de reconnaissance de mots sur un circuit VLSI (système Weitek, vocabulaire de 8 mots, multilocuteurs)
1983	Première mondiale de commande vocale à bord d'un avion de chasse en France (Crouzet).
1985	Commercialisation des premiers systèmes de reconnaissance de plusieurs milliers de mots (Dragon, IBM, Kurzweil).
1986	lancement du projet japonais ATR de téléphone avec traduction automatique en temps réel.

Tableau 2.1 Grandes étapes de la reconnaissance de la parole.[2]

Année	Evénement
1988	Apparition des premières machines à dicter par mots isolés.
1989	Recrudescence des modèles connexionnistes neuromimétiques.
1990	Premières véritables applications de dialogue oral homme-machine.
1995	Premières applications télématiques aux USA (ATT, Nynex).
1996	Premières machines à dicter en parole continue commercialisées (IBM, Dragon, Philips)
2002	Premières introductions de la reconnaissance de la parole dans les centres d'appel.
2008	Premiers téléphones portables avec commandes vocales.

Tableau 2.1 Grandes étapes de la reconnaissance de la parole (la suite).[2]

2.4 Domaines d'application de la PAP

Nous allons citer dans ce qui suit les domaines d'application de la RAP les plus important et dans les quels des SRAP existent déjà ou ils sont en voie de construction, ces domaines sont:

- la sécurité
- le contrôle d'accès, on peut citer à titre d'exemples:
  - ✓ commander une voiture vocalement, sécurisé l'accès à une banque , ou une entreprise vocalement.
  - ✓ consultation d'un compte bancaire à distance à travers l'utilisation de téléphone.
- le domaine militaire :

- ✓ police criminelle (identification de suspects)
- ✓ filtrage de voix suspectes (avec validation humaine)
- ✓ commandes vocales en navigation aérienne

2.5 Quelques principaux objectifs de la RAP

La reconnaissance vocale a eu une énorme utilisation surtout dans les services a usage générale tel que le service Télécom (utilisation des téléphones portables) et cela pour atteindre les objectifs suivants:

- améliorer la fiabilité des systèmes tout en passant de monde de fonctionnement indépendant de locuteur vers un autre monde totalement sécurisé et fortement liée au locuteur.
- augmenter l'interactivité des systèmes Hommes-Machines , tout en intégrant le module de la reconnaissance de la voix dans les systèmes.
- rendre la phase de reconnaissance de la parole robuste surtout dans les environnements bruités.
- tester l'adaptation de la reconnaissance sur des applications réelles et avec un énorme vocabulaire.
-

2.6 Le principe général d'un système de RAP

Le principe général d'un système de RAP peut être décrit par la figure (2.1)

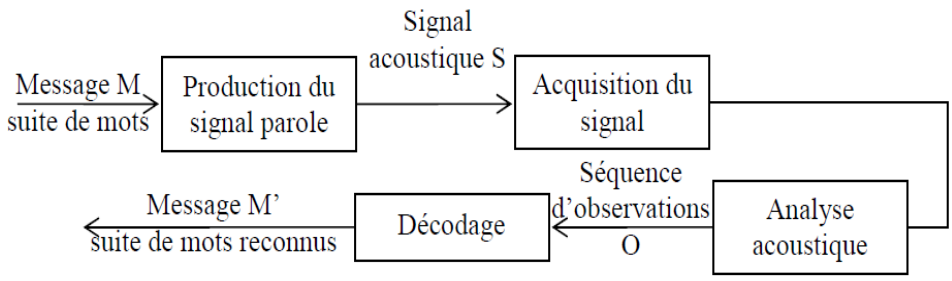


Figure 2.1 *Principe de la reconnaissance de la parole* . [abdenour]

La suite de mots prononcés M est convertie en un signal acoustique S par l'appareil phonatoire. Ensuite le signal acoustique est transformé en une séquence de vecteurs acoustiques ou d'observations O (chaque vecteur est un ensemble de paramètres acoustiques). Finalement le module de décodage consiste à associer à la séquence d'observations O une séquence de mots reconnus M'.

Un système RAP transcrit la séquence d'observations O en une séquence de mots M' en se basant sur le module d'analyse acoustique et celui de décodage.

2.6.1 L'analyse acoustique du signal vocal

a Conversion analogique numérique

La parole apparaît physiquement comme une variation de la pression de l'air causée et émise par le système articulatoire .

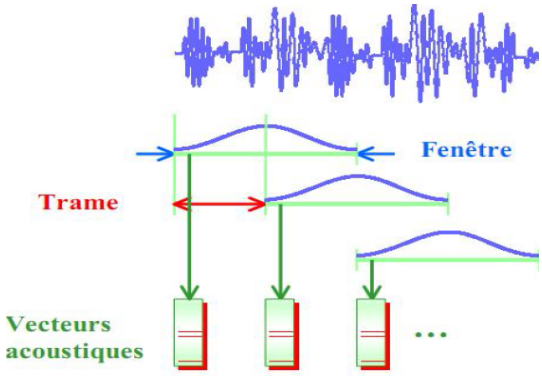


Figure 2.2 L'analyse acoustique. [trai2]

La phonétique acoustique étudie ce signal en le transformant dans un premier temps en signal électrique grâce au transducteur approprié : le microphone (lui-même associé à un préamplificateur).

De nos jours, le signal électrique résultant est le plus souvent numérisé. Il peut alors être soumis à un ensemble de traitements statistiques qui visent à en mettre en évidence les traits acoustiques : sa fréquence fondamentale, son énergie, et son spectre. Chaque trait acoustique est lui-même intimement lié à une grandeur perceptuelle : pitch, intensité, et timbre.

L'opération de numérisation, schématisée à la figure (2.3), requiert successivement : un filtrage de garde, un échantillonnage, et une quantification [3].

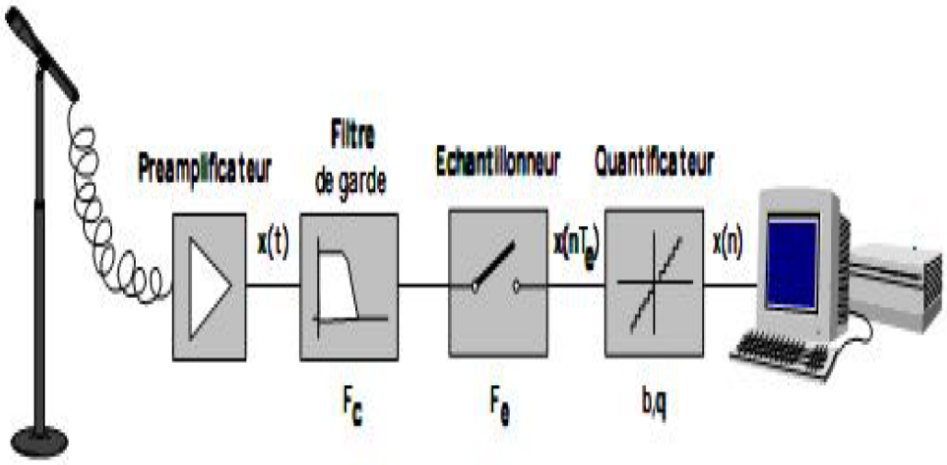


Figure 2.3 La numérisation du signal vocale.[]

- **Echantillonnage**

L'échantillonnage consiste à transformer une fonction *a*(*t*) à valeurs continues en une fonction â(*t*) discrète constituée par la suite des valeurs *a*(*t*) aux instants d'échantillonnage *t* = *kT* avec *k* un entier naturel (figure 2.4). Le choix de la fréquence d'échantillonnage n'est pas aléatoire car une petite fréquence nous donne une présentation pauvre du signal. Par contre une très grande fréquence nous donne des mêmes valeurs, redondance, de certains échantillons voisins donc il faut prélever suffisamment de valeurs pour ne pas perdre l'information contenue dans *a*(*t*). Le théorème suivant traite cette problématique :

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

$$Fech = 2 \times Fmax$$

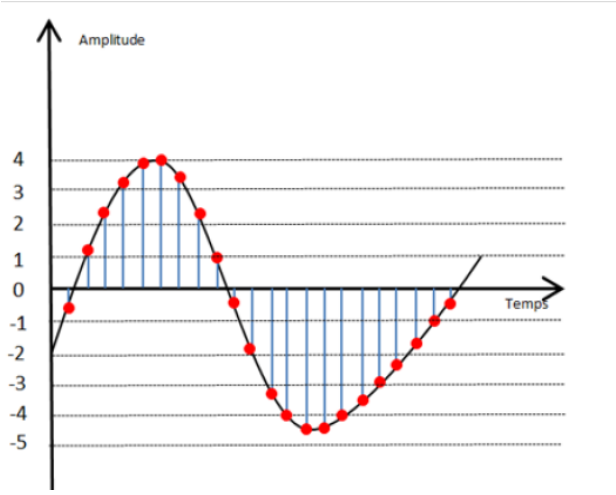


Figure 2.4 Un signal échantillonné .[]

- **Quantification**

Cette étape consiste à approximer les valeurs réelles des échantillons selon une échelle de *n* niveaux appelée échelle de quantification. Il y a donc 2^n valeurs possibles comprises entre $-2n-1$ et $2n-1$ pour les échantillons quantifiés (figure 2.5). L'erreur systématique que l'on commet en assimilant les valeurs réelles de l'écart au niveau du quantifiant le plus proche est appelé bruit de quantification.

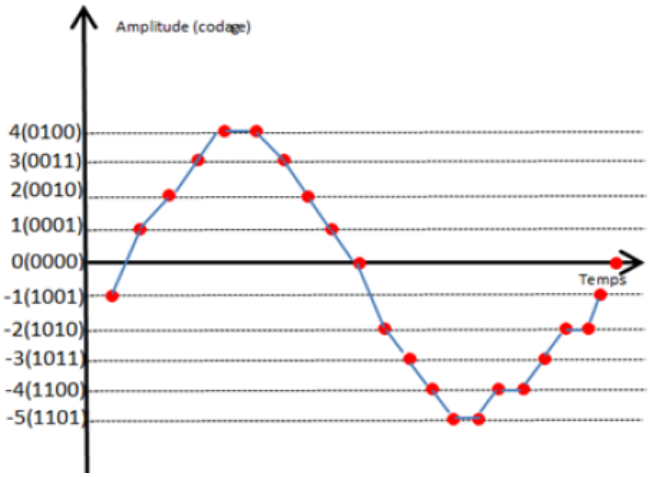


Figure 2.5 Un signal quantifié.[]

- **Codage**

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

b Paramétrisation du signal vocal

L'objectif de cette phase de reconnaissance est d'extraire des coefficients représentatifs du signale de la parole. Ces coefficients sont calculés à intervalles réguliers. En simplifiant les choses, le signal de la parole est transformé en une série de vecteurs de coefficients, ces coefficients doivent représenter au mieux ce qu'ils sont censé modéliser et doivent extraire le maximum d'informations utiles pour la reconnaissance. Parmi les coefficients les plus utilisés et qui représentent au mieux le signal de la parole, nous trouvons les coefficients ceptraux, appelés également ceptres. Dans notre travail, nous utilisons les coefficients MFCC (Mel Frequency Cepstral Coefficient).

- **Étapes de calcul du vecteur caractéristique de types MFCC**

Dans ce qui suit, nous décrivons chacune des étapes nécessaires pour l'obtention d'un vecteur caractéristique tiré des coefficients MFCC, tel qu'illustré par la figure (2.6).

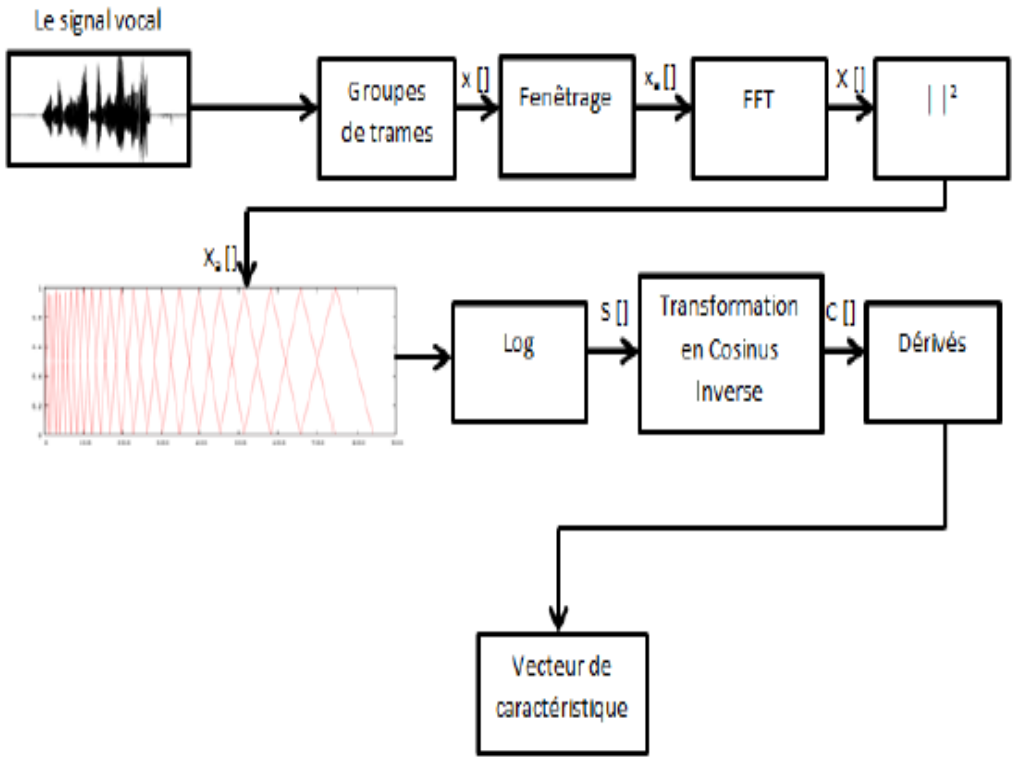


Figure 2.6 Étapes de calcul d'un vecteur caractéristique de type MFCC.[traitement]

✓ **Groupement en trames *(Frame blocking)***

Le signal acoustique continu est segmenté en trames de N échantillons, avec un pas d'avancement de M trames ($M < N$), c'est-à-dire que deux trames consécutives se chevauchent sur $N - M$ échantillons. Les valeurs couramment utilisées pour M et N sont respectivement 10 et 20. Comme prétraitement, il est d'usage de procéder à la préaccentuation du signal en appliquant l'équation de différence du premier ordre aux échantillons $x(n)$, avec l'équation (2.1).

$$x'(n) = x(n) - kx(n-1), \quad 0 < n < N-1 \tag{2.1}$$

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

✓ ***Fenêtrage***

Si nous définissons $w(n)$ comme fenêtre où $0 < n < N - 1$ et N représente le nombre d'échantillons dans chacune des trames, alors le résultat du fenêtrage est le signal xa, donné par la formule (2.2)

$$x_a = x(n)w(n), \quad 0 < n < N-1 \tag{2.2}$$

Les fenêtres les plus utilisées sont :

- Fenêtre de Hamming :(2.3)

$$w(n) = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) & 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \tag{2.3}$$

- Fenêtre rectangulaire :(2.4)

$$w(n) = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \tag{2.4}$$

- Fenêtre triangulaire :(2.5)

$$w(n) = \begin{cases} \frac{2n}{N-1} & si 0 \leq n \leq \frac{N-1}{2} \\ \frac{2(N-n-1)}{N-1} & si \frac{N-1}{2} < n \leq N-1 \\ 0 & sinon \end{cases} \quad (2.5)$$

- Fenêtre de Hann :(2.6)

$$w(n) = \begin{cases} 0.5 - 0.5\cos\frac{2\pi n}{N-1} & si 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \quad (2.6)$$

- Fenêtre de Blackman :(2.7)

$$w(n) = \begin{cases} 0.42 - 0.5\cos\frac{2\pi n}{N-1} + 0.08\cos\frac{4\pi n}{N-1} & si 0 \leq n \leq N-1 \\ 0 & sinon \end{cases} \quad (2.7)$$

La figure (2.7) illustre la forme que prennent les fonctions définies ci-dessus.

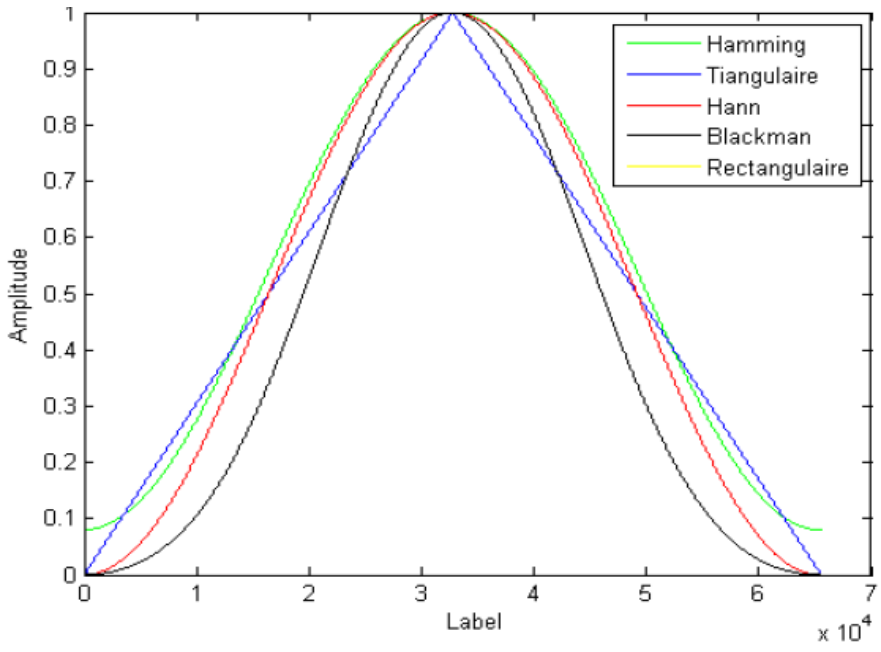


Figure 2.7 Les fonctions de fenêtrage

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

Au cours de cette étape chacune des trames, de N valeurs, est convertie du domaine temporel au domaine fréquentiel. La FFT est un algorithme rapide pour le calcul de la transformée de Fourier discret (DFT) et est définie par la formule (2.8). Les valeurs obtenues sont appelées le spectre.

$$x[k] = \sum_{n=0}^{N-1} x_a[n] e^{\frac{-2j\pi}{N} kn}, \quad 0 \leq k \leq N-1 \tag{2.8}$$

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

✓ ***Filtrage sur l'échelle Mel***

Le spectre d'amplitude est pondéré par un banc de *M* filtres triangulaires espacés selon l'échelle Mel. Dans l'échelle de mesure Mel, la correspondance est approximativement linéaire sur les fréquences au-dessous de 1*kHz* et logarithmique sur les fréquences supérieures à celle-ci. Cette relation est donnée par la formule (2.9)

$$m = 2595 \log_{10}(1 + \frac{f}{700}) \tag{2.9}$$

Le logarithme de l'énergie de chaque filtre est calculé selon l'équation (2.10)

$$S[m] = \ln[\sum_{k=0}^{N-1} X_a[k] H_m[k]], \qquad 0 < m \leq M \tag{2.10}$$

✓ ***Calcul du cepstre sur l'échelle Mel***

Le cepstre sur l'échelle de fréquence Mel est obtenu par le calcul de la transformée en cosinus discrète (équation (2.11)) du logarithme de la sortie des M filtres (reconversion du log-Mel-spectre vers le domaine temporel).

$$c[n] = \sum S[n] \cos \pi n(m - \frac{1}{2})/M, \qquad 0 \leq n < M \qquad (2.11)$$

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

✓ ***Calcul des caractéristiques dynamiques des MFCC***

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

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

où:

- c_k est le vecteur MFCC de la k^{ième} trame.

- Δ*ck* = *ck*+2 − 4*ck*−2, dérivée première des MFCCs calculée à partir des vecteurs de la k^{ième} + 2 trames et k^{ième} - 2

- ΔΔ*ck* = Δ*ck*−1 −Δ*ck*+1, seconde dérivée des MFCCs.

2.7 Conclusion

Le signal de parole est échantillonné avec une fréquence comprise entre 8 et 16 kHz. Une transformée de Fourier à court terme (algorithme FFT) est appliquée sur une fenêtre d'observation de 30 ms et ce, toutes les 10 ms [14]. Un filtrage est effectué pour mettre le spectre à l'échelle MEL. C'est une échelle perceptive qui modélise à l'aide d'un banc de filtres, la réponse en fréquence du système auditif humain. Les coefficients MFCC, peuvent alors être calculés, sont issus d'une transformation de Fourier inverse appliquée au logarithme du spectre de puissance.

Généralement, les douze premiers coefficients sont retenus, auxquels s'ajoute le logarithme de l'énergie normalisée. Treize coefficients sont ainsi obtenus qui représentent un intervalle de signal de 10 ms. La dimension du vecteur acoustique est finalement augmentée à 39 composantes en ajoutant l'approximation de la dérivée première et seconde des treize coefficients (les coefficients delta et delta-delta).

Les coefficients MFCC (Mel Frequency Cepstral Coefficient) sont les paramètres les plus utilisés dans les systèmes de RAP. Ces coefficients sont généralement utilisés avec leurs paramètres dynamiques Δ et $\Delta\Delta$ afin d'améliorer les performances de ces systèmes.

Chapitre 3 Les Modèles de Markov Cachés par l'outils HTK

3.1 Introduction

Les modèles de Markov cachés ont pris depuis une importance prépondérante en reconnaissance automatique de la parole. Ils ont été Introduits depuis les années 75.Ils sont devenus un outil incontournable de la RAP, leur emploi n'y est cependant pas limité.

En RAP, les HMM sont des modèles de production. Ils ont l'avantage d'offrir des procédures automatiques pour le décodage et l'apprentissage.
Nous nous proposons dans un premier temps de présenter les Modèles de Markov cachés MMC , notion mathématique qui nous permettra d'aborder l'application de ces derniers dans les systèmes de reconnaissance automatique de la parole .Et à la fin une présentation de l'outils HTK.

3.2 Les Modèles de Markov Cachés MMC (ou HMM)

Un **HMM** (**H**idden **M**arkov **M**odel) peut être décrit comme un automate probabiliste à états finis comportant deux processus : un processus caché et un processus d'émission. Le premier processus est dit "caché" car il est non observable et il est chargé du changement d'état, alors que par le second, la transition du modèle dans un état génère une observation.[20]

Le modèle utilisé est le modèle **HMM** gauche-droit (ou de Bakis), illustré par la figure (3.1), dans lequel on ne peut pas revenir à un état précédent et les états q2, q3 et q4 sont émetteurs alors que l'´etat initial q1 et l'´etat final q5 ne génèrent pas d'observations.[21]

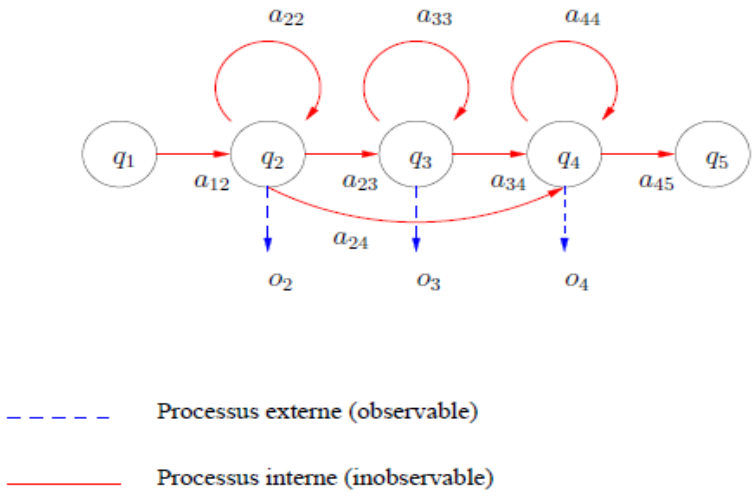


Figure 3.1 Exemple de structure à 5 états d'un HMM.[21]

Un modèle de Markov caché **HMM** est représenté par **Φ = (*A, B, л*)** qui est caractérisé par les éléments suivants :

- $S = \{s_1, s_2, \ldots, s_N\}$: un ensemble des états du modèle avec *N* le nombre d'états . On note q_t l'état à l'instant *t*.

- $O = \{o_1, o_2, \ldots, o_M\}$: un alphabet des observations avec *M* nombre fini de symboles d'observation par état. Les symboles d'observation correspondent à chaque sortie physique du système réel qu'on modélise. On note x_t l'observation à l'instant *t*.

- $A = \{aij\}$: une matrice des probabilités de transition entre états, dont *aij* est la probabilité de transition de l'état *i* à l'état *j*. On a :

$$aij = P\,(qt = sj \mid qt\text{-}1 = si\,),\ 1 \leq i, j \leq N \qquad (3.1)$$

$$\sum_{J=1}^{N} aij = 1 ; 1 \leq i \leq N \quad (3.2)$$

- *B* = {*bi*(*k*)} : une matrice des probabilités d'émission des observations dans chaque état, dont *bi*(*k*) est la probabilité d'émission de l'observation o_k dans l'état s_i . On a :

$$bi(k) = P(xt = ok \mid qt = si), 1 \leq i, j \leq N \quad (3.3)$$

- л = {p*i*} : une matrice de distribution de l'état initial. On a :

$$л = P(q0 = si\ ), 1 \leq i \leq N \quad (3.4)$$

$$\sum_{i=1}^{N} л_i = 1 \quad (3.5)$$

Exemple : Soit l'HMM suivant , paramétré par **Φ = (*A, B, л*):**

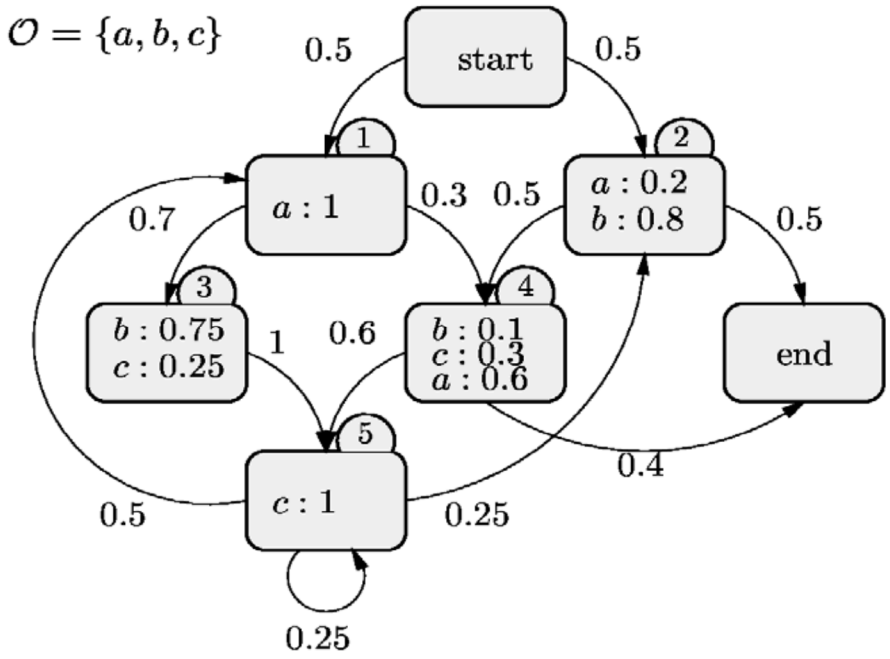


Figure 3.2 Exemple d'un HMM

- La matrice de distribution de l'état initial est :

$$л = [0.5 \;\; 0.5 \;\; 0 \;\; 0 \;\; 0]$$

- La matrice des probabilités de transition entre états est:

$$A = \begin{bmatrix} 0 & 0 & 0.7 & 0.3 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0.5 & 0.25 & 0 & 0 & 0.25 \end{bmatrix}$$

- La matrice des probabilités d'émission des observations dans chaque état est:

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0.2 & 0.8 & 0 \\ 0 & 0.75 & 0.25 \\ 0.6 & 0.1 & 0.3 \\ 0 & 0 & 1 \end{bmatrix}$$

3.3 Les problèmes fondamentaux des HMMs

Un HMM a trois problèmes qu'on doit résoudre pour arriver aux résultats attendus , ces derniers sont :

- **Evaluation :** Soient un modèle Φ et une séquence d'observations *O* = {*o*1, *o*2, …,*oT*}. Comment calculer *P* (*O* | Φ), la probabilité que la séquence des observations ait été émise par le modèle Φ ?.

- **Décodage:** Soient un modèle Φ et une séquence d'observations *O*= { *o*1,*o*2, …, *oT* }. Comment déterminer la séquence d'états cachés *Q* = {*q*0, *q*1, …, *q*T} qui a la plus forte probabilité d'avoir généré la séquence des observations ?.

- **Apprentissage :** Soient un modèle Φ et un ensemble d'observations. Comment ajuster les paramètres du modèle Φ pour maximiser la probabilité $P(O \mid \Phi)$? .

Le problème de l'***évaluation*** est résolu par l'algorithme ***Forward***. Le problème de ***décodage*** peut être résolu en utilisant l'algorithme de ***Viterbi***. Quant au problème d'***apprentissage*** du modèle, il peut être résolu par l'algorithme ***Baum-Welch*** (ou ***Forward-Backward***). [Annexe A].

3.4 Système de RAP fondé sur les modèles HMM

La méthode HMM fournit une manière de reconnaître la parole, naturelle et très fiable pour une large gamme d'applications et intègre facilement les niveaux lexical et syntaxique.

Les différentes étapes d'un système de reconnaissance de la parole fondé sur les HMM sont représentées sur la figure (3.3).

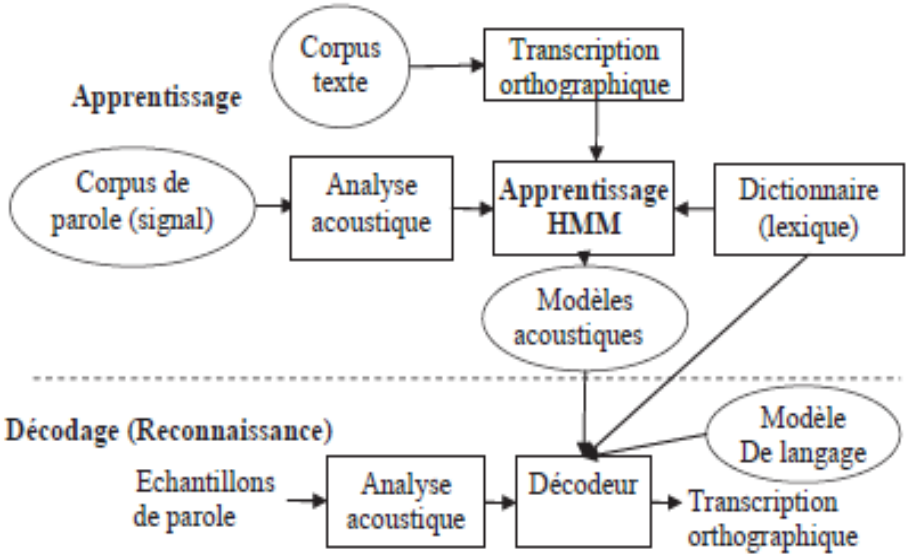


Figure 3.3 Synoptique du système de reconnaissance de la parole incluant la procédure d'apprentissage et le décodage.[1]

La ligne pointillée marque une séparation entre le processus d'apprentissage et le processus de reconnaissance. Les principaux composants utilisés pour le développement d'un tel système de reconnaissance sont les principales sources de connaissances (corpus de parole, corpus de texte, et lexiques de prononciations), le dispositif de paramétrisation acoustique (analyse acoustique), les modèles acoustiques et de langage dont les paramètres sont estimés durant la phase d'apprentissage, et le décodeur qui utilise ces modèles pour reconnaître la séquence de mots prononcés.

Les modèles acoustiques représentent les éléments à reconnaître : mots, ou unités phonétiques. Ces modèles sont usuellement développés à partir de grands corpus de données acoustiques et de textes. Ainsi, l'entraînement de ces modèles exige une définition des unités lexicales de base utilisées et un dictionnaire de prononciation décrivant la liste des mots qui pourront être reconnus.

Le modèle de langage fournit les informations syntaxiques pour la reconnaissance de la séquence de mots la plus probable.

Au centre de ce synoptique se trouve l'apprentissage par HMM qui est l'une des approches les plus utilisées dans les systèmes de RAP.

Lors de la reconnaissance, après l'analyse acoustique, un décodage est effectué et le système de reconnaissance fournit en sortie la séquence de mots la plus probable étant donné le modèle de langage et les modèles HMM.

3 .5 L'outils HTK ToolKit

3.5.1 Définition

Hidden Markov Model Toolkit (HTK) est un ensemble d'outils portable permettant la création et la manipulation de Modèles de Markov Cachés (HMM). HTK est principalement utilisé dans le domaine de la recherche de la reconnaissance vocale bien qu'il soit tout à fait utilisable dans de nombreuses autres applications telles que la

synthèse vocale, la reconnaissance de l'écriture ou la reconnaissance de séquences d'ADN (Acide désoxyribonucléique).

La plate-forme logicielle HTK (Hidden Markov toolkit) était originalement mise au point à l'université de Cambridge, dédiée au développement de systèmes à base de HMMs. Le tableau (3.1) présente quelques caractéristiques de la librairie HTK.

Caractéristiques	HTK
Organisme	Microsoft et Cambridge University
URL	http://htk.eng.cam.ac.uk/
Langage	C
Environnement	Unix, Linux, Windows
Date de la première version	1993
Disponibilité de la source	Sous licence

Tableau 3.1 Quelques caractéristiques de la librairie HTK .

Il est composé d'un ensemble de modules et outils écrits en langage C. Ces différents outils facilitent l'analyse vocale, l'apprentissage des HMM, la réalisation de tests et l'analyse des résultats. Il est à noter, que ce qui a contribué au succès de HTK, est qu'il est accompagné d'une bonne documentation.

Principalement, la boîte à outils HTK est utilisée pour la construction des systèmes RAP basés sur les modèles HMM dans un but de recherche scientifique. Généralement les deux processus indispensables pour le fonctionnement d'un RAP sont le processus d'apprentissage et celui de reconnaissance (ou décodage). La figure

(3.4) illustre l'enchaînement de ces processus. Premièrement, les outils d'apprentissage HTK sont utilisés pour estimer les paramètres de l'ensemble des modèles HMM en utilisant des signaux de parole ainsi que leurs transcriptions associées. Ensuite, les signaux de parole inconnue sont transcrits en utilisant les outils de reconnaissance.

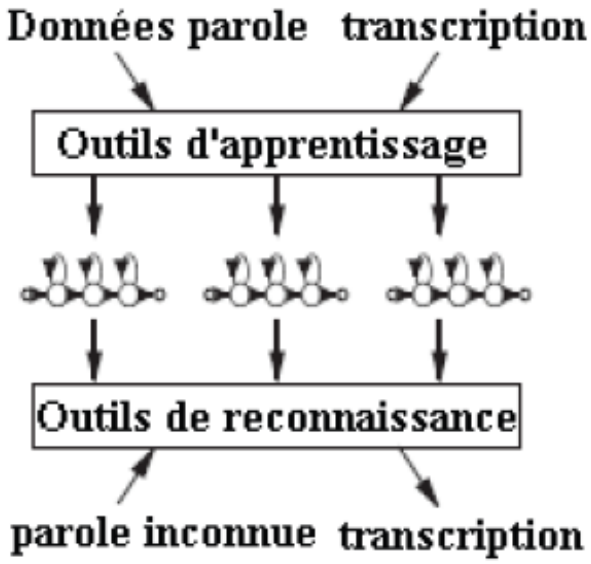


Figure 3.4 Processus d'un système RAP sous HTK.[]

3.5.2 Principe de fonctionnement

Pratiquement , La construction d'un système RAP se base sur quatre (04) phases principales: préparation des données , apprentissage, test et analyse. La figure (3.5) illustre les différents outils HTK de chaque phase .

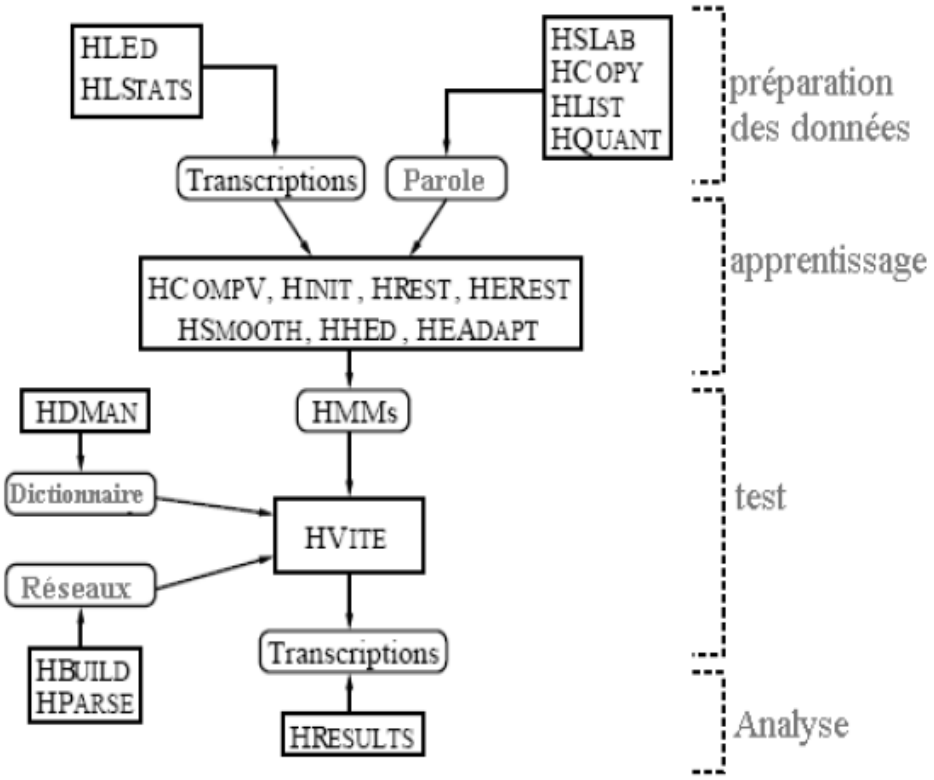


Figure 3.5 Les différentes phases du système RAP sous HTK et outils associés.[24]

a Outils de préparation des données

La construction d'un ensemble de modèles HMM exige un ensemble de fichiers de données de parole (signaux), ainsi que leurs transcriptions correspondantes. Souvent les données de parole sont récupérées à partir d'une base de données.

Cette base doit être répartie en un corpus d'apprentissage et un corpus de test. Chacun de ces corpus contient un ensemble de fichiers texte contenant la transcription orthographique des phrases et un ensemble de fichiers de données contenant les échantillons des signaux correspondant aux fichiers texte. Avant d'être utilisées dans l'apprentissage, ces données doivent être converties en un format paramétrique approprié et ses transcriptions associées doivent être converties en format correct (étiquetées en label de mot).

Si les données de parole ne sont pas disponibles, alors l'outil **HSLab** peut être utilisé pour enregistrer la parole et l'étiqueter manuellement par n'importe quelle transcription (par phonème ou mot). Ainsi pour chaque phrase prononcée, on lui correspond un fichier signal (exemple d'extensions : wav, sig,…) et un fichier de transcription (extension lab).

La dernière étape dans la phase de préparation des données est la conversion du signal de chaque phrase en une séquence de vecteurs acoustiques (figure 3.4). Cette conversion est effectuée par une analyse acoustique en utilisant l'outil **HCopy**. Différents types de paramètres acoustiques sont supportés par cet outil comme : MFCC (Mel Frequency Cepstral Coefficients).

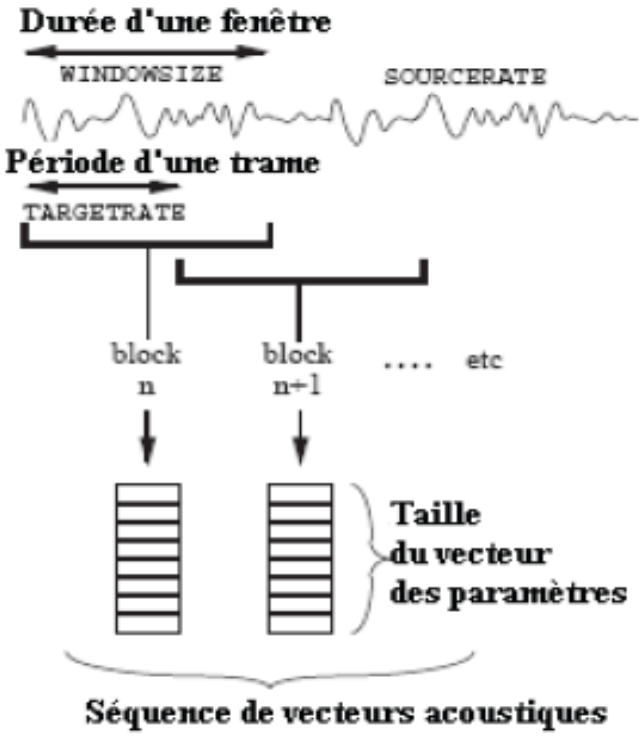


Figure 3.6 Processus de l'analyse acoustique.[23]

La ligne de commande pour l'exécution de **HCopy** s'écrit comme suit :

HCopy -T 1 -C config -S codetr.scp

La figure (3.4) montre le principe de fonctionnement de cet outil pour la conversion d'un ensemble de fichiers parole d'extension wav en un ensemble de fichiers d'extension mfc contenant des vecteurs de paramètres acoustiques MFCC. La liste de l'ensemble de ces fichiers est donnée dans un fichier appelé **codetr.dcp** dont un extrait est fourni :

```
root/training/corpus/sig/S0001.wav      root/training/corpus/mfcc/S0001.mfc
root/training/corpus/sig/S0002.wav      root/training/corpus/mfcc/S0002.mfc
root/training/corpus/sig/S0003.wav      root/training/corpus/mfcc/S0003.mfc
...etc
```

Figure 3.7 Le contenu de codetr.dcp .[23]

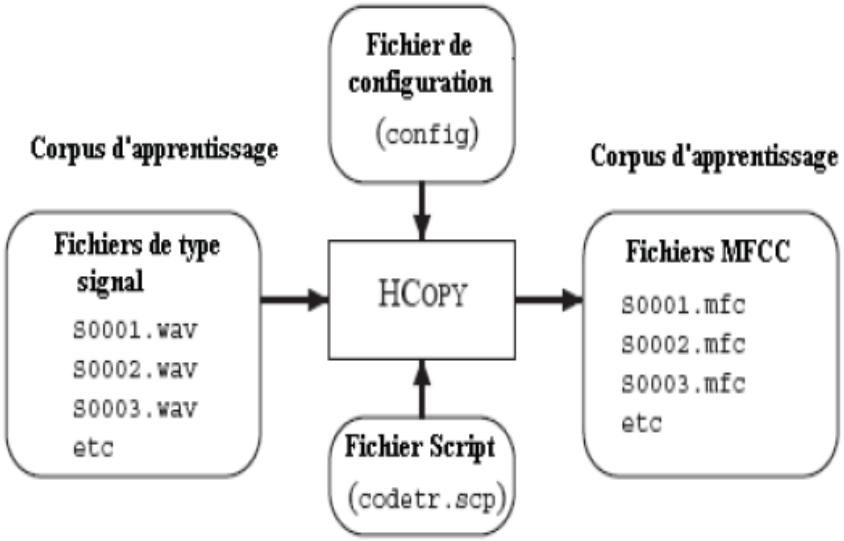


Figure 3.8 Principe de fonctionnement de l'outil **HCopy.[**23]

Cependant l'exécution de l'outil **HCopy** exige un fichier de configuration (**config**) pour définir les différents paramètres de l'analyse acoustique considérée. Voici un exemple de ce type de fichier associé à une analyse acoustique MFCC :

```
# Exemple d'un fichier de configuration pour une analyse acoustique MFCC
SOURCEFORMAT = HTK # donne le format des fichiers des signaux
TARGETKIND = MFCC_0_D_A # identificateur des coefficients à utiliser
# Unit = 0.1 micro-second :
WINDOWSIZE = 250000.0 # = 25 ms = longueur de la durée d'une trame
TARGETRATE = 100000.0 # = 10 ms = période des trames
NUMCEPS = 12 # nombre des coefficients MFCC(ici de c1 to c12)
USEHAMMING = T # utilisation de la fonction de Hamming pour le fenêtrage
des trames
PREEMCOEF = 0.97 # coefficient de prè-accentuation
NUMCHANS = 26 # nombre de canaux des bancs de filtres
CEPLIFTER = 22 # longueur de liftrage cepstral
# la fin
```

Figure 3.9 Exemple d'un fichier configuration.[23]

b Outils d'apprentissage

La deuxième phase consiste à construire les modèles HMM des mots appartenant au dictionnaire de la tâche considérée. Premièrement , pour chaque mot , il faut définir un modèle prototype contenant la topologie choisie à savoir le nombre d'états du modèle, la disposition de transitions entre les états , le type de la loi de probabilité associée à chaque état. L'état initial et final de chaque modèle n'émettent pas des observations mais servent seulement à la connexion des modèles dans la parole continue.

Les probabilités d'émissions associées aux états sont des mélanges de gaussiennes multi-variées dont les composantes sont les probabilités a priori définies chacune par une matrice de covariance et un vecteur de moyennes dans l'espace des paramètres acoustiques. La matrice de covariance peut être choisie diagonale si l'on suppose l'indépendance entre les composantes des vecteurs acoustiques .

Ces modèles prototypes sont générés dans le but de définir la topologie globale des modèles HMM. Ainsi, l'estimation de l'ensemble des paramètres de chaque modèle HMM est le rôle du processus d'apprentissage.

Les différents outils d'apprentissage sont illustrés dans la figure (3.10). Selon cette figure, deux chaines de traitement peuvent être envisagées pour l'initialisation des modèles HMM. La première chaîne tient en compte des signaux étiquetés en label de mot. Dans ce cas, l'outil **HInit** extrait tous les segments correspondant au mot modélisé et initialise les probabilités d'émission des états du modèle au moyen de la procédure itérative des "k moyennes segmentales". Ensuite l'estimation des paramètres d'un modèle est affinée avec **HRest**, qui applique l'algorithme optimal de Baum-Welch jusqu'à la convergence et réestime les probabilités d'émission et de transition.

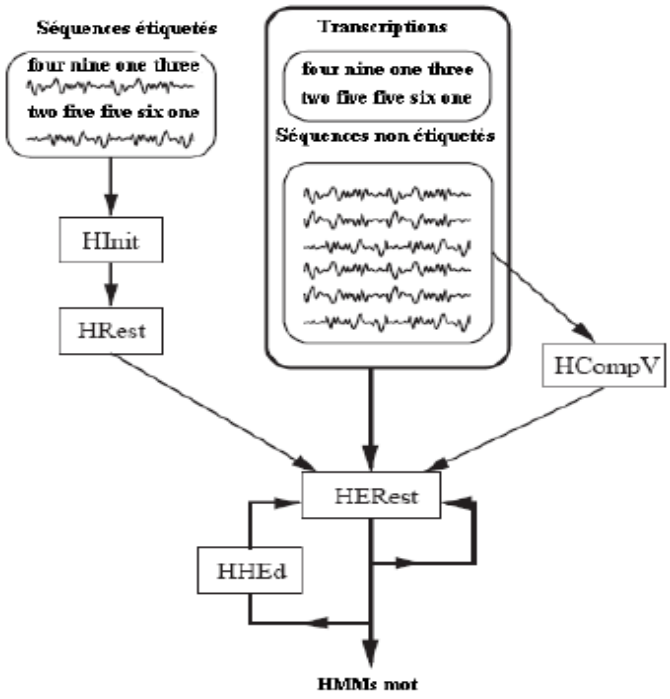


Figure 3.10 Outils d'apprentissage HTK.[23]

Dans la deuxième chaîne, les signaux ne sont pas étiquetés. Dans ce cas, tous les modèles HMM sont initialisés avec le même modèle dont les moyennes et les variances sont égales respectivement à la moyenne et la variance globales de tous les vecteurs acoustiques du corpus d'apprentissage. Cette opération est effectuée par l'outil **HCompV**.

Après l'initialisation des modèles, l'outil **HERest** est appliqué en plusieurs itérations pour ré-estimer simultanément l'ensemble des modèles sur l'ensemble de toutes les séquences de vecteurs acoustiques non étiquetés. Les modèles obtenus peuvent être améliorés, en augmentant par exemple le nombre de gaussiennes servant à estimer la probabilité d'émission d'une observation dans un état. Cette augmentation est effectuée par l'outil **HHEd**. Les modèles doivent être ensuite ré-estimés par **HRest** ou **HERest**.

c Outils de reconnaissance

La boîte HTK fournit un outil de reconnaissance appelé **HVite** qui permet la transcription d'une séquence de vecteurs acoustiques en une séquence de mots. Le processus de reconnaissance est illustré dans la figure (III.7).

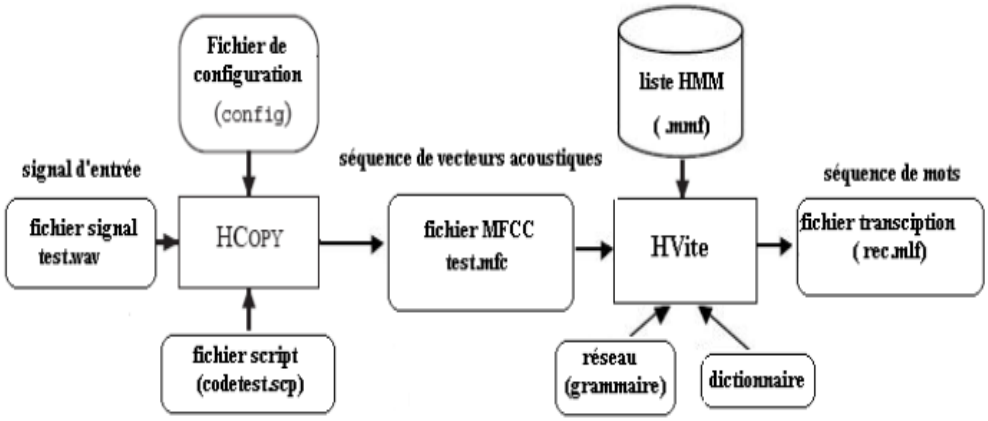


Figure 3.11 Processus de reconnaissance sous HTK.[23]

HVite utilise l'algorithme de Viterbi pour trouver la séquence d'états la plus probable qui génère la séquence d'observations (vecteurs acoustiques) selon un modèle HMM composite, ceci afin d'en déduire les mots correspondants. Le modèle composite permet la succession des modèles acoustiques en fonction du réseau de mots qui définit la grammaire de la tâche considérée.

Le résultat de décodage par l'outil **HVite** est enregistré dans un fichier d'extension (.mlf) contenant l'étiquetage en mots du signal d'entrée .

d Outils D'évaluation

Généralement les performances des systèmes sont évaluées sur un corpus de test contenant un ensemble de fichiers d'échantillons parole ainsi que leurs fichiers d'étiquetage associés. Les résultats du corpus de test sont comparés aux étiquettes de référence par un alignement dynamique réalisé par **HResults**, afin de compter les étiquettes identifiées, omises, substituées par une autre, et insérées. Ces statistiques permettent de calculer le taux ou la précision .

3.6 Conclusion

A travers ce chapitre , nous avons décrits les modèles de Markov Cachés (MMC), la reconnaissance automatique de la parole fondé sur les modèles HHM et nous avons représenter l'outil HTK. Dans le chapitre qui suit nous entament notre coté pratique.

Chapitre 4 Résultats et Interprétations

4.1 Introduction

Ce dernier chapitre décrit la construction d'un dispositif de reconnaissance vocale pour une application simple d'une calculatrice vocale à l'aide HTK. Ce système de reconnaissance sera conçu pour reconnaître les chaînes des chiffres et des opérations composés.

4.2 Corpus utilisé

Dans notre travail, nous avons choisi la base de données BDWAVE construite en Arabie Saoudite. Cette base de données paroles est composée de mots suivants: **cifr , waahid, ithnaan , thalaatha , arbaaa, khamsa, sita, sabaa,thamaania, tisaa, faacila, tossaawii, fii, zaaid, kisma, naakis** . Ces derniers ont été prononcées en Arabe par 15 locuteurs différents. Seul le sous-corpus contenant les mots précédents a été retenu.

Ce sous-corpus a été divisé en deux sous-ensembles : corpus d'apprentissage et de test.

- Corpus d'apprentissage : Il contient 1600 mots prononcés par 10 locuteurs.
- Corpus de test : Il est composé de 800 chiffres prononcés par 05 locuteurs.

Les locuteurs des corpus d'apprentissage et de test sont différents. Les tests sont donc réalisés en mode indépendant du locuteur.

4.3 Description du système

La réalisation de ce système se déroule en 4 étapes :

- la phase de la préparation des données .
- la phase d'apprentissage .
- la phase de reconnaissance.

- la phase évaluation (de test) de la performance du système.

4.3.1 La phase de la préparation des données

La première étape de tout projet de développement de reconnaissance est la préparation des données. Dans le système à construire ici. Nous avons utilisé la base de données BDWAVE qui se compose de type de fichiers (fichiers.wav) .Nous avons défini la grammaire et nous avons construit le dictionnaire nécessaire à cette fin.

a La tache grammaire

L'objectif du système est de construite une grammaire pour les mots cités précédemment . On a utilisé pour cette fin un éditeur de texte, la grammaire de notre langage est définie dans le fichier **gram.txt.**

Cette grammaire va définir la structure de chaque phrase utilisée dans l'apprentissage, et puisque toutes les phrases se composent par des chiffres , des opérations arithmétiques et de symbole égal , on a utilisé comme symbole $WORD. Où les barres verticales « | » représentent des solutions de rechange, les deux { } signifient une ou plusieurs répétitions.

Cette grammaire peut être décrite sous la forme d'un réseau, comme il est montré dans la figure suivante :

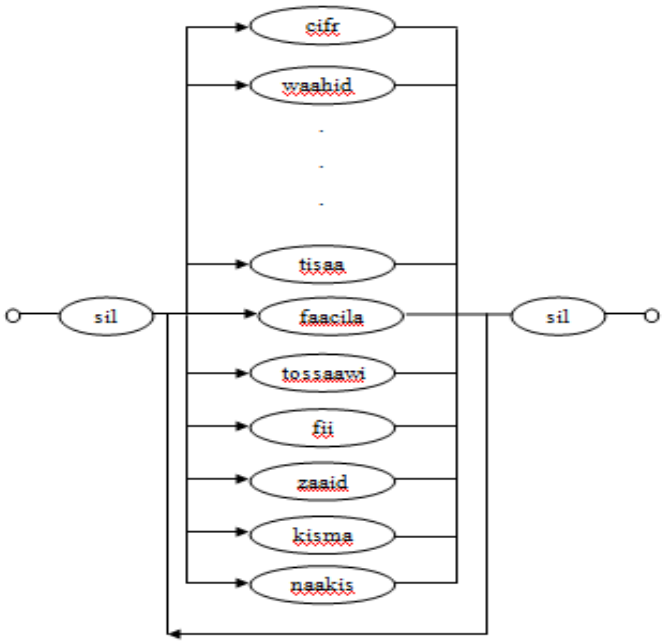


Figure 4.1 Réseau de la grammaire

HTK exige en fait un réseau de mot à définir à l'aide d'une notation appelé HTK Standard Lattice Format (SLF) dans lequel chaque mot, et chaque transition mot-à-mot est explicitement mentionné. Ce réseau de mots peut être créé automatiquement à partir de la grammaire ci-dessus à l'aide de l'outil **HParse**, en supposant que le fichier **gram.txt** contient la grammaire ci-dessus.

Il se génère le modèle de mot « **net.slf »** comme le montre la figure suivante :

Un fichier SLF contient une liste de nœuds représentant des mots et une liste d'arcs représentant les transitions entre les mots.

Chaque définition de nœud et d'arc qui est écrit sur une seule ligne, se compose d'un certain nombre de domaines. Chaque spécification de champ est constituée d'une paire "nom = valeur" comme il est illustré dans la figure précédente.

b Le dictionnaire

Dans notre cas, on a créé une liste de mots nécessaires à la main.

Puisque l'unité du traitement est le mot, la prononciation de chaque mot, entre crochets, est la même. On a ajouté une petite pause « sp » après chaque prononciation, le silence « sil » au début et à la fin de chaque mot est, comme sa prononciation l'indique, un symbole de sortie nulle (un espace entre crochets).

- Test d'un Réseau de mots à l'aide **HSGen**

Lors de la conception de la grammaire, il est utile d'être en mesure de vérifier que le langage défini par le réseau de mots est envisagé. HTK fournit un outil très simple appelé HSGen pour cet effet.

c Organisation de l'espace de travail

Dans notre cas, on a pris les fichiers wave des données pour l'apprentissage et le test, et on a organisé notre espace de travail.

On a créé la hiérarchie de répertoires suivante :

- data / train/ locuteur i /wav /: Emmagasine les fichiers wave d'apprentissage.

- data / train / locuteur i / lab /: Emmagasine les étiquettes des fichiers wave d'apprentissage après l'étape d'étiquetage.

- data / test / locuteur i /wav /: Emmagasine les fichiers wave de test.

- data / test / locuteur i / lab /: Emmagasine les étiquettes des fichiers wave de test après l'étape d'étiquetage.

- data / train / locuteur i /mfc/ : Emmagasine les fichiers de coefficients mfcc calculés.

- model / hmm0flat / : Emmagasine les modèles HMM initiaux générés avec la commande HCompv.

- model / proto / : Emmagasine le prototype de chaque modèle HMM.

- model/hmm i : Emmagasine les estimations de chaque mot (chiffre) du langage avec la commande HERest.

Les signaux vocaux (wave) sont échantillonnés à 11025 kHz, cette fréquence est suffisante pour prendre en compte la variabilité du signal de parole. Le corpus d'apprentissage est constitué de 10 locuteurs dont chaque locuteur a prononcé chaque mot 10 fois tandis que le corpus de test est constitué de 5 locuteurs .

d Création des fichiers de transcriptions

Pour créer un ensemble de HMM, tous les fichiers de données d'apprentissage doivent avoir les transcriptions de mots associés. Pour ce faire, deux séries de transcriptions seront nécessaires.

L'ensemble utilisé initialement n'aura pas de court-pause (sp) entre les mots. Puis, une fois les modèles de mots ont été générés, un modèle (sp) sera inséré entre les mots pour prendre soin de toute pause introduite par le locuteur.

- **Etiquetage manuel des données**

L'outil HTK met à disposition une fonction intitulée **HSLAB** qui permet de visualiser un fichier audio dans une interface graphique pour ensuite étiqueter les zones significatives en sélectionnant leurs parties associés.

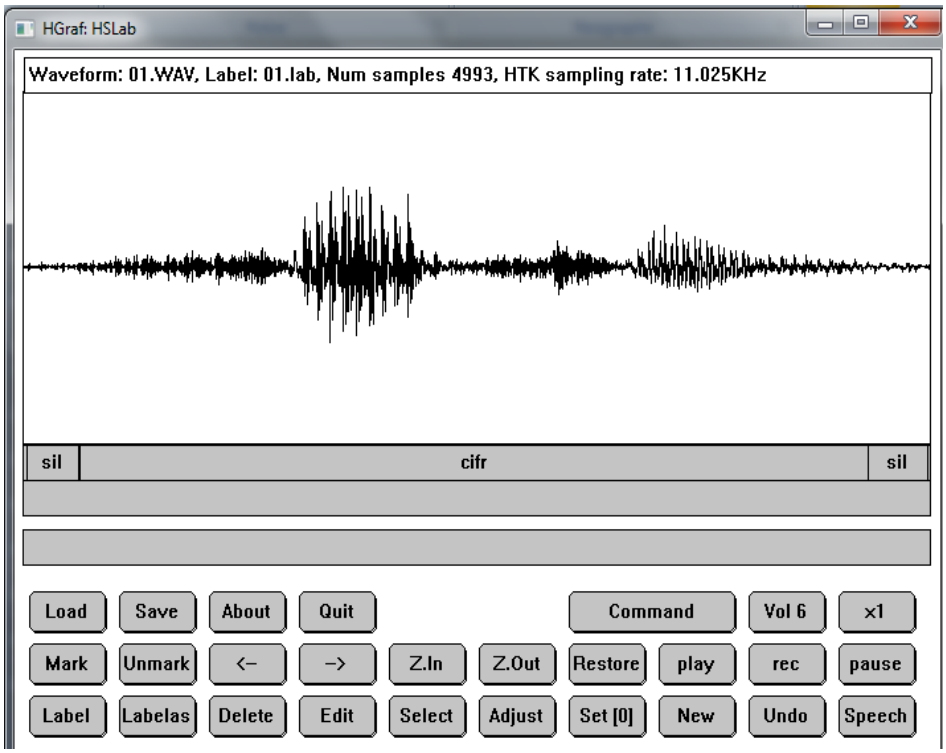


Figure 4.2 Etiquetage de mot « cifr » pour le locuteur 1

Le mot	Prononciation en Arabe	Etiquetage des mots
0	صفر	cifr
1	وَاحـد	waahid
2	اثنا ن	ithnaan
3	ثــلا ثة	thalaatha
4	أربعة	arbaaa
5	خمسة	khamsa
6	ستة	sita
7	سبعة	sabaa

8	ثمـــانية	thamaania
9	تسعة	tisaa
,	فـــاصلة	faacila
=	تـــساوي	tossaawii
x	فـــــي	fii
+	زائـــد	zaaid
/	قســمة	kisma
-	نـــاقص	naakis

Tableau 4.1 Etiquetage des mots de vocabulaire

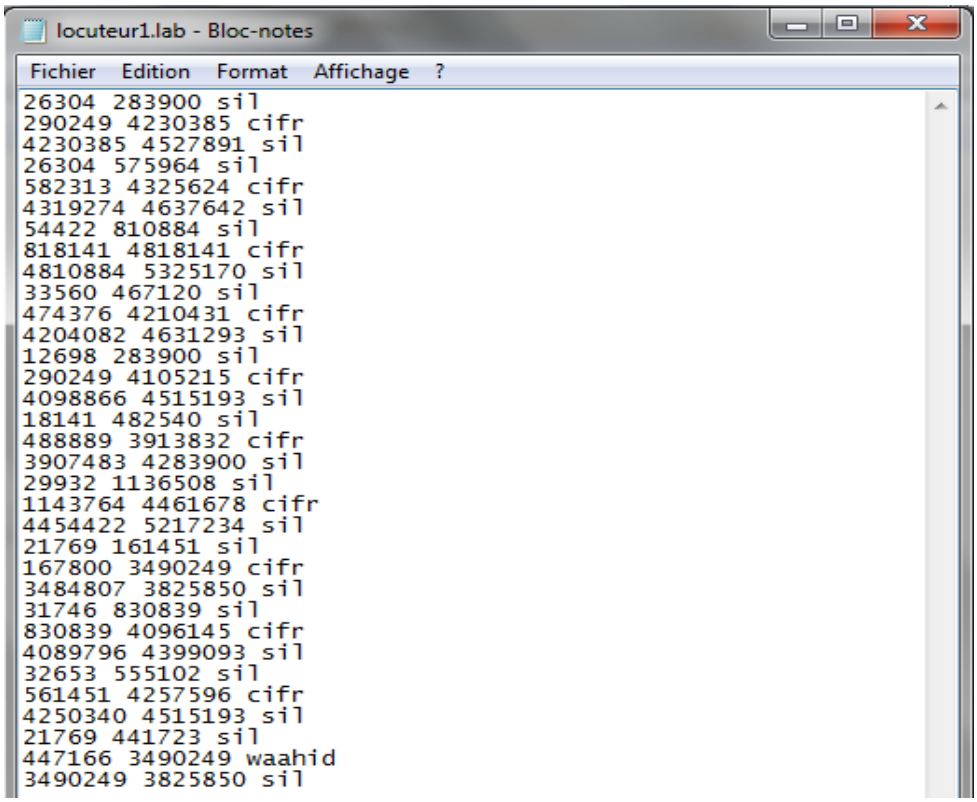


Figure 4.3 Le fichier locuteur1.lab transcrit

e Paramétrisation des données

Reconnaître la parole nécessite l'extraction du signal acoustique un ensemble de paramètres pertinents de ce signal variant au cours du temps. Cette paramétrisation est effectuée sur des trames successives de faible durée (10 ms) sur lesquelles le signal peut être considéré stationnaire. Pour améliorer l'analyse et limiter les effets de bord, les trames sont pondérées par une fenêtre temporelle de façon à réduire des discontinuités dans le signal, les fenêtres successives se recouvrent en partie. La fenêtre la plus utilisée en reconnaissance de la parole est la fenêtre de Hamming.

Après avoir préparé les fichiers son, nous avons construit une représentation acoustique du signal en appelant l'outil **Hcopy** pour générer les fichiers contenants les coefficients cepstraux.

Les coefficients cepstraux (MFCC Mel Frequency Cepstral Coefficients) sont les meilleurs pour modéliser les signaux de parole. La description de ces coefficients est bien détaillée dans l'annexe A.

La paramétrisation s'effectue en utilisant l'outil **HCopy** configuré pour convertir automatiquement les signaux son en séquences des vecteurs MFCC.

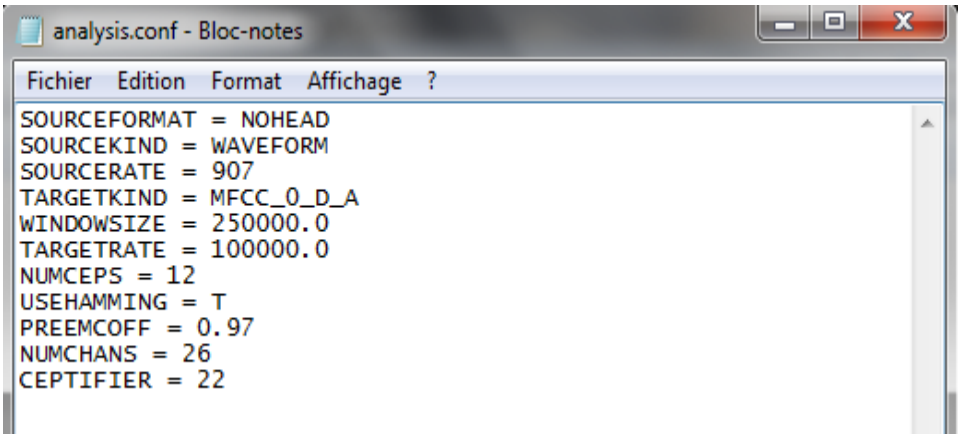


Figure 4.4 Le fichier de configuration analysis.conf

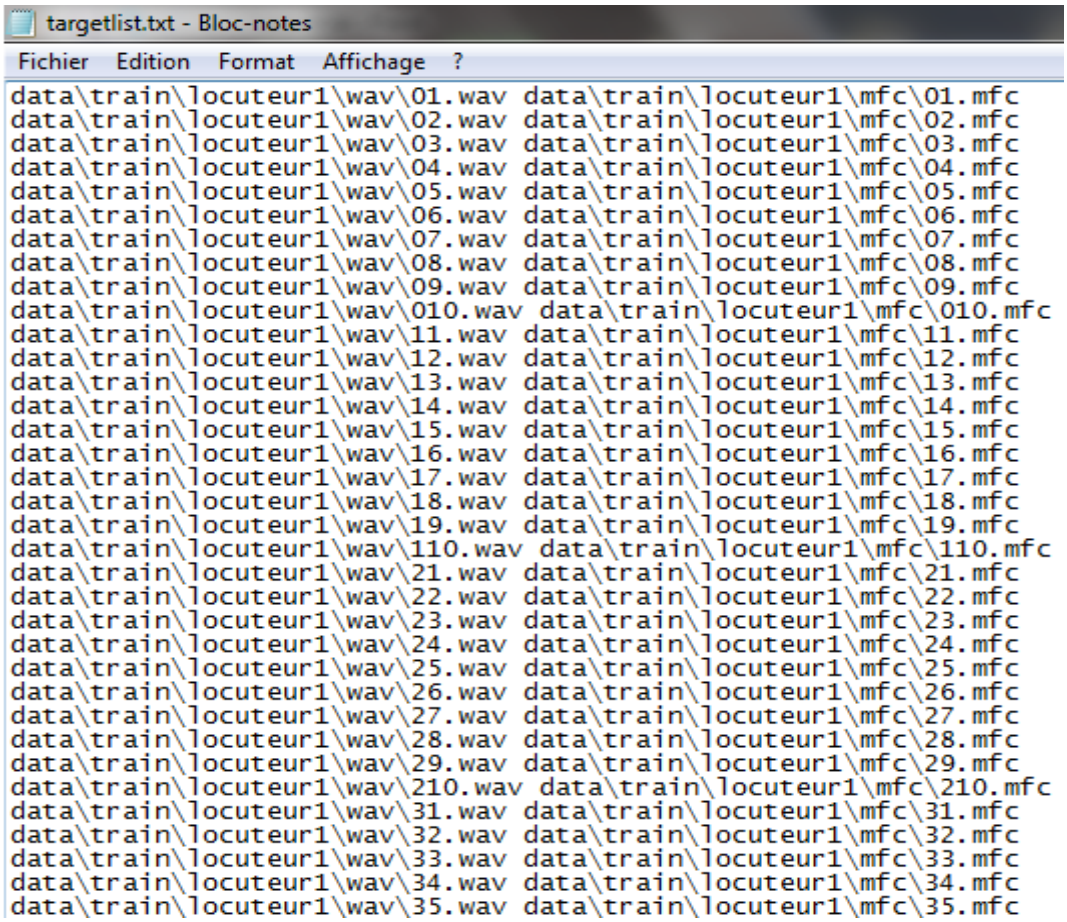


Figure 4.5 Le fichier targetlist.txt

La première colonne indique les répertoires de chaque fichier son (.wav) et la deuxième indique les répertoires où sont stockés les fichiers résultants (.mfc).

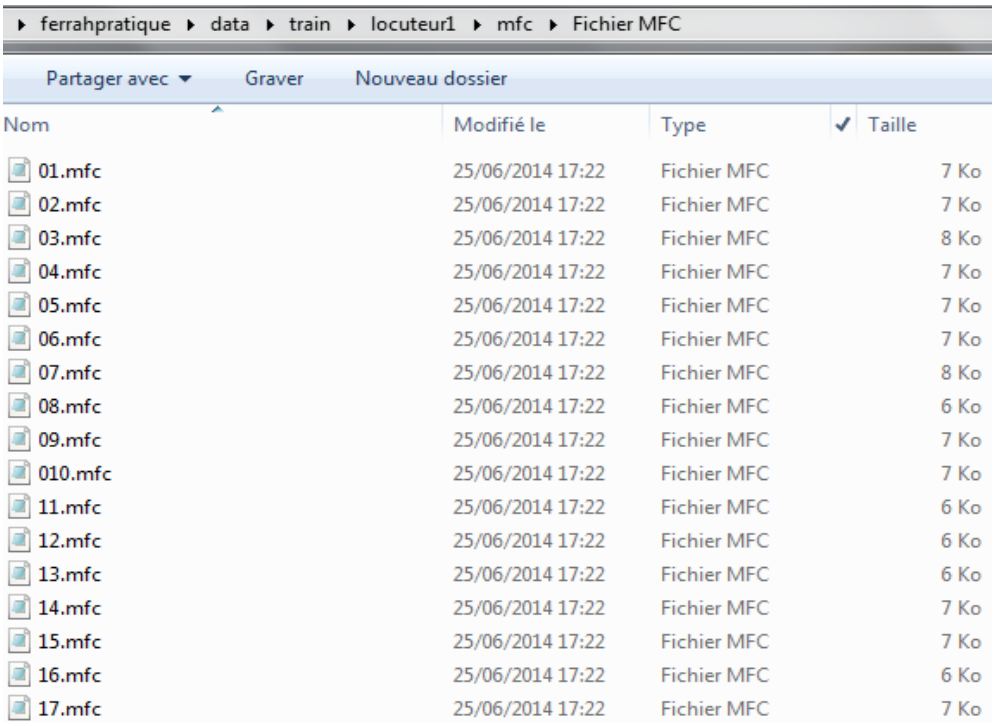


Figure 4.9 Les fichiers MFCC résultants.

4.3.2 Phase d'apprentissage

Pour la phase d'apprentissage , en pratique, la démarche de travail est la suivante:

- Création de modèles HMMs en utilisant l'outil HCompv de HTK.
- Apprentissage à l'aide de l'algorithme de Baum-Welch en utilisant l'outil HERest de HTK.

a Création du monophone initial

La première étape dans l'apprentissage des HMM est de définir un modèle de prototype appelé « proto ». Le but est de définir la topologie du modèle. Dans notre travail, la topologie utilisée est de 5 états .

Tous les prototypes des HMM sont stockés dans le répertoire model/proto/. Le fichier hmmsdef.mmf est un fichier qui contient l'ensemble de tous les prototypes.

Chaque fichier de prototype contient les informations suivantes:

<VecSize>

~h "proto"

<NumStates>

<state >

<TransP>

<BeginHMM> <EndHMM>

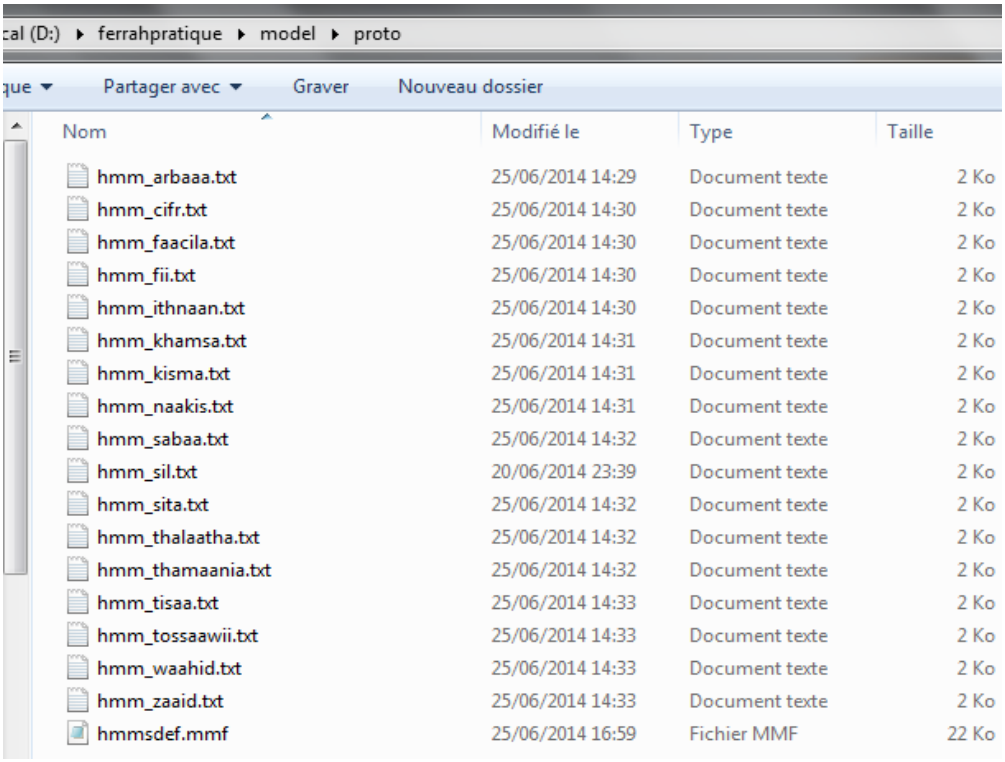


Figure 4.10 L'ensemble des prototypes HMM.

Les figures 4.11 et 4.12 nous montrent un exemple d'un prototype de mot « cifr » et celui du « sil ».

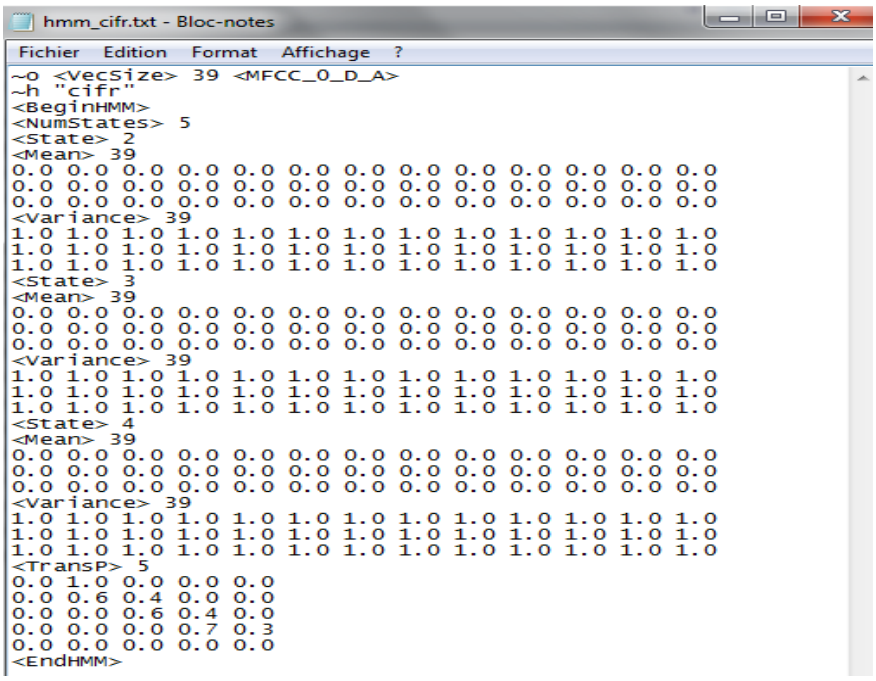


Figure 4.11 Prototype hmm_cifr

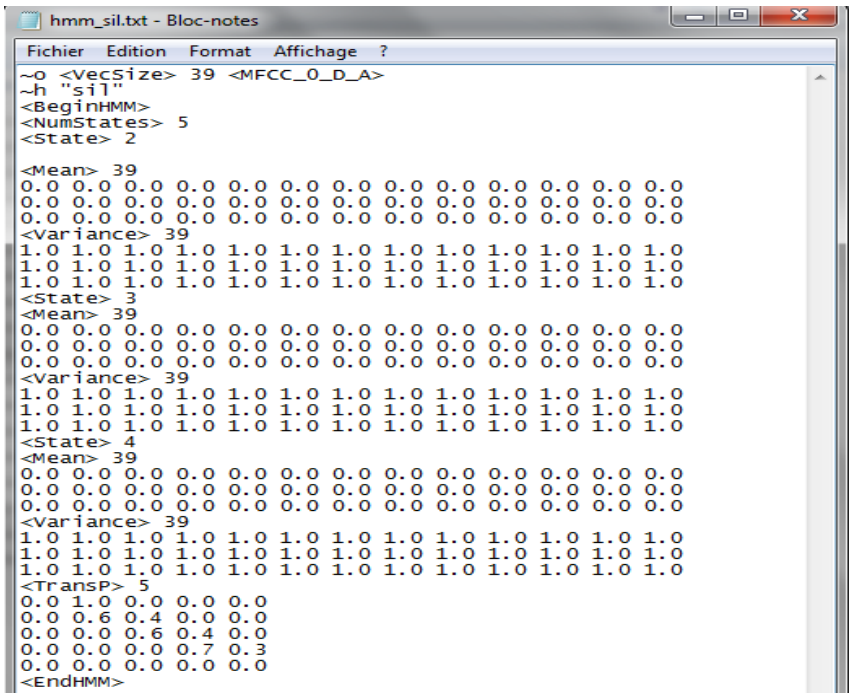


Figure 4.12 Prototype hmm_sil

HTK fournit des outils de base pour l'estimation des paramètres: HCompV et HERest. HCompV est utilisée pour l'initialisation pour fixer la moyenne et la variance de chaque composante gaussienne dans une définition de HMM pour être égale à la moyenne et la variance globale des données d'apprentissage de la parole.

L'outil **HCompv** va créer une nouvelle version du proto dans le répertoire **model/hmm0flat**.

Le fichier **hmmdefs .mmf** contenant une copie pour chacune des HMM requis est construit en copiant manuellement le prototype et le ré-étiqueter pour chaque monophone nécessaire (y compris le "sil").

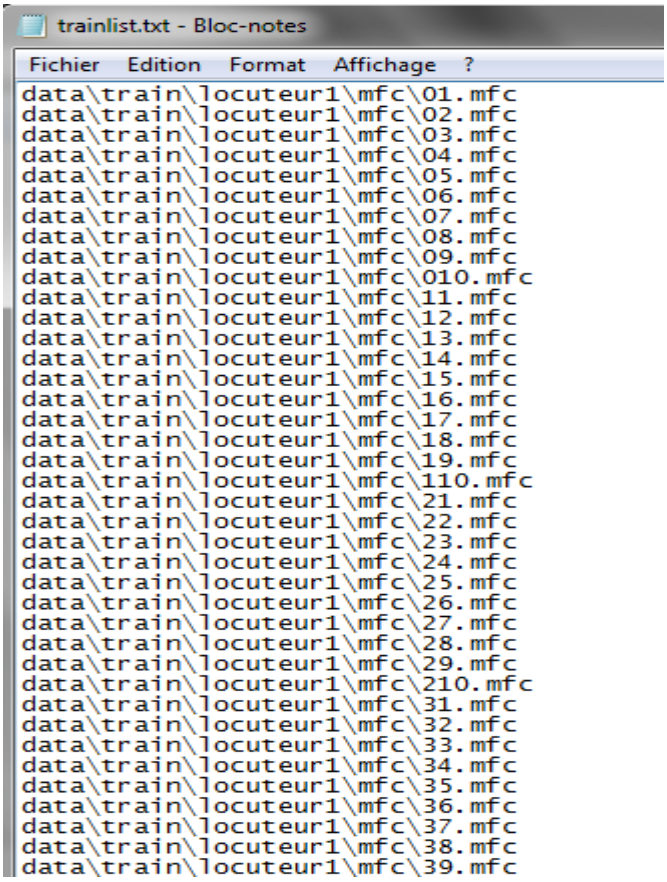


Figure 4.13 Le fichier trainlist.txt

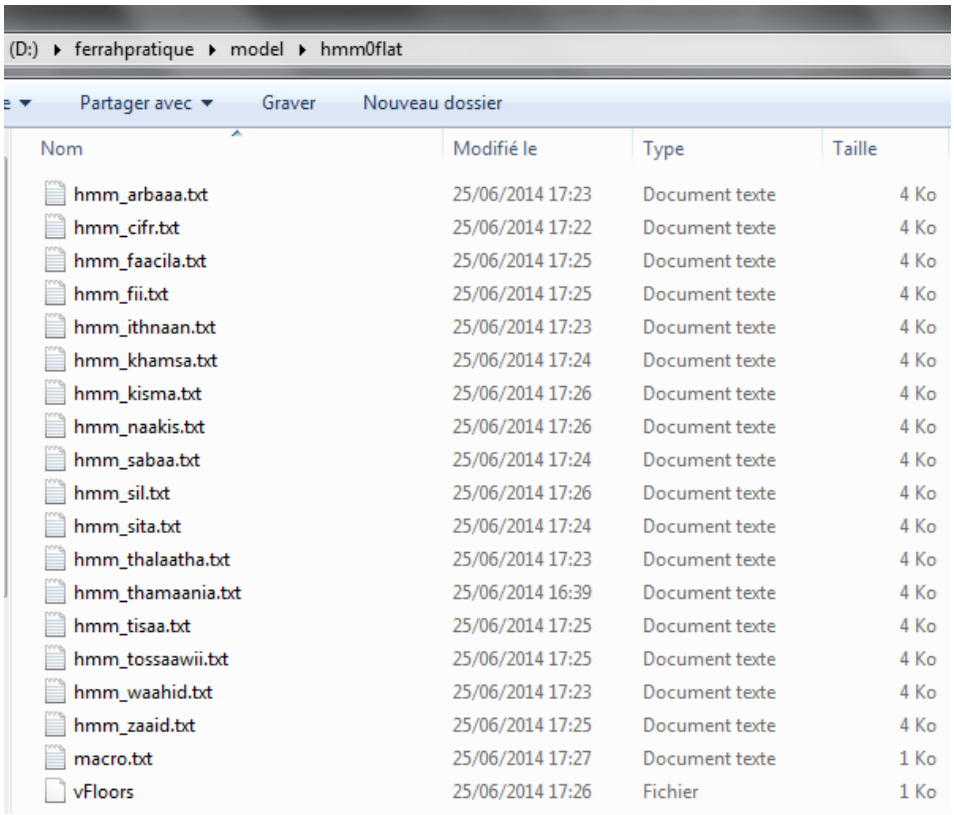


Figure 4.14 Les modèles HMM initiaux

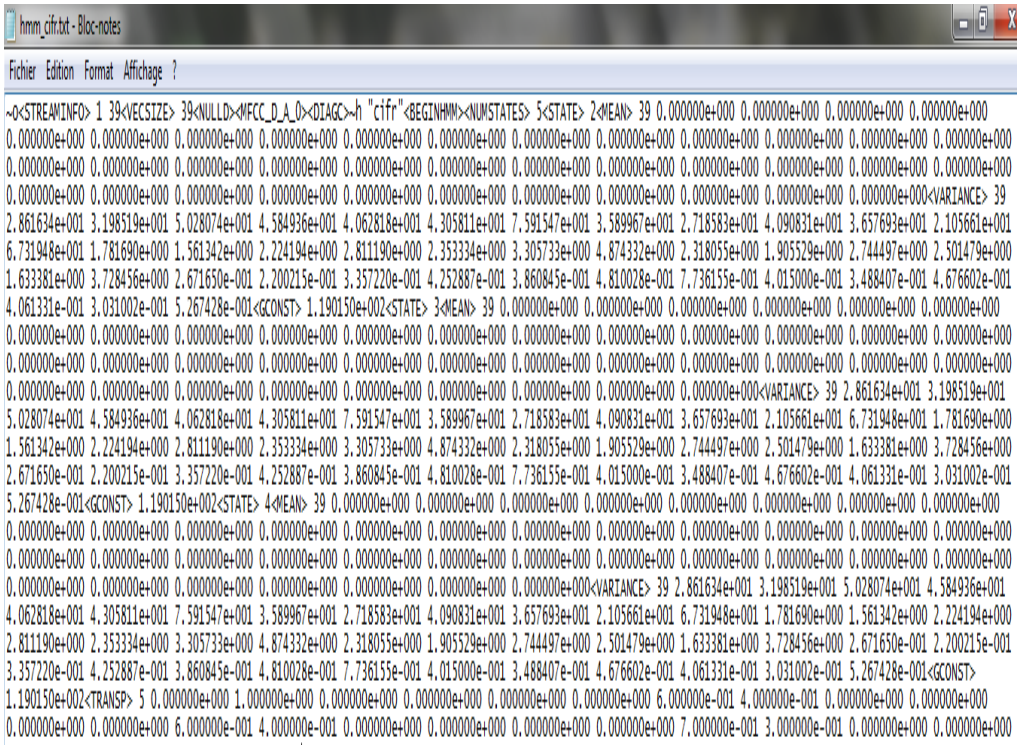


Figure 4.15 hmm_cifr.txt initial

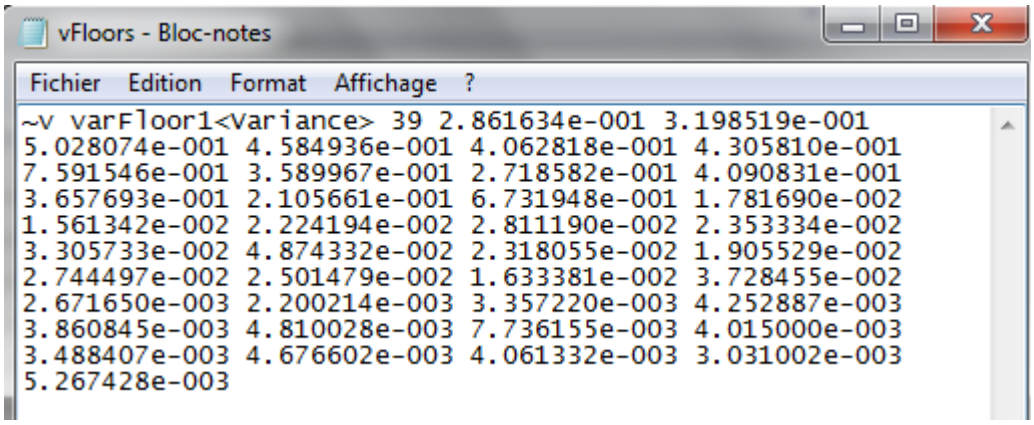


Figure 4.16 Le fichier vFloors

b Ré-estimation du modèle HMM à l'aide de l'outil HERest (Apprentissage avec l'algorithme de Baum-Welch)

Après avoir initialisé les modèles HMM par HCompv, nous avons ré-estimer ces modèles en appliquant HERest. Les modèles sont ensuite estimés de façon globale avec l'algorithme

La ligne de commande ci-dessus permet de ré-estimer les modèles avec l'outil HERest, qui calcule les valeurs optimales pour les paramètres HMM. Cette procédure doit être répétée plusieurs fois pour chaque HMM à entrainer.

Chaque fois, les itérations HERest sont affichées à l'écran ce qui peut vérifier au moyen de la vraisemblance. Dès que cette valeur ne décroit plus d'une itération à une autre, il est temps de stopper le procédé. Dans notre exemple, trois ré-estimations devraient être suffisantes (voir figures 4.22 et 4.23).

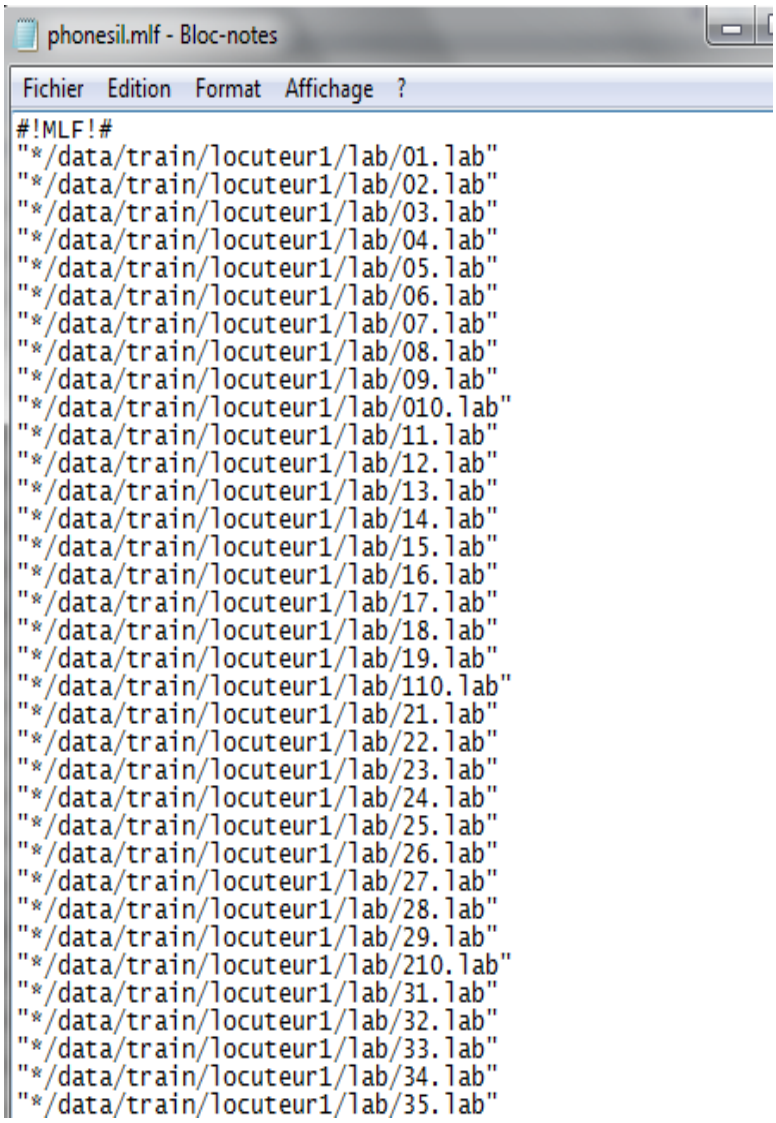


Figure 4.17 Le fichier phonesil.mlf

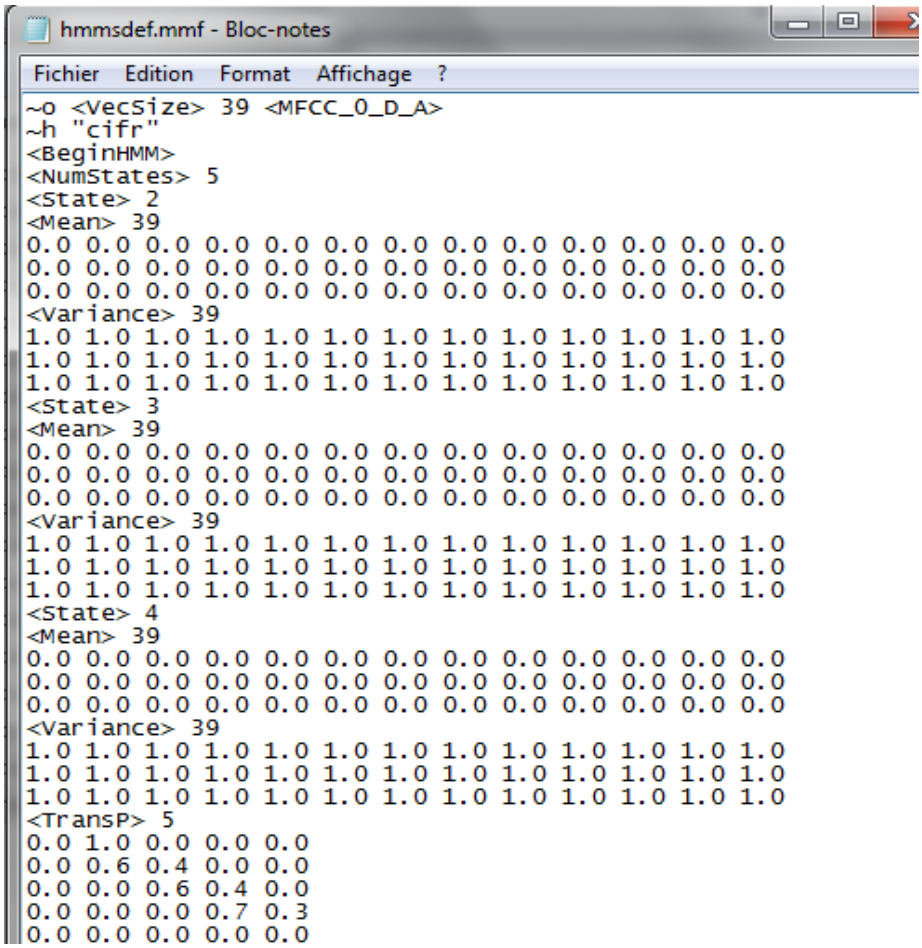


Figure 4.18 Le fichier hmmsdef.mmf

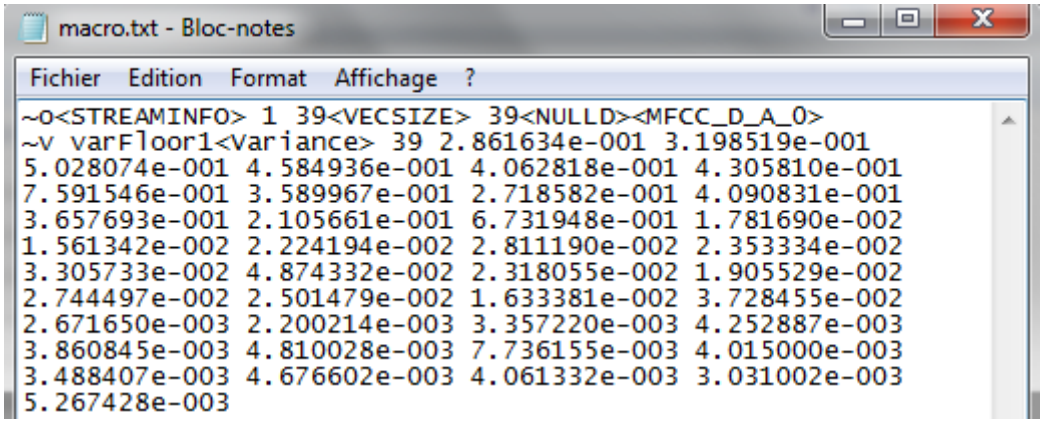


Figure 4.19 Le fichier macro.txt

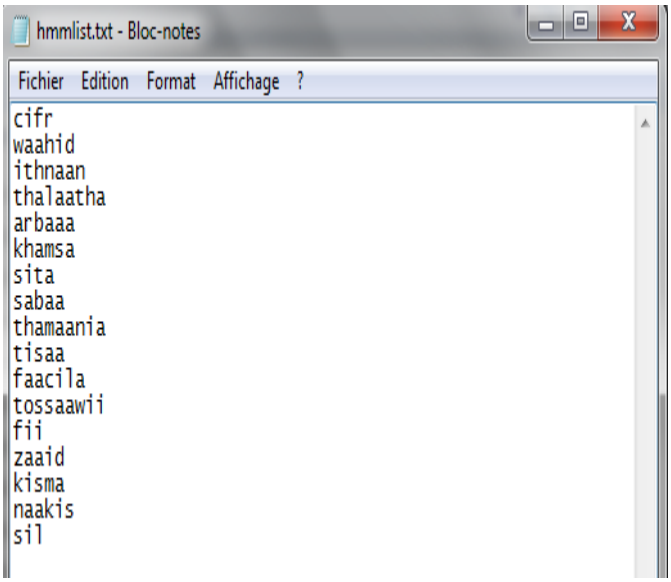


Figure 4.20 La liste des HMM dans le fichier hmmlist.txt

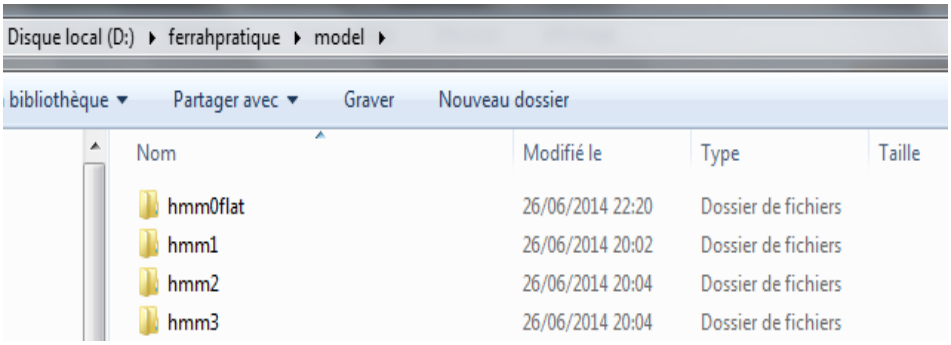


Figure 4.21 Les modèles HMM entrainés hmm1,hmm2,hmm3

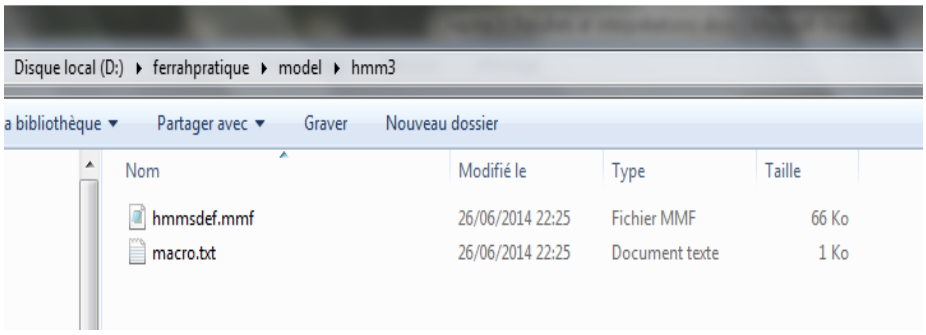


Figure 4.22 Le modèle final hmmsdef.mmf de tous les modèles HMM

c Insertion de la courte-pause « sp » à laide de HHEd

Dans la dernière étape où nous avons créé des modèles HMM qui ne comprennent pas un « sp » (courte pause) modèle silence, qui se réfère aux types de courtes pauses entre les mots qui se produisent dans parole.

Le modèle « sp » doit avoir son «état émission liée à l'état du centre du modèle de silence". Ce qui veut dire que nous avons besoin pour créer un nouveau modèle « sp » dans notre hmmsdef, qui utilisera l'état centre de sil, puis ils ont tous deux besoin d'être «lié » ensemble.

Cela peut être fait en copiant l'état du centre du modèle « sil » situé dans notre fichier hmm3 / hmmsdef.mmf et en ajoutant ce modèle « sp » dans un autre fichier hmm4 / hmmsdef.mmf, et de lancer l'outil **HHEd** sur 'égalité' du modèle « sp » au modèle « sil » afin qu'ils partagent le même état du centre comme il est indiqué dans la figure (4.23).

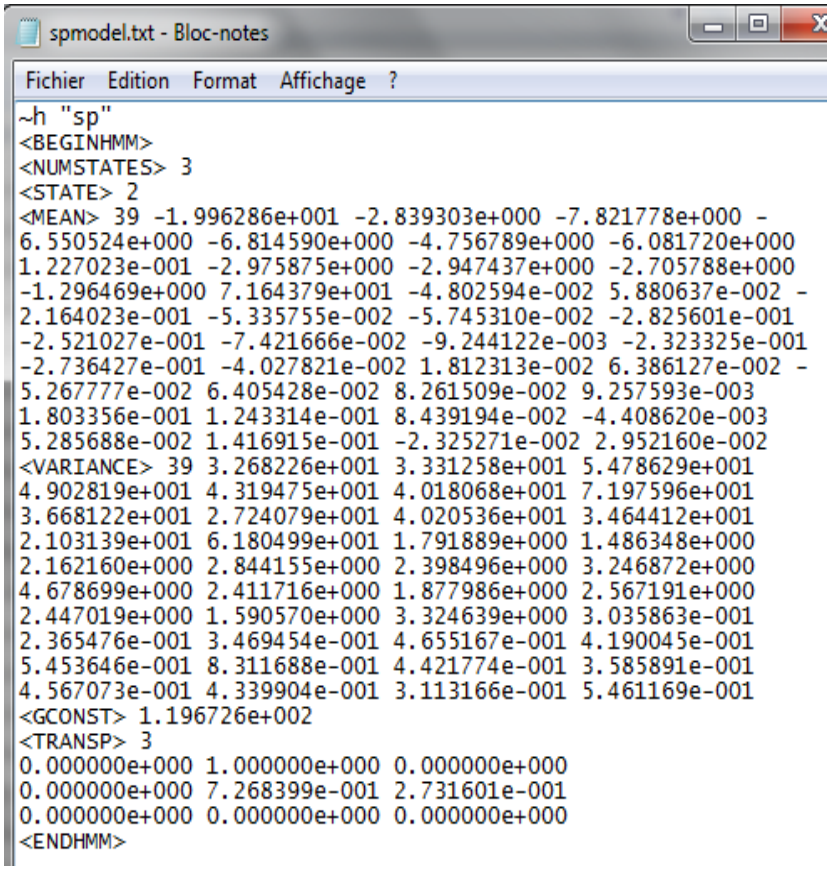


Figure 4.23 Le modèle « sp »

Ensuite, on lance l'exécution de **HHEd** pour lier l'état de sp à l'état du centre du modèle sil, pour qu'ils se partagent le même modèle HMM. Pour ce faire, on doit créer le script de appelé **sil.hed** (voir figure 4 .24) et monophones1.txt qui contient la liste les HMM et sp (voir figure 4.25). Le résultat est stocké dans hmm5.

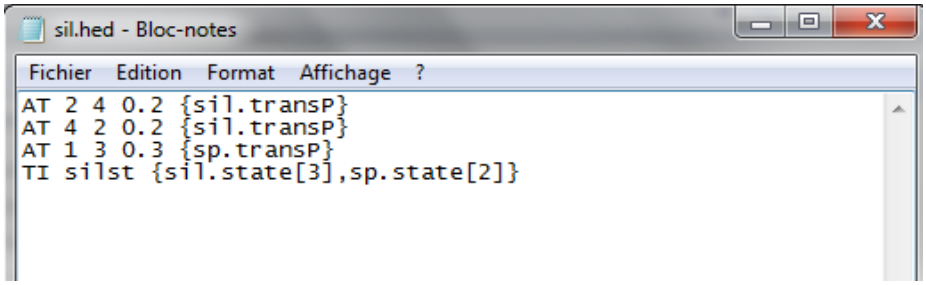


Figure 4.24 Le script sil.hed

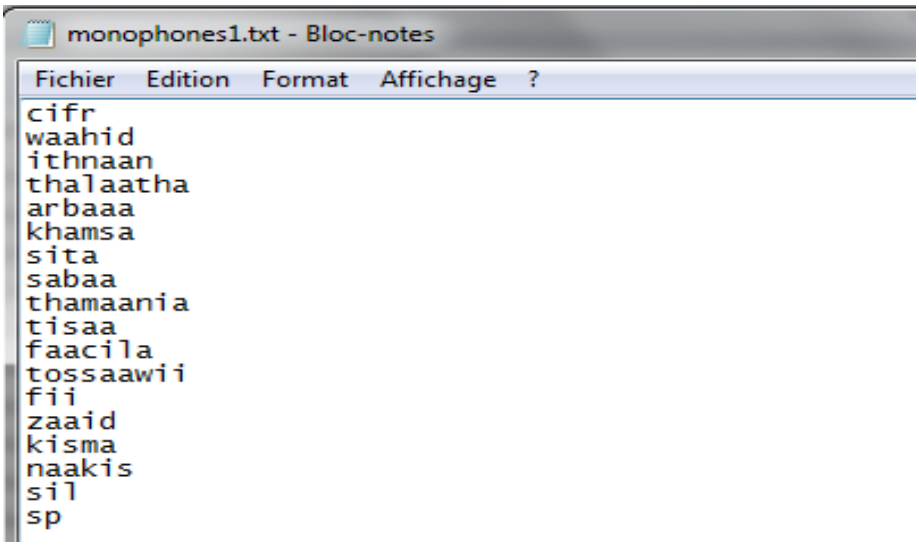


Figure 4.25 La liste des HMM en intégrant « sp»

Ensuite, on exécute HERest 2 fois plus, en utilisant cette fois le fichier monophones1.txt et le résultat est stocké dans hmm6 et hmm7 respectivement et cela en utilisant le fichier mlf, l'ensemble des fichiers de transcription contenant la pause (sp) entre chaque chiffre (voir la figure 4.26).

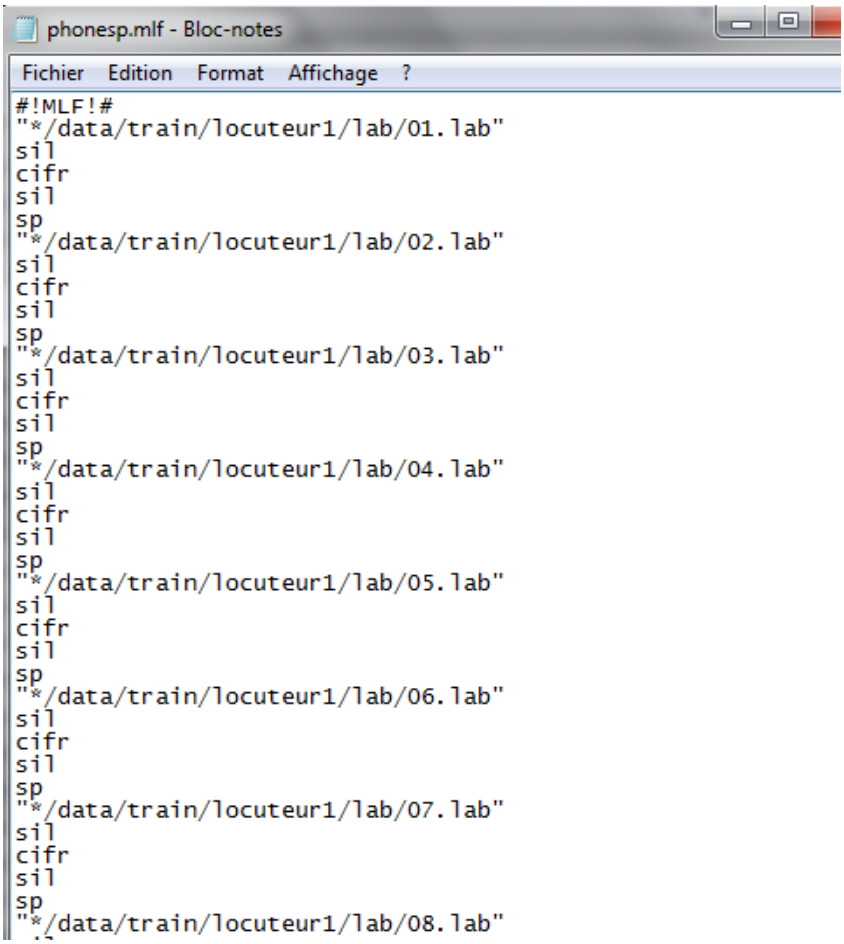


Figure 4.26 Le fichier de transcription contenant « sp » entre les mots

4.3.3 Phase d'apprentissage

Dans cette phase, nous avons utilisé les fichiers test qui ne sont pas utilisé dans la phase d'apprentissage. La reconnaissance consiste à calculer la vraisemblance du mot ou phrase inconnue à partir de tous les fichiers son du test, puis de décider le maximum de vraisemblance.

La recherche du chemin le plus probable d'état du HMM se fait par l'algorithme de Viterbi. Pour cette fin, nous avons utilisé l'outil Hvite de HTK

Toutes les données sont disponibles. Sauf, testlist.txt qui représente la liste les fichiers MFCC des fichiers test. Pour cela, nous avons calculé les coefficients MFCC de

ces fichiers en utilisant HCopy, avec le même fichier de configuration puisque ceux sont de même format. Mais le résultat est stocké dans targetlisttest.txt.

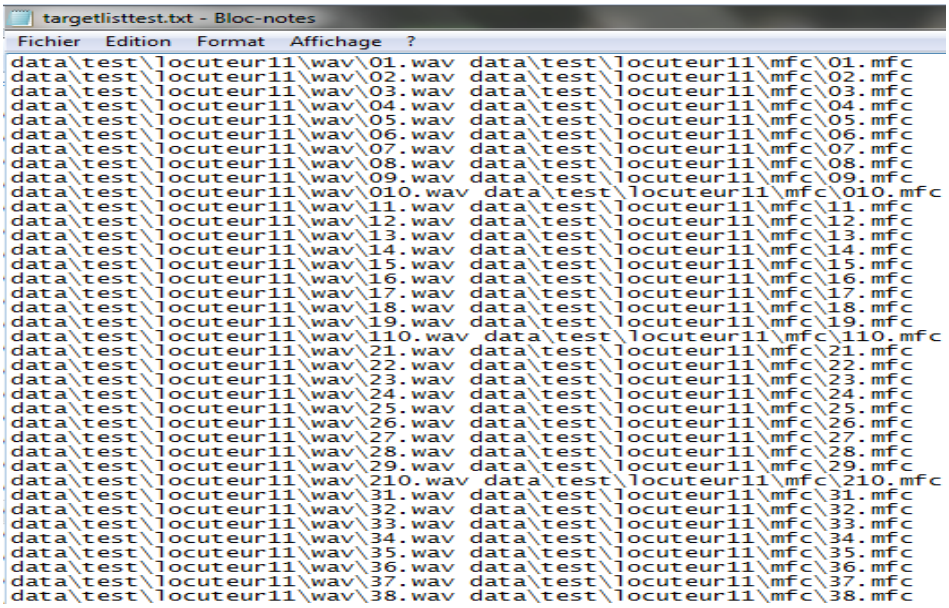


Figure 4.27 Le fichier **targetlisttest.txt**

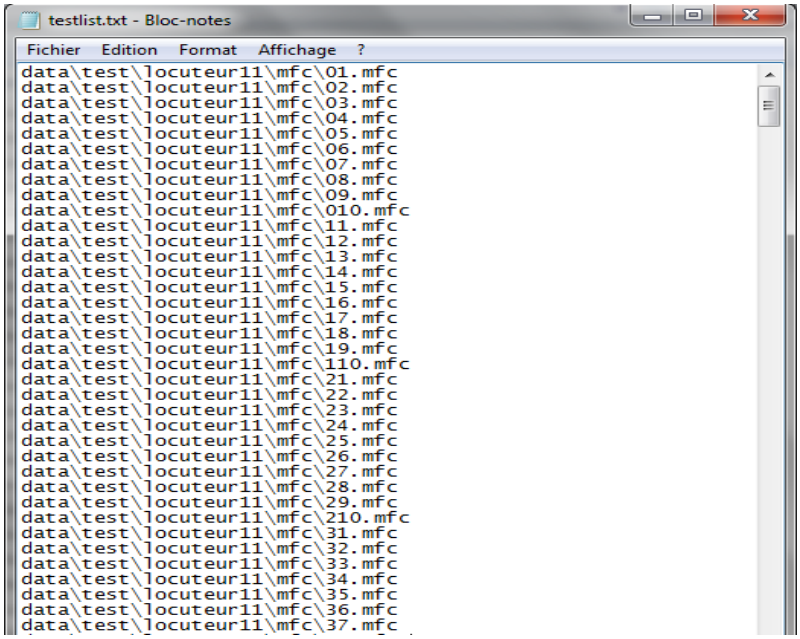


Figure 4.28 La liste des coefficients MFCC des fichiers test de locuteur 11

Après l'exécution de la commande HVite de l'outil HTK, il se génère les fichier

rec.mlf comme il est illustré dans la figure ci-dessous.

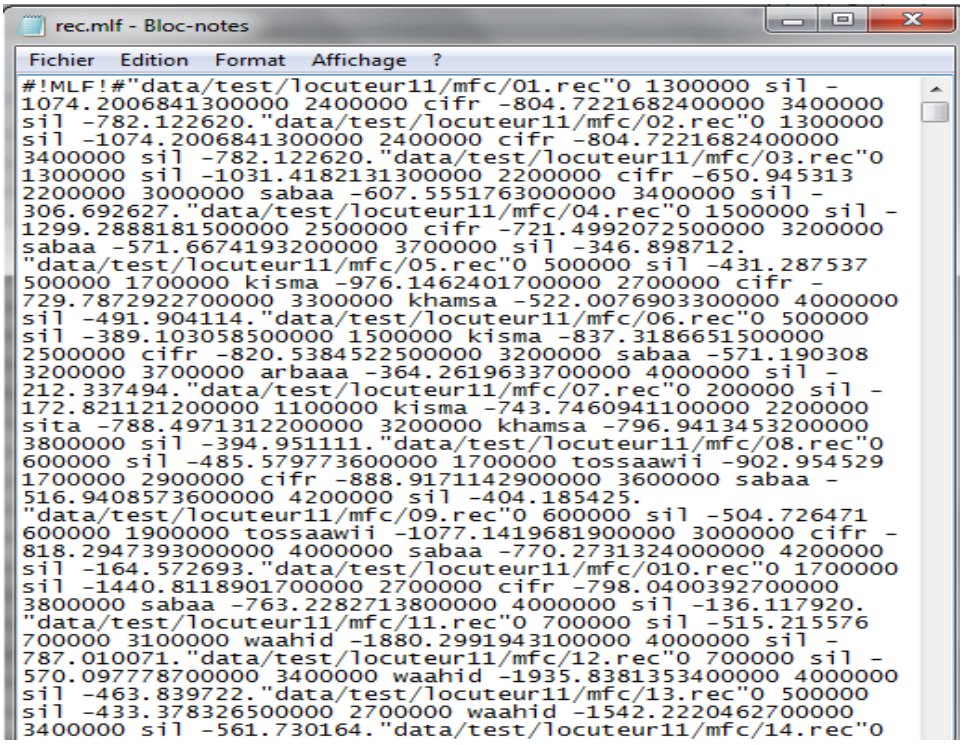


Figure 4.29 Le fichier rec.mlf résultant

4.3.4 Phase d'évaluation de la performance du système

Une fois que les données de test ont été générées, l'étape suivante consiste à analyser les résultats. L'outil HResult est prévu à cet effet. HResult compare les transcriptions contenant le fichier rec.mlf résultant par HVite avec les transcriptions de référence original.

Le fichier ref.mlf contient l'ensemble des transcriptions des fichiers étiquette de tous les fichiers test, monophones1.txt contient la liste des HMM, rec.mlf est le fichier résultant de HVite contient les fichiers reconnus avec ses transcriptions.

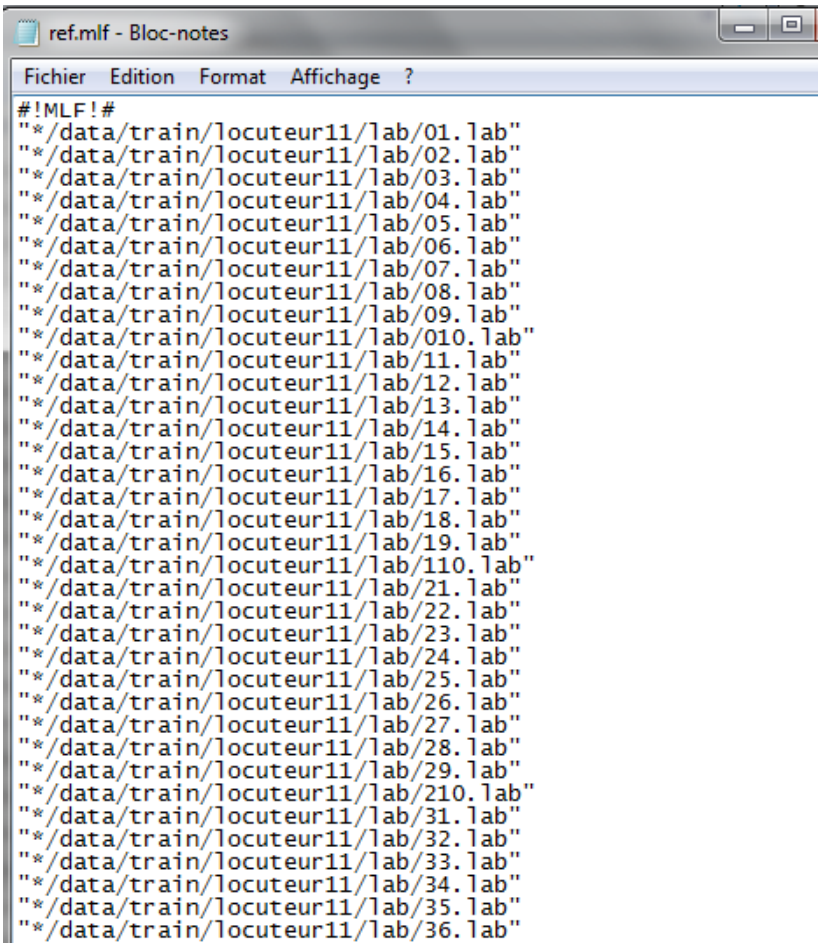


Figure 4.30 Le fichier ref.mlf

Les taux de reconnaissance, de substitution (S), d'insertion (I) et de suppression (D) ont été calculés aussi et représentés dans le fichier resultat.txt.

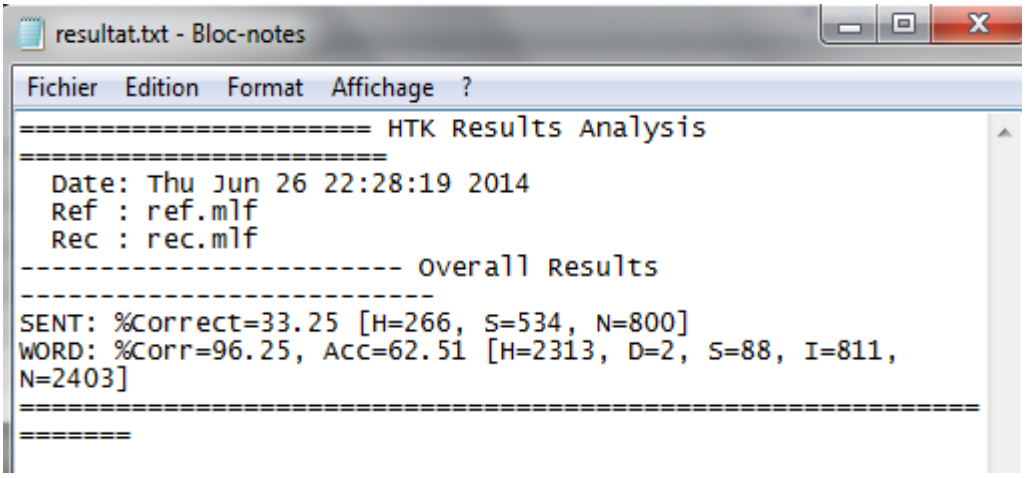


Figure 4.31 Le fichier rec.mlf

La première ligne commencée par SENT, donne le taux de reconnaissance de 800 fichiers de test, %correct = 33.25% car le système a substitué 534 étiquettes dans 534 fichiers test, pour cela S = 534, alors que les 226 fichiers restants sont correctement reconnus (H = 266).

La deuxième ligne commencée par WORD, donne le taux de reconnaissance de 2403 mots contenant les 800 fichiers test ; %corr = 96.25%, la précision Acc = 62.51%, 2313 mots correctement reconnus (H = 2313) , nombre de mots supprimés (D = 2) , nombre de mots insérés (I = 811) mais il y a 88 mots substituts (S = 88). Ces 88 mots substitués sont remplacés par d'autres mots du même vocabulaire, ces erreurs de reconnaissances sont créées à cause du mauvais étiquetage et si par exemple, un mot est substitué alors le système va le remplacer

4.5 Conclusion

Nous avons consacré ce chapitre pour la réalisation d'un système de reconnaissance de la parole Arabe à l'aide de HTK en prenant comme exemple une simple calculatrice vocale à base d'une approche qui est la méthode des Modèles de Markov Cachés (HMM).

Nous avons passé par quatre étapes essentielles : la préparation des données et le modèle de langage, l'apprentissage, la reconnaissance et finalement l'évaluation des performances du système de reconnaissance.

Nous avons paramétrisé chaque signal de parole en utilisant les coefficients cepstaux (MFCC) puis nous les avons modélisé sous forme de modèles HMM et après avoir insérer la pause (sp) entre les mots dans chaque fichier d'apprentissage et ré-estimer les modèles une autre fois en utilisant l'algorithme de Baum-Walch. Nous avons appliqué l'algorithme de Viterbi et obtenu les fichiers reconnu de ce système, sachant qu'on a utilisé une base de données BDWAVE pour l'apprentissage et le test. A la fin, on a obtenu un taux de reconnaissance de 33.25% et il est possible de dépasser ces performances avec ce système en utilisant une base de données mieux étiquetée et en augmentant le nombre des fichiers dans le corpus de test.

Conclusion générale

Le domaine de la reconnaissance de la parole nous a permis de découvrir beaucoup de choses très intéressantes concernant la programmation, l'outil informatique, la programmation sous MATLAB et l'exécution sou DOS qui nous a facilité l'édition et l'exécution de notre programme sous HTK pas à pas.

Dans notre travail, nous avons utilisé deux algorithmes essentiels :

- L'algorithme de Baum-Welch pour l'apprentissage et la ré-estimation des modèles acoustiques en utilisant l'approche des modèles HMM.

- L'algorithme de Viterbi pour la reconnaissance.

Nous avons obtenu un taux de reconnaissance de la base de données (BDSON) qui est égal à 96.25 % et une précision de 62.51%. Il est possible de dépasser ce taux en utilisant comme unité de traitement « le phonème », et en facilitant l'étiquetage des fichiers de transcriptions .

Ce système de reconnaissance est de type indépendant de locuteur car nous avons utilisé une base de données de la langue Arabe multi-locuteurs. Après avoir construit une simple grammaire et générer le réseau de mots équivalent en utilisant l'outil HParse, nous avons préparé un dictionnaire à la main puisque le vocabulaire est petit.

.

Bien sûr, nous avons préparé les données d'apprentissage et celles de test pour reconnaitre les performances de notre système concernant son taux de reconnaissance et sa précision. Nous avons calculé les coefficients MFCC de chaque fichier d'apprentissage en utilisant l'outil HCopy avant de réaliser le modèle HMM de chaque mot . Pour la modélisation, il faut passer par deux stades essentiels : l'initiation des modèles HMM en utilisant l'outil HCompv et la ré-estimation des modèles HMM jusqu'à la convergence après trois itérations en utilisant l'outil HERest.

Une fois les modèles HMM ré-estimés, nous avons inséré la petite pause (sp) en cas de pause entre les mots et ré-estimer une autre fois les HMM finaux.

En utilisant l'outil Hvite, nous avons obtenu les chaines de chiffres reconnus. Afin d'évaluer les performances du système il faut utilisé HResults pour afficher le résultat final concernant le taux de reconnaissance des chaines de mots de test qui est de 96.25% le taux de substitutions, d'insertion et de suppression, avec une précision de 62.51%.

Ces résultats sont satisfaisants malgré la possibilité d'améliorer plus les résultats, en utilisant les phonèmes comme unité de traitement et une base de données mieux étiquetée que notre base de données tel que TIMIT.

Le domaine de la reconnaissance de la parole reste difficile à cause de la complexité du signal de parole. Nous nous contentons des résultats obtenus dans cette étude puisqu'au stade où nous sommes, ce travail est pour nous une initiation à la recherche scientifique dans ce domaine.

Perspectives

Ce travaieut être perfectionné en utilisant une base de donnée mieux étiquetée et en élargissant le vocabulaire.

Nous pouvons aussi utiliser refaire le travail sur d'autres techniques de classification .

Annexes

ANNEXE - A- Les problèmes fondamentaux d'un HMM

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

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

A.1 L'algorithme FORWARD (problème d'évaluation)

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

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

Puis la probabilité inconditionnelle de la séquence partielle d'observation est la somme de *Pt*(*i*) sur tous les états *N*. L'algorithme Forward est un algorithme récursif pour calculer *αt*(*i*) pour la séquence d'observation à l'instant *t*. Tout d'abord, on calcule la probabilité de générer le premier symbole de la séquence par la formule *αt*(*i*) = *π*(*i*).*P* (*o*1|*i*), puis à chaque étape de l'induction,

$$\alpha_t(i) = (\sum_{i' \in S} \alpha_{t-1}(i').P(i' \to i)P(o_t|i)$$

on rajoute un symbole et on réitère la procédure jusqu'à ce que l'on ait calcule la probabilité de génération de la séquence entière et par la suite *P*(*O*|*λ*) par la formule:

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

A.2 L'Algorithme de Viterbi (problème de décodage)

Afin de résoudre le problème de décodage, l'algorithme de Viterbi est employé. Le critère d'optimalité ici est de rechercher un meilleur ordre simple d'état par la technique modifiée de la programmation dynamique. L'algorithme de Viterbi est un algorithme de recherche parallèle, à savoir il recherche le meilleur ordre d'état en traitant tous les états en parallèle.

Nous devons maximiser *P*(*Q*|*O, λ*) pour détecter le meilleur ordre d'état. Soie la probabilité *δt*(*i*) qui représente la probabilité maximale le long du meilleur chemin probable d'ordre d'état d'une séquence d'observation donné après *t* instants et en étant à l'état *i* ;

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

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

1. Initialisation :

$$\psi_1(i) = 0; \quad \delta_1(i) = \pi(i)P(o_1|i);$$

2. Induction :

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

Une fois les variables *δt*(*i*) et ψ *t*(*j*) calculées pour chaque étape de l'induction et pour chaque état, il ne reste plus qu'à lancer une procédure inductive de retro-propagation pour "dérouler" le chemin de Viterbi *s*1*...*sT* * :

1. Initialisation :

$$s_T{}^* = arg\max_{i \in S}(\delta_T(i))$$

2. Induction :

$$s_t{}^* = \psi_{t+1}(s_{t+1}{}^*), \quad t \in \{T-1...1\}$$

A.3 L'Algorithme de Baum-Welch (problème de d'apprentissage)

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

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

La probabilité qu'en générant *O* avec _ on passe par l'état *i* à l'instant *t* et par l'état *i*0 à l'instant *t* + 1. et en utilisant les variables forward et backward :

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

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

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

Si l'on somme ɣ*t*(*i*) sur l'ensemble des instants *t*, on obtient une quantité que l'on peut interpréter comme l'espérance du nombre de fois où l'état *i* est utilisé pour générer la séquence *O*. De même, si on somme *ξt*(*i*, *i*0) sur l'ensemble des instants *t*, on obtient une quantité que l'on peut interpréter comme l'espérance du nombre de fois où la transition *s* ! *s*0 est utilisée pour générer la séquence *O*. On a donc un es mateur ˆ*H* du HMM défini par les expressions suivantes :

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

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

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

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

ANNEXE - B - L'outils HTK

B.1 Présentation

HTK ou Hidden Markov Model ToolKit est un outil puissant, développé par Cambridge University Engineering Department (CUED), de construction et de manipulation des modèles de Markov cachés.

HTK est une boite à outils de modèle de Markov cachés HMM, conçue pour la construction et la manipulation de ces modèles . Cette boite est constituée d'un ensemble de module bibliothèque et d'outils disponibles en codes sources C. Chaque outil a un nombre d'arguments obligatoires en plus d'arguments optionnels préfixés par le signe "–".

B.2 L'architecture logicielle

La plupart des fonctionnalités de HTK est intégré dans les modules de la bibliothèque. Ces modules assurent que chaque outil communique avec le monde extérieur par le biais d'un interface . Ils fournissent également une ressource centrale de fonctions fréquemment utilisées. La figure (B.1) illustre l'architecture logicielle du HTK et ses interfaces d'entrée / sortie.

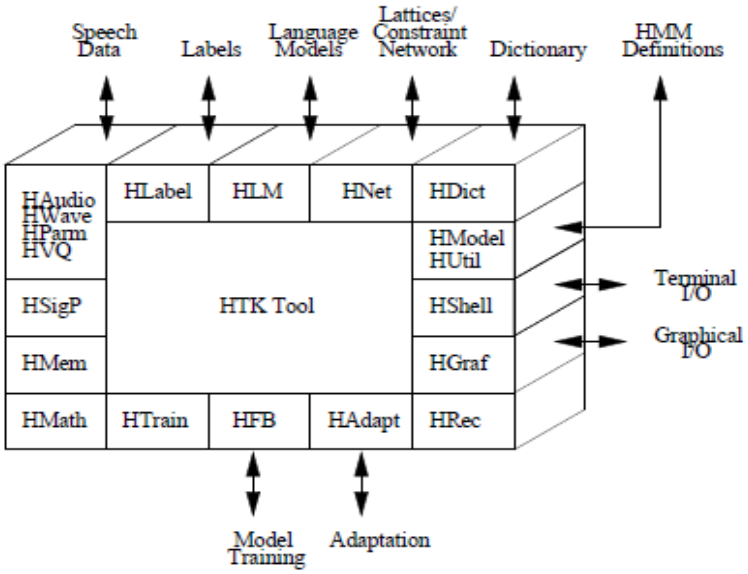


Figure B.1 L 'architecture logicielle de HTK.[23]

Le tableau (B.1) résume les différentes fonctions de ces modules :

N°	Modules	Explication
1	**HAudio**	Pour la prise en charge de l'entrée directe Audio.
2	**HWave**	Toute la parole entrée et sortie au niveau forme d'onde.
3	**HParm**	Toute la parole entrée et sortie au niveau paramétrisation .
4	**HVQ**	Pour VQ codebooks.
5	**HLabel**	fournir l'interface par des fichiers étiquetés
6	**HLM**	Utiliser pour les fichiers de modèles de langage.
7	**HNet**	pour les réseaux de mots équivalent (Network) et (lattices)
8	**HDict**	pour le dictionnaire
9	**HModel**	pour les définitions des modèles HMM
10	**HUtil**	Pour fournir un certain nombre de routines utilitaires pour manipuler les HMM.
11	**HGraf**	Pour fournir des graphiques interactifs .
12	**HShell**	Pour commander l'entrée/sortie utilisateur et l'interface avec le système d'exploitation.

13	**HRec**	contient les principaux fonctions de traitement de reconnaissance
14	**HAdapt**	Pour fournir un support pour les différents outils d'adaptation HTK.
15	**HFB**	Contiennent le support pour les divers outils d'apprentissage de HTK
16	**HTrain**	
17	**HMath**	Pour fournir le support Math .
18	**HMem**	Pour contrôler toute la gestion de la mémoire
19	**HSigP**	Utiliser pour les opérations nécessaires de traitement du signal et l'analyse de la parole

Tableau B.1 la fonction des différents modules de l'outil HTk.

B.3 Fonctionnement de l'outils HTK

La figure ci-dessous illustre le principe ce fonctionnement général de l'outil HTK

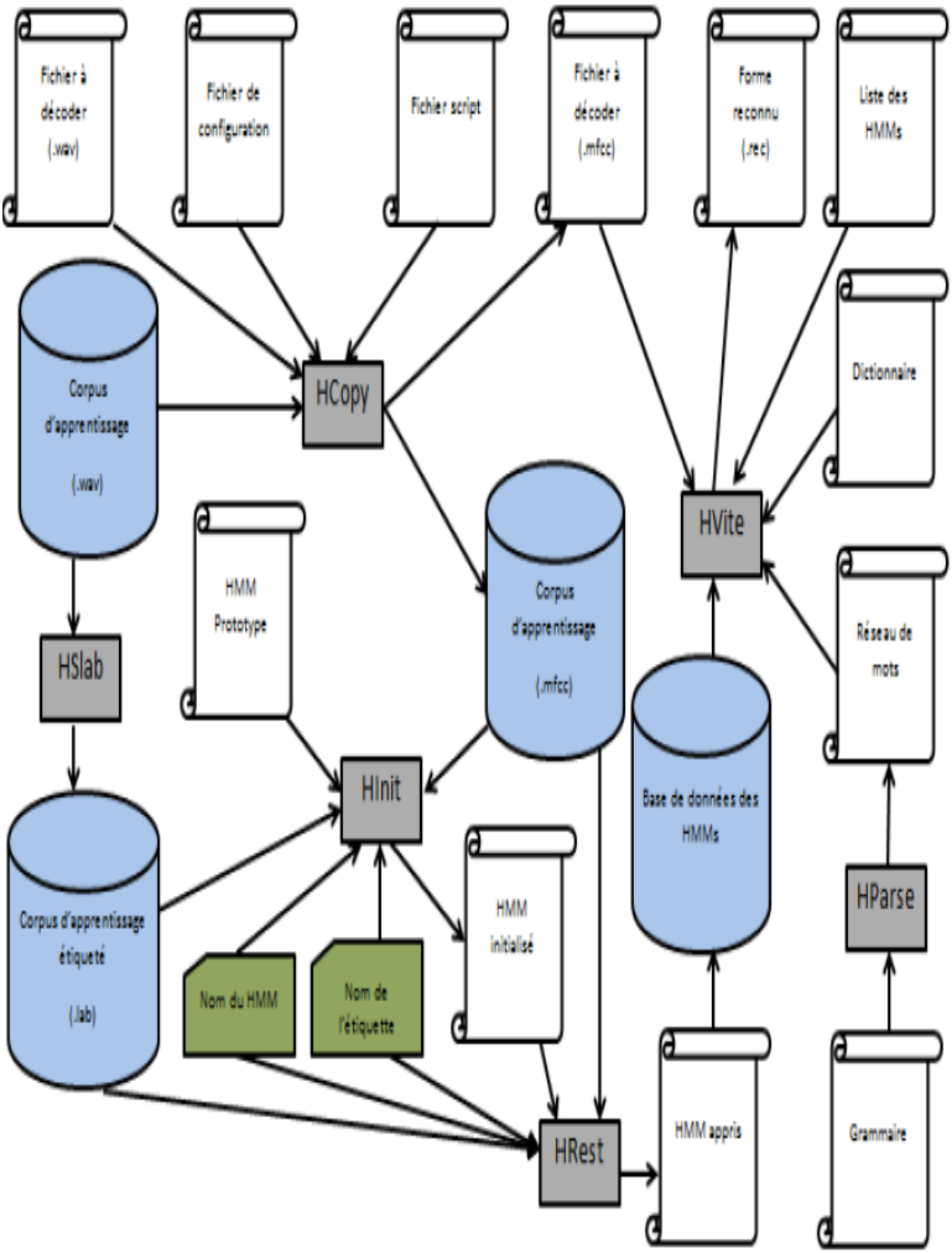


Figure B.2 Fonctionnement de l'outil HTK

B.4 Les codes d'erreurs

B.4.1 Generic Errors

+??00	Initialisation failed.

	The initialisation procedure for the tool produced an error. This could be due to errors in the command line arguments or configuration file.
+??01	Facility not implemented. HTK does not support the operation requested.
+??05	Available memory exhausted. The operation requires more memory than is available.
+??06	Audio not available. The audio device is not available, either there is no driver for the current machine, the library was compiled with NO AUDIO set or another process has exclusive access to the audio device.
+??10	Cannot open file for reading Specified file could not be opened for reading. The file may not exist or the filter through which it is read may not be set correctly.
+??11	Cannot open file for writing Specified file could not be opened for writing. The directory may not exist or be writable by the user or the filter through which the file is written may not be set correctly.
+??13	Cannot read from file. Cannot read data from file. The file may have been truncated, incorrectly formatted or the filter process may have died.
+??14	Cannot write to file. Cannot write data to file. The file system is full or the filter process has died.
+??15	Required function parameter not set. You have called a library routine without setting one of the arguments.
+??16	Memory heap of incorrect type. Some library routines require you to pass them a heap of a particular type.
+??19	Command line syntax error. The command line is badly formed, refer to the manual or the command summary printed when the command is executed without arguments.
+??9?	Sanity check failed. Several functions perform checks that structures are self consistent and that everything is functioning correctly. When these sanity checks fail they indicate the code is not functioning as intended. These errors should not occur and are not correctable by the user.

Tableau B.1 Generic errors

B.4.2 Summary of Errors

HSLab	**-1589**	ALIEN format set Input/output format has been set to ALIEN, ensure that this was intended.

HParse	**± 3130**	Variable not defined You have referenced a network that has not yet been defined. Check that all networks are defined before they are referenced.
	± 3131	Loop or word expansion error There is either a mismatch between the WD BEGIN WD END pairs or a triphone loop is badly formed.
	± 3132	Dictionary error When generating a dictionary a word exceeded the maximum number of phones, a word occurred twice or no dictionary was produced.
	± 3150	Syntax error in HParse file The HParse network definition contains a syntax error, check the input file against the network description .
HSGen	**- 3420**	Network malformed The word network is malformed. The information in a node (word and following arcs) is set incorrectly.
HCopy	**+1030**	Non-existent part of file specified. HCopy needed to access a non-existent part of the input file. Check that the times are specified correctly, that the label file contains enough labels and that it corresponds to the data file.
	±1031	Label file formatted incorrectly. HCopy is only able to properly copy label files with the same number of levels/alternatives. When using labels with multiple alternatives only the first one is used to determine segment boundaries.
	+1032	Appending files of different type/size/rate. Files that are joined together must have the same parameter kind and sample rate.
	- 1089	ALIEN format set Input/output format has been set to ALIEN, ensure that this was intended.
HCompV	**+2020**	HMM does not appear in HMMSet Supplied HMM filename does not appear in HMMSet. Check correspondence between HMM filename and HMMSet.
	+2021	Not enough data to calculate variance There are not enough frames of data to evaluate a reliable estimate of variance. Use more data.
	+2028	Load/Make HMMSet failed The model set could not be loaded due to either an error opening the file or the data within being inconsistent.
	+2030	Needs continuous models HCompV can only operate on models with an HMM set kind of PLAINHS or SHAREDHS
	+2050	Data does not match HMM An aspect of the data does not match the equivalent aspect in the HMMSet. Check the parameter kind of the data.
	- 2089	ALIEN format set Input format has been set to ALIEN, ensure that this was intended.
HERest	**+2320**	Unknown update flag Unknown flag set by -u option, use combinations of tmvw.
	+2321	Load/Make HMMSet failed The model set could not be loaded due to either an error opening the file or the data within being inconsistent.
	- 2326	No transitions No transition out of an emitting state, ensure that there is a transition path from beginning to end of model.
	+2327	Floor too high Mix weight floor has been set so high that the sum over all mixture components exceeds unity. Reduce the floor value.

	+2328	No mixtures above floor None of the mixture component weights are greater than the floor value, reduce the floor value.
	- 2330	Zero occurrence count Parameter has had no data assigned to it and cannot be updated. Ensure that each parameter can be estimated by using more training data or fewer parameters.
	- 2331	Not enough training examples Model was not updated as there were not enough training examples. Either reduce the minimum specified by -m or use more data.
	- 2389	ALIEN format set Input format has been set to ALIEN, ensure that this was intended.
HHEd	**+2628**	Load/Make HMMSet failed The model set could not be loaded due to either an error opening the file or the data within being inconsistent.
	± 2630	Tying null or different sized items You have executed a tie command on items which do not have the appropriate structure or the structures are not matched. Ensure that the item list refers only to the items that you wish to tie together.
	- 2631	Performing operation on no items The item list was empty, no operation is performed.
	+2632	Command parameter invalid The parameters to the command are invalid either because they refer to parts of the model that do not exist (for instance a state that does not appear in the model) or because they do not represent an acceptable value (for instance HMMSet kind is not PLAINHS, SHAREDHS, TIEDHS or DISCRETEHS).
	+2634	Join parameters invalid or not set Make sure than the join parameters (set by the JO command) are reasonable. In particular take care that the floor is low enough to ensure that when summed over all the mixture components the sum is below 1.0.
	+2635	Cannot find matching item Search for specified item was unsuccessful. When this occurs with the CL or MT commands ensure that the appropriate monophone/biphone models are in the current HMMSet.
	- 2637	Small gConst A small gConst indicates a very low variance in that particular Gaussian. This could be indicative of over-training of the models.
	- 2638	No typical state When tying states together a search is performed for the distribution with largest varianceand all tied states share this distribution. If this cannot be found the first in the list will be used instead.
	- 2639	Long macro name In general macro names should not exceed 20 characters in length.
	+2640	Not implemented You have asked HHEd to perform a function that is not implemented.

Bibliographie

[1] Auteur1, Auteur2 et Auteur3 : 'Titre du livre', éditeur, année de l'édition.

[2] Auteur1 et Auteur2 : 'Titre de l'article', Thème de la conférence, Pays, numéros de page, année.

[3] Auteur1 et Auteur2 : 'Titre de l'article', 'Titre de la revue, éditeur, numéros de volume et de page, année.