الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث 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 Master

Filière Télécommunication

Spécialité Système de Télécommunications

présenté par

CHEFFI Sid Ahmed

&

BOUCHAHMA Abderrahmane

Détection des mots clés dans un signal de parole

Proposé par :

M. ABED Ahcéne MCB USD Blida 1

Année Universitaire 2019-2020

REMERCIEMENTS

Tout d'abord, nous tenons à remercier Allah, le clément et le miséricordieux de nous avoir donné la force et le courage de mener à bien ce travail.

Un grand merci à notre encadreur Dr. ABED Ahcéne pour son soutien et sa disponibilité.

Nous voudrions aussi remercier tous les professeurs qui ont contribué à notre formation.

Merci enfin à tous ceux qui m'ont soutenu dans ce travail et qui n'ont pas été cités ici.

ملخص :

يندرج العمل المنجز ضمن أنظمة الكشف عن الكلمات المفاتيح في إشارة الكلام. يمكن اعتبار هذا النظام جزءا من عملية التعرف على الكلمات المعزولة. يسمح لنا النظام المقترح بتحديد المكالمات الهاتفية التي تحتوي على تهديدات معينة مثل التشهير أو المخاوف الأمنية. يرتكز النظام أساسا على نماذج ماركوف المخفية مع المعاملات *(MFCC)* و*(LPC)*. ولتحقيق أهدافنا ، قمنا بإنجاز قاعدة بيانات صوتية تغطي خمس كلمات لكل من الفئتين المدروستين (التشهير والأمن). قمنا بمحاكاة واختبار العديد من التكوينات اعتمادًا على عدد الحالات ونوع المعاملات. كما أظهرت النتائج التي تم الحصول عليها كفاءة النظام المقترح لجميع الكلمات المدروسة.

كلمات المفاتيح : التعرف على الكلام ؛ *HMM* ؛ *VQ* ؛ *DMC* ؛ *MFCC* ؛ *LPC*.

***Résumé**:*

Le travail réalisé s'inscrit dans le cadre de la détection de mots clés dans un signale de parole. Ceci peut être vu comme un problème de reconnaissance de mots isolés. Le système proposé nous permet d'identifier des appels téléphoniques contenant certaines menaces comme la diffamation ou les problèmes de sécurité. Le reconnaisseur utilisé repose sur les modèles de Markov cachés avec une représentation cepstral (MFCC) et prédictive (LPC) du signal de parole. Pour aboutir à nos objectifs, nous avons construit un corpus de parole couvrant cinq mots pour chacune des deux classes étudiées (diffamation et sécurité). Plusieurs configurations ont été simulées et testées en fonction du nombre d'états et de type de paramètres. Les résultats obtenus montrent l'efficacité du système proposé pour tous les mots étudiés.

Mots clés : Reconnaissance de la parole ; HMM ; QV ; DMC ; MFCC ; LPC.

Abstract :

The work carried out is part of keywords detection in a speech signal. This can be seen as a problem of isolated words recognition. The proposed system allows us to identify phone calls containing some threats such as defamation or security problems. The recognizer used is based on the Hidden Markov Models with a cepstral (MFCC) and predictive (LPC) representation of the speech signal. To achieve our objectives, we constructed speech corpora covering five words for each of the two studied classes (defamation and security). Several configurations were simulated and tested depending on the number of states and parameters type. The obtained results show the efficiency of the proposed system for all the studied words.

Keywords : Speech recognition ; HMM ; VQ ; DMC ; MFCC ; LPC.

Liste des Abréviations

RAL : Reconnaissance Automatique du Locuteur

DMC : Détection des Mots Clés

MFCC : Mel Frequency Cepstrum Coefficients

LPC : Liner Predictive Coding

PLP : Perceptual Linear Prediction

FFT : Fast Fourier Transform

F0 : Fréquence Fondamentale

DCT : Discrete Cosines Transform

EM : Expectation Maximisation

HMM : Hidden Markov Model

HTK : Hidden Markov Model toolkit

ANN : Artifice Neuron Network

MAP : Maximum a Posterior

MMI : Maximum mutual information

MLE : Maximum likelihood estimation

VQ : Vector Quantization

TCC : Taux de Classification Correct

TId : Taux d'identification

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

Les systèmes de reconnaissance de la parole ont connu un développement important durant ces dernières années. Ces systèmes nous ont permet de les utilisés pour la construction de plusieurs applications qui reposent sur le signal de parole. Nous avons exploité un système de reconnaissance automatique de parole pour l'adapter vers un système de détection de mots clés. Dont l'objectif principal est d'identifier la présence d'une menace en terme de diffamation ou de problème de sécurité dans un appel téléphonique. Le système proposé peut être utilisé dans plusieurs applications et surtout dans le domaine judiciaire.

Un système de Reconnaissance Automatique de la Parole (RAP) est un système qui a la capacité de reconnaître le message linguistique incorporé dans un signal de parole. L'entrée du système est généralement un signal de parole, dont la sortie est une suite de mots ou de phonèmes. La majorité des systèmes RAP repose sur le modèle de Markov caché qui peut être considéré comme l'état de l'art dans ce domaine.

Tout système de reconnaissance nécessite une opération d'analyse acoustique, en transformant le signal d'entrée en une suite de vecteurs acoustiques. Le signal de la parole est l'un des signaux les plus complexes, il n'est pas facile de le caractériser par un modèle simple. L'un des problèmes de la reconnaissance de la parole est la variabilité du signal. D'où plusieurs méthodes ont été introduites dans ces applications, on cite à titre d'exemple les paramètres MFCC (Mel Frequency Cepstral Coefficients) et les paramètres LPC (Linear Predictive Coding). Dans ce travail l'extraction des paramètres repose sur ces deux types de paramétrisation.

Avant d'étudier les performances de notre système et pour validé son efficacité nous avons construit un corpus de parole. Ce corpus couvre dix mots clés avec cinq mots pour la diffamation et l'autre pour le problème de la sécurité. Les phrases construites assurent la position des différents mots au début, au milieu et à la fin de la phrase. Les phrases sont enregistrées en utilisant le logiciel PRAAT à l'aide de 10 locuteurs. La fréquence d'échantillonnage est de $8000Hz$ pour l'adapté avec la bande passante téléphonique.

Ce mémoire se compose de trois chapitres :

Dans le premier chapitre, nous décrivons tout d'abord les caractéristiques du signal de

la parole, puis nous évoquons les difficultés liées à la reconnaissance vocale. Ensuite nous exposons les différentes étapes utiles à l'extraction des paramètres du signal. Puis nous présentons le décodage des informations acoustiques. Enfin nous détaillons, l'approche analytique, l'approche globale et l'approche statistique dans le domaine de la reconnaissance automatique de la parole.

Nous proposons dans le deuxième chapitre l'application des HMMs pour la reconnaissance automatique de la parole, en résolvant les trois problèmes fondamentaux des HMMs à savoir l'évaluation, le décodage et l'apprentissage. Ensuite nous présentons quelques généralités sur l'outil HTK et la quantification vectorielle.

Le troisième chapitre est réserve à l'évaluation expérimentale et l'étude des performances du système proposé. Nous avons mesuré le taux de classification correct et le taux d'identification pour les dix mots clés choisis. Ces deux taux sont étudiés en fonction du nombre d'états pour les trois types de paramètrisation : les MFCCs, les MFCCs avec ses paramètres dynamiques et les LPCs. Pour finir, nous présentons les conclusions et les perspectives de ce travail.

Chapitre 1 Reconnaissance Automatique de la Parole (RAP)

1.1 Introduction

La reconnaissance automatique de la parole est un domaine d'étude actif depuis le début des années 50. Il est clair qu'un outil de reconnaissance de la parole efficace facilitera l'interaction entre les hommes et les machines. Les applications possibles associées à un tel outil sont nombreuses et sont amenées à connaître un grand essor. La plupart des applications en reconnaissance de la parole peuvent être regroupées en quatre catégories : commande et contrôle, accès à des bases de données ou recherche d'informations, dictée vocale et transcription automatique de la parole.

Ce chapitre est organisé comme suite : nous présentons tout d'abord le signal de parole avec ses complexités et ses variabilités ; ensuite nous donnons un aperçu sur la reconnaissance automatique de parole ; enfin nous détaillons le décodage de l'information acoustique par les trois approches analytiques, globales et statistiques.

1.2 Signal parole

La parole est l'un des principaux moyens de communication entre les êtres humains, sa simplicité en fait d'ailleurs le moyen de communication le plus populaire dans la société humaine. Néanmoins, cette simplicité (pour l'être humain) renferme un traitement très complexe fait par notre cerveau, de la production de la parole jusqu'à sa perception et sa compréhension, ce qui rend la parole difficilement automatisable pour une machine [1].

Le signal de parole est une onde acoustique, qui est physiquement représentée comme un changement de pression atmosphérique provoqué et émis par le système phonatoire. Par conséquent, ce signal vocal se propage dans un milieu donné (en général l'air) et qui est le résultat de la modulation par le conduit vocal d'une onde d'excitation.[2]

Le signal de parole est une série de réalisations acoustiques de base. Ces réalisations Appelé phonème. Les phonèmes sont définis comme La plus petite unité acoustique, Chaque

langue peut être alors caractérisée par un ensemble de phonèmes qui constituent en quelque sorte les briques acoustiques élémentaires à partir desquelles les syllabes, les mots et les phrases sont construites. Tout signal de la parole peut alors être exprimé comme une succession de phonèmes. Ce signal véhicule un ensemble d'informations très diverses : le message que veut faire passer le locuteur, son humeur, son identité, etc. Le signal à reconnaître fait, dans un premier temps, l'objet d'un prétraitement, appelé paramétrisation, consistant à extraire de ce signal des paramètres pertinents permettant d'identifier la séquence des phonèmes prononcés [3].

1.2.1 Complexité du signal de parole

Les signaux vocaux ne sont pas des signaux ordinaires, mais des vecteurs des phénomènes extrêmement complexe. La reconnaissance automatique de la parole pose de nombreux Problèmes. D'un point de vue mathématique, il est difficile de modéliser les signaux vocaux parce que ses propriétés statistiques changeront avec le temps [4].

L'aspect continu du signal de parole ajoute une autre contrainte à la tâche de reconnaissance. En effet, lorsqu'on écoute la parole d'une personne, on perçoit une suite de mots, alors que l'analyse du signal vocal ne permet de déceler aucun séparateur. Le même problème de segmentation se retrouve à l'intérieur du mot lui-même. Celui-ci est perçu comme une suite de sons élémentaires, les phonèmes. L'analyse du signal ne permet pas aussi de découper en segments distincts le signal acoustique afin d'identifier les différents phonèmes qui le composent.

1.2.2 Variabilité du signal de parole

Le problème de la reconnaissance de la parole réside essentiellement dans la spécifié du signal vocal. Ce signal possède une très grande variabilité. Une même personne ne prononce jamais un mot deux fois de façon identique. La vitesse d'élocution peut varier, la durée du signal est alors modifie, toute altération de l'appareil phonatoire peut modifier la qualité du signal produit.

Dans le cas de la perception de la parole, le traitement du signal acoustique est complexe du fait des caractéristiques mêmes du signal de parole. Le premier point est le caractère directional du signal de parole. Contrairement au langage écrit où le traitement du mot entier est possible, le signal de parole subit une contrainte temporelle correspondant à l'ordre dans lequel les sons arrivent aux oreilles.

Le second point est la nature continue du signal de parole. Ce qui pose des difficultés, notamment, pour segmenter le signal en unités phonétiques discrètes.

La variabilité des sons de parole est en partie attribuée au phénomène de coarticulation. Celle-ci rend compte d'un chevauchement des gestes articulatoires sur l'axe temporel. Les indices acoustiques vont donc être distribués sur le signal. Au niveau des traits phonétiques, on parlera d'assimilation.

Ces trois caractéristiques, direction alité, continuité et variabilité, compliquent la reconnaissance des mots parlés. Pour que les mots soient correctement identifiés, deux problèmes majeurs doivent être résolus : la segmentation et la catégorisation [5].

1.3 Reconnaissance automatique de la parole

1.3.1 Historique de RAP

La reconnaissance automatique de la parole (RAP) s'agit d'un système matériel et logiciel qui permet de capter le son de la voix et d'identifier les mots prononcés. Il peut être considéré comme un système informatique qui permet d'analyser la parole captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine [6].

Parmi les techniques de traitement de la parole, la reconnaissance de la parole y rejoint la synthèse de la parole, l'identification du locuteur ou la vérification du locuteur. Des domaines en plein essor tant dans les laboratoires de recherches que dans nos produits technologiques de tous les jours. L'ensemble de ces techniques permettent notamment de réaliser des interfaces vocales c'est-à-dire des interfaces homme-machine (IHM) où l'interaction se fait à la voix [7].

La reconnaissance de la parole doit son existence à divers pans de la science. On peut citer en vrac, le traitement automatique des langues, la linguistique, le traitement du signal, les réseaux neuronaux, l'intelligence artificielle, etc.

1.3.2 Caractéristiques d'un système de RAP

Les caractéristiques du système RAP dépend directement du mode de fonctionnement, du mode d'élocution, des types de vocabulaire et de grammaire linguistique. Le système RAP peut être aussi caractérisé selon le nombre de locuteurs utilisant ces systèmes, ils sont répartis en trois catégories : mono locuteur et multi locuteur et indépendant du locuteur [8].

Mode d'élocution : Les systèmes de reconnaissance varient selon la voix. Généralement, trois types de systèmes de reconnaissance de mots sont utilisés :

- Système de reconnaissance de mots isolés : chaque mot est prononcé séparément marque une pause entre les mots.
- Système de reconnaissance de mots connecté : le système peut reconnaître une suite de mots Il n'y a pas de pause entre les mots.
- Système de reconnaissance de parole continue : la parole continue est le discours usuel, dans ce cas, un modèle de langage sera introduit.

Vocabulaire : C'est un ensemble de mots que le système peut reconnaître et se caractérise par :

- Sa taille peut varier de quelques mots à des dizaines de milliers de mots.
- Sa nature : par exemple, si nous utilisons un vocabulaire composé de mots phonétiquement proche, il est donc difficile de les distinguer.

Grammaire linguistique : La grammaire précise les restrictions imposées à la séquence parlée. Le but de la grammaire est de faciliter les tâches du système lors de la reconnaissance

1.3.3 Principe de fonctionnement de RAP

Le principe général de la reconnaissance automatique est résume par la figure 1 [9] :

- Le message en transmettre est converti en un signal acoustique Y par l'appareil phonatoire.
- Le signal acoustique est alors transformé en une séquence de vecteurs d'observation K.
- Finalement, le système de reconnaissance s'efforcera d'interpréter K en une séquence de mots R.

Le principe même de la reconnaissance automatique de la parole est de parvenir à une séquence de mots R correspondant au message transmis à partir de la séquence d'observation K.

1.4 Analyse acoustique

En termes d'acoustique, le son est généralement déterminé par son amplitude, sa durée et son timbre. Le traitement du signal vocal a pour but de quantifier ces trois grandeurs pour faire correspondre à l'onde sonore une description multidimensionnelle. En particulier

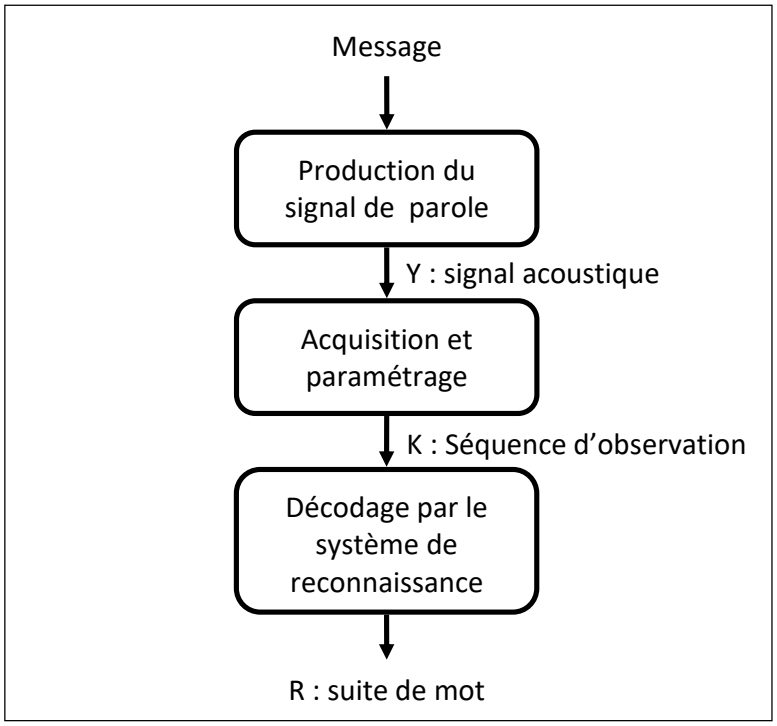

FIGURE 1.1 – Principe de la reconnaissance de parole

l'analyse acoustique du signal est utilisée pour résoudre le problème lié à la redondance du signal de parole et pour diminuer la quantité des calculs. Cette analyse permet de représenter le signal par des vecteurs de coefficients qui sont calculés sur des intervalles de temps [2].

Le paramétrage des signaux vocaux comprend l'extraction d'un ensemble de vecteurs Acoustique. Le but de cette opération est d'obtenir une nouvelle représentation, Plus compact et plus adapté à la modélisation statistique et vectorielle.

1.4.1 Mise en forme du signal parole

Avant l'extraction des paramètres acoustiques du signal vocal, la mise en forme du signal de la parole est nécessaire. Pour cela, l'ensemble des opérations suivantes sont prises en considération :

a Echantillonnage

L'opération d'échantillonnage consiste à transformer le signal à temps continu $x(t)$ en signal à temps discret $x(nT_e)$ défini aux instants d'échantillonnage, multiples entiers de la période d'échantillonnage T_e. Celle-ci est elle-même l'inverse de la fréquence d'échantillonnage f_e. Ce qui concerne le signal vocal, le choix de f_e résulte d'un compromis.

$$f_m \leq \frac{f_e}{2} \tag{1.1}$$

Par exemple, si vous enregistrez via une ligne téléphonique, alors $f_m = 3,5$ khz, ce qui signifie :

$$f_e > 7kHz \tag{1.2}$$

b Préaccentuation du signal

La préaccentuation est un filtre numérique du premier ordre qui passe après l'échantillonnage selon l'équation suivante :

$$H(z) = 1 - \alpha z^{-1} \quad (\alpha = 0,97) \tag{1.3}$$

c Segmentation du signal

La plupart des méthodes d'analyse acoustique utilisent des hypothèses de stationnarité du signal. Pour la parole, le signal est supposé être quasi stationnaire.

Deux types de segmentation sont utilisés, c'est-à-dire que le signal est segmenté en unités de trames de longueur variable et basé sur l'algorithme de segmentation automatique. L'algorithme peut être uniforme, qui divise le signal en plusieurs trames chevauchant de longueur fixe. La longueur de trame varie de $20ms$ à $40ms$. Si la longueur est égale à $32ms$ et $f_e = 8kHz$, le nombre d'échantillons par trame est de 256 échantillons.

d Application de fenêtre de pondération (Hamming):

L'application d'une fenêtre de pondération (fenêtre de Hamming par exemple). Pour ne pas perdre d'information et assurer un meilleur suivi des non-stationnarités, les fenêtres se recouvrent. Elles ont généralement une longueur de 256 ou 512 points et le recouvrement est de 50%, soit 128 ou 256 points.

Pour réduire ces effets le signal est multiplié par une fenêtre $w(n)$ dont la transformée de Fourrier s'approche d'une impulsion de Dirac. Le nouveau signal devient :

$$x_n = x_n w(n) \tag{1.4}$$

Fenêtrage de Hamming, son équation est :

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right) \tag{1.5}$$

II existe plusieurs types de fenêtrage par exemple fenêtrage de Hanning et de Blackman, et le fenêtrage de Kaiser.

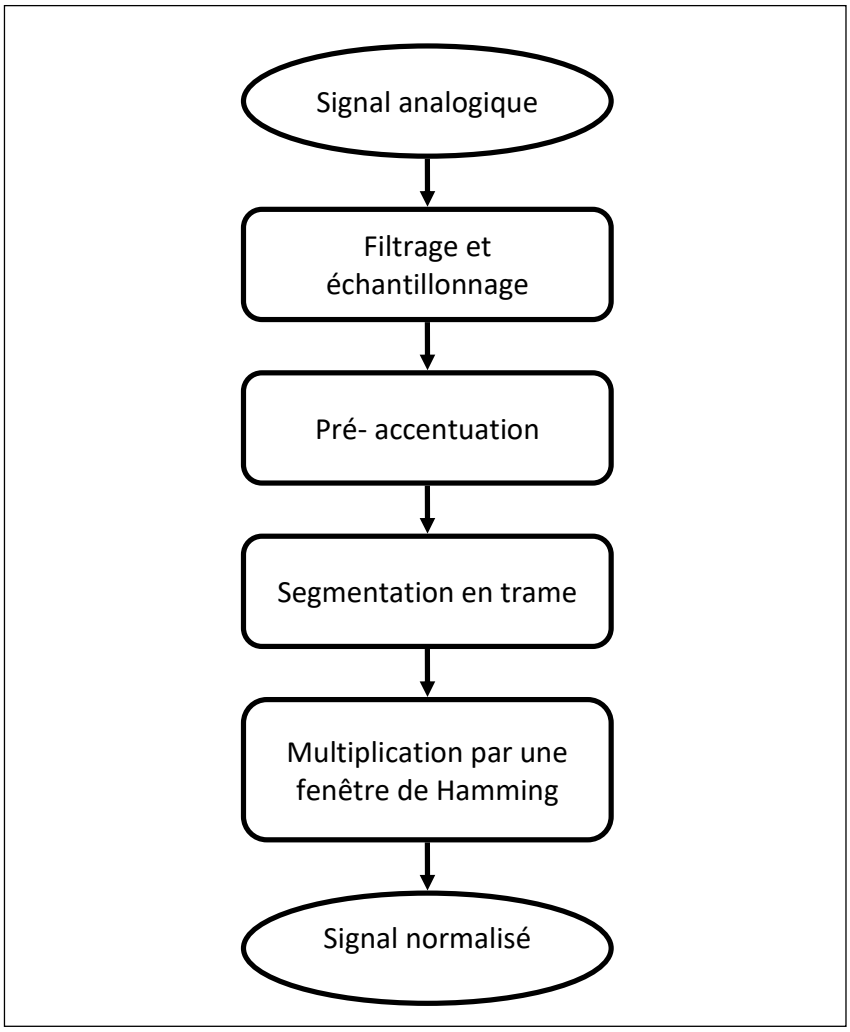

FIGURE 1.2 – Mise en forme d'un signal de parole

1.4.2 Extraction des paramètres

L'objectif de cette phase de reconnaissance est d'extraire des coefficients représentatifs du signal de parole. Ces coefficients sont calculés à intervalles temporels réguliers. En simplifiant les choses, le signal de parole est transformé en une série de vecteurs de coefficients, ces coefficients doivent représenter au mieux ce qu'ils sont censés modéliser et doivent extraire le maximum d'informations utiles pour la reconnaissance.

Parmi les coefficients les plus utilisés et qui représentent au mieux le signal de la parole en reconnaissance de la parole, nous trouvons les coefficients cepstraux, appelés également

cepstres. Les deux méthodes les plus connues pour l'extraction de ces cepstres sont : l'analyse spectrale et l'analyse paramétrique. Pour l'analyse spectrale (par exemple, Mel-Scale Frequency Cepstral Coefficients (MFCC)) comme pour l'analyse paramétrique (par exemple, le codage prédictif linéaire (LPC)). Le signal de parole est transformé en une série de vecteurs calculés pour chaque trame. Ces coefficients jouent un rôle capital dans les approches utilisées pour la reconnaissance de la parole.

a Représentation cepstrale

La parole peut être représentée sous la forme d'un modèle source-filtre. Cette représentation permet ainsi de représenter le signal de parole $s(t)$ sous la forme de la convolution du signal source $g(t)$ par la réponse impulsionnelle du filtre $h(t)$ représentant le conduit vocal :

$$s(t) = g(t) * h(t) \tag{1.6}$$

L'étude de ce signal à l'aide de la FFT présente un défaut particulier liée à cette convolution qui rend difficile l'observation de la seule contribution du conduit vocal. Le cepstre (parfois appelé lissage cepstral) permet de séparer les contributions respectives de la source et du conduit vocal.

En effet, l'équation précédente se récrit dans le domaine spectral sous la forme :

$$S(\omega) = G(\omega)H(\omega) \tag{1.7}$$

Où $S(\omega)$, $G(\omega)$ et $H(\omega)$ représentent respectivement les transformées de Fourier de $s(t)$, $g(t)$ et $h(t)$

Le cepstre qui est défini par la transformée de Fourier inverse du logarithme de module de $S(\omega)$ s'écrit donc sous la forme :

$$c(\tau) = FFT^{-1} \log |S(\omega)| = FFT^{-1} \log |G(\omega)| + FFT^{-1} \log |H(\omega)| \tag{1.8}$$

Lorsque le cepstre est obtenu en calculant la transformée de Fourier discrète, on obtient la forme suivante :

$$c_n = \frac{1}{N} \sum_{k=0}^{N-1} \log |X(k)| e^{\frac{2j(\pi)kn}{N}} \qquad 0 \leq n \leq N-1 \tag{1.9}$$

b Paramètres MFCC "Mel Frequency Cepstral Coefficients"

Le codage MFCC est une technique très utilisée en traitement de la parole. Il est basé sur la variation des bandes critiques de l'oreille humaine avec la fréquence, les filtres espacés linéairement aux basses fréquences et logarithmiquement à hautes fréquences [10]. Ces filtres sont modélisés par une échelle non-linéaire issue de connaissances sur la perception humaine : l'échelle Mel.

La fréquence de l'échelle de Mel est définie par

$$Mel(f) = \frac{1000}{\log 2} \log\left(1 + \frac{f}{1000}\right) \tag{1.10}$$

Où f est la fréquence en Hz, $Mel(f)$ est la fréquence Mel-échelle de f.

L'avantage de l'échelle Mel est d'être assez proche d'échelles issues d'études sur la perception sonore et sur les bandes passantes critiques de l'oreille.

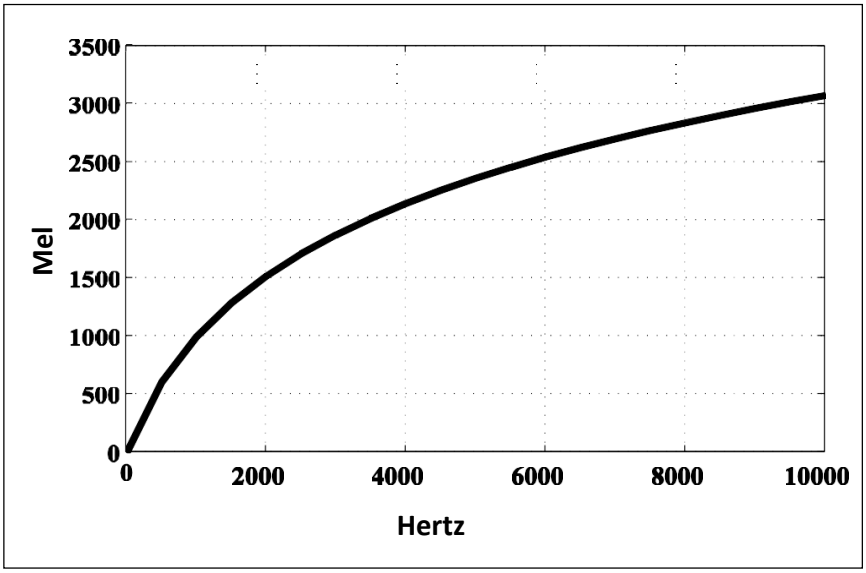

FIGURE 1.3 – Échelle Mel

Le calcul des paramètres MFCC se réalise à partir des étapes suivantes :

Pour les MFCCs, on utilise la fenêtre de Hamming durant la transformation du domaine temporel au domaine fréquentiel. Cette transformation est faite en utilisant la transformée de Fourier.

Si x_n est le signal observé alors :

$$x_n = g_n * h_n \tag{1.11}$$

g_n Représente la source glottique et h_n représente la réponse impulsionnelle.

La transformée de Fourrier sur le produit de convolution :

$$X_\omega = TF(x_n) = TF(g_n)TF(h_n) \tag{1.12}$$

Puis :

$$\log|X_\omega| = \log|TF(g_n)| + \log|TF(h_n)| \tag{1.13}$$

Dont :

— $|X_\omega|$ est le spectre d'énergie de x_n.

— $TF(x_n)$ est la transformée de Fourrier de x_n

Pour minimiser le temps de calcul, il est préférable d'utiliser l'algorithme FFT (Fast Fourier Transform).

Pour les MFCCs, on utilise la fenêtre de Hamming durant la transformation du domaine temporel au domaine fréquentiel. Cette transformation est faite en utilisant la transformée de Fourier [9].

Un filtrage, est appliqué ensuite, par un banc de filtres triangulaires espacés selon l'échelle de Mel. Cette échelle reproduit la sélectivité de l'oreille qui diminue avec l'accroissement de la fréquence.

Afin de réduire ces phénomènes, nous lissons ces spectres en appliquant une série de filtres triangulaires répartis sur la bande passante $[100Hz, 7,5kHz]$ selon l'échelle de Bark ou du Mel, pour se rapprochez de l'oreille humaine

L'équation de l'échelle de Bark :

$$Bark(f) = 6\, Arcsinh\left(\frac{f}{600}\right) \tag{1.14}$$

f représente la fréquence

Ensuite, Les limites de ces filtres sont exprimées sur l'échelle de mel.

Après le calcul de log, une transformée de Fourrier inverse est appliquée pour assurer un retour au domaine temporel.

La formule de calcul de cette transformée donne les coefficients cepstraux C_k et qui sont notés MFCC.

$$C_k = \sum_{i=1}^{F} \log(e_i) \cos\left(\frac{\pi k(i-0.5)}{F}\right) \qquad k = 1, ..., d \tag{1.15}$$

Avec : d est le nombre de coefficients cepstraux

Figure 1.4 illustre l'algorithme de calcul des coefficients MFCC.

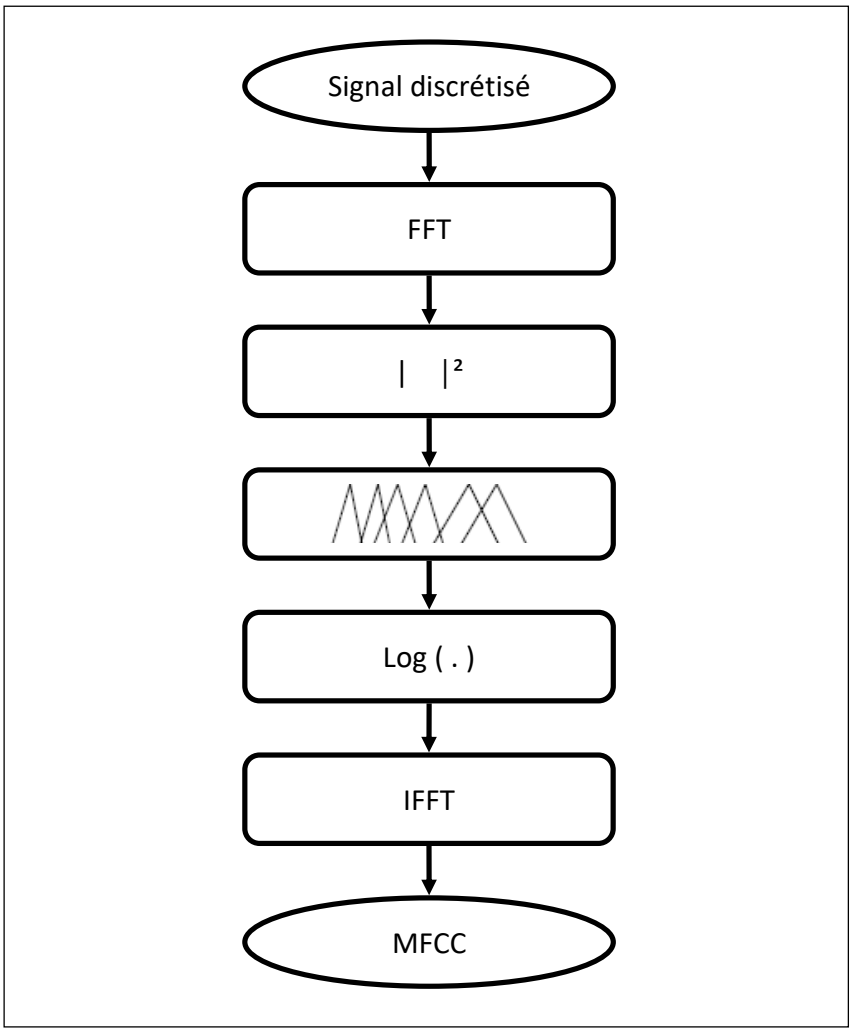

FIGURE 1.4 – Calcul des MFCC

c Paramètres LPC (Linear Predictive Coding)

Il consiste à synthétiser des échantillons de signal de parole à partir d'un modèle de système de production vocale et d'excitation. Il permet de prédire une valeur future du signal à partir d'une combinaison des valeurs précédentes. Le codage LPC est l'une des techniques les plus puissantes d'analyse de la parole qui a gagné en popularité en tant que technique d'estimation de formants. Les coefficients LPC sont calculés en découpant le signal de la parole en de petites fenêtres de courte durée. La fenêtre de Hamming est ensuite appliquée sur les différentes portions de signal obtenues. L'application de la fenêtre de Hamming permet de diminuer la distorsion spectrale. Avec l'équation :

$$s(n) = \sum_{k=1}^{p} a_k s(n-k) + e(n) \tag{1.16}$$

Le signal à l'instant n est prédit à partir des p échantillons précédents.

La moyenne que constitue la somme pondérée du signal sur p pas de temps introduit une erreur car la parole ne constitue pas un processus parfaitement linéaire. Cette erreur est corrigée par l'introduction du terme $e(n)$. Le codage par prédiction linéaire consiste donc à déterminer les coefficients a_k qui minimisent l'erreur $e(n)$, ceci en fonction d'un ensemble de signaux constituant un corpus d'apprentissage [11].

1.4.3 Paramètres dynamiques

Les informations sont souvent données par les dérivés cepstrales. La première dérivée des coefficients cepstraux s'appelle les coefficients Delta, et la dérivée deuxième des coefficients cepstraux s'appelle les coefficients Delta-Delta. Les coefficients delta nous donnent quelques informations sur la variation de ces vecteurs dans le temps, et les coefficients Delta-Delta nous donnent des informations sur l'accélération de la parole. Ces coefficients sont données par :

$$\Delta c_m = \frac{\sum_{k=-1}^{l} k^2 c_{m+k}}{\sum_{k=-1}^{l} |k|} \tag{1.17}$$

$$\Delta\Delta c_m = \frac{\sum_{k=-1}^{l} k^2 c_{m+k}}{\sum_{k=-1}^{l} k^2} \tag{1.18}$$

1.5 Décodage des informations acoustiques

Pour décoder les informations acoustiques, on distingue généralement en reconnaissance de la parole trois approches : l'approche analytique, l'approche globale et l'approche statistique. La première approche cherche à traiter la parole continue en décomposant le problème, le plus souvent en procédant à un décodage acoustique phonétique exploité par des modules de niveau linguistique. La seconde consiste à identifier globalement un mot ou une phrase en le comparant avec des références enregistrées, la troisième approche permet aussi d'intégrer les niveaux acoustiques et linguistiques dans un seul processus de décision.

1.5.1 Approche analytique

La méthode d'analyse tente de résoudre le problème de la reconnaissance de parole continue par isolement des unités sonores courtes (petite taille), telles que les phonèmes, les diphones ou syllabe. Un exemple classique de cette approche est l'analyse par traits des

indices acoustiques calculés à partir du signal de parole, ils permettent de faire des hypothèses locales sur certains traits phonétiques, comme le voisement, la nasalisation, le lieu d'articulation ou le degré d'ouverture du conduit vocal. En fonction de ces traits, le signal acoustique est segmenté et une identification phonétique des segments est réalisée.

Cette méthode est plus adaptée aux grands systèmes de vocabulaire et de la parole Continue. Le décodage acoustique-phonétique utilisé par cette approche est une tâche difficile, peu de systèmes s'approchent du taux de reconnaissance souhaitable

1.5.2 Approche globale

L'approche globale traite les mots ou les phrases comme entité de base et comparaison avec la référence enregistrée. Plusieurs exemples de chaque mot qui peut être reconnu, sont enregistrés comme vecteurs acoustiques.

L'étape de reconnaissance consiste à analyser le signal inconnu sous la forme d'une suite de vecteurs acoustiques similaires, et à comparer la suite inconnue à chacune des suites des exemples préalablement enregistrés. Leur essor en reconnaissance de parole est dû à l'exploitation de critères de comparaison performants, comme l'alignement temporel dynamique des formes acoustiques, et à leur application à des représentations adaptées du signal, qu'il s'agisse de l'analyse spectrale ou de la prédiction linéaire. L'approche globale a une grande capacité de reconnaissance et une indépendance vis-à-vis des particularités de la langue à reconnaître, mais elle utilisé vocabulaires très limités

1.5.3 Approche statistique

L'approche statistique permet aussi d'intégrer les niveaux acoustiques et linguistiques dans un seul processus de décision (ce qui est impossible dans l'approche analytique). Les unités acoustiques modélisées peuvent être des mots comme dans le cas de l'approche globale, comme elles peuvent être des unités plus courtes comme les phonèmes (le cas de l'approche analytique).

1.6 Conclusion

Dans ce chapitre nous avons vu les difficultés lies aux signaux de parole et sa complexité, et nous avons présenté quelques notions générale sur le système de la reconnaissance automatique de parole. Ainsi nous avons étudié l'analyse acoustique par une extraction des paramètres MFCC et LPC. Et enfin nous avons terminé par une description des approches utilisés

pour le décodage de l'information acoustique. Dans le chapitre qui se suivre nous allons présenter quelques techniques utilisés pour RAP comme les HMMs, le HTK, et la quantification vectoriel.

Chapitre 2 Implementation du système de détection de mots clés DMC

2.1 Introduction

Avons de mettre en place le système proposé pour la détection de mots clés, nous allons tout d'abord décrire les techniques les plus utilisées dans les système de reconnaissance automatique de parole. Nous présentons également les modèles de markov cachés ainsi que sa variante nommée outil HTK. En suite, nous donnons un aperçus sur les réseaux de neurones et la quantification vectorielle. Enfin, nous allons détailler les différentes étapes pour construire notre système de détection de mots clés. Le système proposé est basé les modèles de markov cachés avec une représentation basée sur les MFCC et les LPC pour l'extraction des paramètres.

2.2 Modèles de Markov Cachés HMM

La technologie la plus utilisée depuis plus de 20 ans est basée sur des modèles statistiques : les modèles de Markov cachés (en anglais Hidden Markov Models : HMM) capables de modéliser simultanément les caractéristiques fréquentielles et temporelles du signal de parole. Depuis l'introduction de ces modèles, de nombreux progrès ont été réalisés dans le domaine de la reconnaissance de la parole. Néanmoins, les performances obtenues sont encore largement inférieures à celles des êtres humains, même si les progrès réalisés en moins de 50 ans sont énormes.

2.2.1 Généralités sur les HMMs

Un problème majeur de la reconnaissance de la parole est de modéliser au mieux des unités représentatives du signal de parole. Il existe en fait deux types de modélisation possibles des propriétés d'un signal donné [12] :

— La modélisation déterministe, qui exploite les propriétés intrinsèques du signal.

— La modélisation statistique, qui caractérise les propriétés statistiques du signal.

Un HMM peut être vu comme un ensemble discret de nœuds ou d'états et de transitions ou d'arcs reliant ces états entre eux. Formellement, il peut être défini par l'ensemble des paramètres [13] :

$$\lambda = (N, A, B, \pi) \tag{2.1}$$

— N est le nombre de nœuds ou d'états du modèle.

— $A = a_{ij} = P(q_j|q_i)$ est la matrice des probabilités de transition sur l'ensemble des états du modèle. La probabilité de transition est la probabilité de choisir la transition a_{ij} pour accéder à l'état q_j, étant donné un processus à l'état q_i. Pour un HMM d'ordre un, cette probabilité ne dépend que de l'état précédent :

$$p(q_t = j|q_{t-1} = i, q_{t-2} = k, ...) = p(q_t = j|q_{t-1} = i) \tag{2.2}$$

Elle dépend des deux précédents dans le cas d'un HMM d'ordre deux :

$$p(q_t = j|q_{t-1} = i, q_{t-2} = k, ...) = p(q_t = j|q_{t-1} = i, q_{t-2} = k) \tag{2.3}$$

En d'autres termes, l'évolution du système entre deux instants $t-1$ et t ne dépend que de l'état de ce système au temps $t-1$ (ordre 1) ou des deux instants précédents $t-1$ et $t-2$ (ordre deux).

— $B = b_j(o_t) = P(o_t|q_j)$ est l'ensemble des probabilités d'émission de l'observation o_t dans l'état q_j. La forme que prend cette distribution détermine le type du HMM. C'est ainsi qu'on parle de HMMs discrets, semi-continus, continus, etc. Pour plus d'informations sur les déférents types de HMMs, le lecteur pourra consulter les ouvrage suivants [13] [14].

— π est la distribution initiale des états,$P(q_0 = j), \forall j \in [1, N]$, q_0 représente l'état initial du modèle HMM. Il ne peut émettre de vecteurs acoustiques.

En reconnaissance de la parole, des modèles de Markov gauche-droite d'ordre 1 sont le plus souvent utilisés du fait de l'aspect séquentiel du signal de la parole [15]. La figure 2.1 illustre un HMM à 3 états typiques utilisés en RAP pour la modélisation d'un phonème. Les états d'entrée et de sortie sont fournis pour faciliter la concatenation des modèles entre eux.

L'état de sortie d'un modèle de phonème peut être fusionné avec l'état d'entrée d'un autre modèle de Markov caché pour former un modèle composite. Ceci permet aux modèles

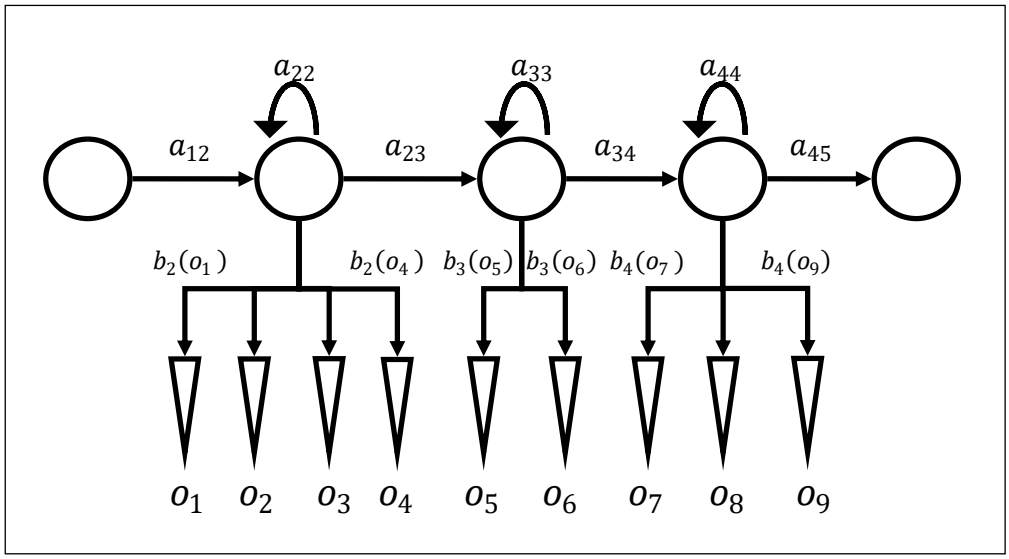

FIGURE 2.1 – Model de Markov caché à 3 états

de phonèmes d'être concaténés ensemble pour former les mots et ainsi les phrases.

On remarque que les seules transitions permises sont de type gauche-droite et ceci dans le but de mieux modéliser la contrainte temporelle de la parole. Un HMM est considéré comme un générateur de vecteurs acoustiques, c'est une machine à états finis qui change d'état à chaque unité de temps. Pour chaque unité de temps t, une fois arrivé à l'état q_j, un vecteur acoustique o_t est généré avec une densité de probabilité $b_j(o_t)$. De plus, la transition de l'état q_i à l'état q_j est probabiliste, sa probabilité est généralement notée a_{ij}. En pratique, c'est seulement la séquence d'observations : $O = O_1; O_2; ...; O_T$ qui est connue. La séquence d'états est non directement observable, d'où le nom de modèle de Markov caché [12].

2.2.2 Étapes fondamentales des HMMs

Soientλ un modèle de Markov caché et O une séquence d'observations acoustiques. La reconnaissance de cette séquence s'effectue en trouvant le modèle λ qui maximise la probabilité $P(\lambda|O)$ (probabilité qu'un modèle λ génère une séquence de vecteurs acoustiques O). Cette probabilité est aussi appelée probabilité a posteriori. Malheureusement, il n'est pas possible d'accéder directement à cette probabilité. Mais on peut calculer la probabilité qu'un modèle donné générera une certaine séquence de vecteurs acoustiques $P(O|\lambda)$ [12].

En utilisant la loi de Bayes, il est possible de lier ces deux probabilités par :

$$P(\lambda|O) = \frac{P(O|\lambda)P(\lambda)}{P(O)} \tag{2.4}$$

— $P(O|\lambda)$ est la vraisemblance de la séquence d'observations O étant donné le modèle λ.

— $P(\lambda)$ est la probabilité a priori du modèle.

— $P(O)$ est la probabilité a priori de la séquence des vecteurs acoustiques.

Pour une séquence d'observations connue $O = O_1; O_2; ...; O_T$. $P(O)$ peut être considéréeconstante, puisqu'elle est indépendante du modèle λ si les paramètres de ce dernier sont fixés.Ainsi maximiser $P(\lambda|O)$ revient à maximiser $P(O|\lambda)P(\lambda)$.

Pour cela, il faut résoudre les trois étapes fondamentales des HMMs suivants :

Évaluation : Étant donnés une séquence d'observations : $O = O_1; O_2; ...; O_T$ et le modèle $\lambda = (N, A, B, \pi)$, comment calculer efficacement $P(O|\lambda)$ la probabilité d'observer la séquence O sachant le modèle λ ?

Décodage : Étant donnés une séquence d'observations : $O = O_1; O_2; ...; O_T$ et le modèle $\lambda = (N, A, B, \pi)$, comment choisir la séquence d'états $Q = q_1; q_2; ...; q_T$ qui a le plus de chance d'émettre la séquence d'observations O ?

Apprentissage : Comment déterminer les paramètres du modèle $\lambda = (N, A, B, \pi)$ afin de maximiser $P(O|\lambda)$?

a Evaluation

Soient le modèle $\lambda = (N, A, B, \pi)$, $O = O_1; O_2; ...; O_T$ une séquence d'observations et $Q = q_1; q_2; ...; q_T$ une séquence d'états. La probabilité d'observer la séquence O pour une séquence d'états Q est [12] :

$$P(O|Q, \lambda) = b_{q_1}(o_1)b_{q_2}(o_2)b_{q_T}(o_T) \tag{2.5}$$

Or, la probabilité de la séquence Q peut s'écrire sous la forme suivante :

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} a_{q_T q_T} \tag{2.6}$$

La probabilité conjointe du chemin Q et des observations O est :

$$P(O, Q|\lambda) = P(Q|\lambda)P(O|Q, \lambda) \tag{2.7}$$

La probabilité de la séquence d'observations O sachant le modèle λ est obtenue par la sommation de $P(O, Q|\lambda)$ sur toutes les séquences d'états Q possibles. Ainsi la probabilité d'émission des observations est :

$$P(O|\lambda) = \sum_{Q} P(O, Q|\lambda) \tag{2.8}$$

$$P(O|\lambda) = \sum_{q_1;q_2;...;q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) ... a_{q_{T-1} q_T} b_{q_T}(o_T) \tag{2.9}$$

Pour une machine à N états, ce calcul direct nécessite $(2T-1) * NT$ multiplications et $N^T - 1$ additions, ce qui le rend trop complexe et impossible à implémenter. Il existe heureusement un algorithme rapide et efficace dit Avant-Arrière (Forward-Backward) qui donne une solution pour mener efficacement ce calcul.

Soit, la probabilité avant : $\alpha_t(i) = P(o_1, o_2, ..., o_t, q_t = i|\lambda)$, la probabilité d'observer la séquence $o_1, o_2, ..., o_t$ et d'être à l'état i à l'instant t sachant le modèle λ. Cette probabilité est calculée d'une manière récursive.

Algorithme avant

Initialisation :

$$\alpha_1(i) = \pi_i b_i(o_1) \quad 1 \leq i \leq N \tag{2.10}$$

Récurrence :

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(o_{t+1}), t \in \{1, 2, ..., T-1\} \quad 1 \leq j \leq N \tag{2.11}$$

Terminaison :

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{2.12}$$

Cette récursion dépend du fait que la probabilité d'être à l'état j au temps $t+1$ et d'observer o_{t+1} Peut être déduite en sommant les probabilités avant pour tous les états prédécesseurs de j pondérées par les probabilités de transition a_{ij}.

De la même manière, soit la probabilité arrière $\beta_t(j)$ définie par :

$$\beta_t(j) = P(o_{t+1}, o_{t+2}, ..., o_T|q_t = j, \lambda) \tag{2.13}$$

C'est la probabilité d'observer la séquence $o_{t+1}, o_{t+2}, ..., o_T$ sachant qu'on est à l'état i au temps t et qu'on a le modèle λ.

De la même façon cette probabilité est calculée d'une manière récursive :

Algorithme arrière

Initialisation :

$$\beta_T(i) = 1 \quad 1 \leq i \leq N \tag{2.14}$$

Récurrence :

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(o_{t+1}) \beta_{t+1}(j), \;\; t \in T-1, T-2, ..., 1 \quad 1 \leq i \leq N \tag{2.15}$$

Terminaison :

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \pi_i b_i(o_i) \beta_1(i) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i) \tag{2.16}$$

b Décodage

Étant donnés une séquence d'observations O, et un modèle $\lambda = N, A, B, \pi$, le problème de décodage revient à la recherche d'une séquence d'états "optimale". Cela peut-être fait de différentes façons. La difficulté réside dans la définition de la séquence d'états optimale. Donc, il faut choisir un critère parmi plusieurs critères d'optimalité. Par exemple, un critère envisageable pour répartir les vecteurs de la séquence d'observations sur les états de la chaîne, consiste à optimiser séparément chaque état q_t. Pour implémenter cette solution, une variable est définie par :

$$\gamma_t(i) = P\left(q_t = i | O, \lambda\right) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i) \beta_t(i)} = \alpha_t(i) \beta_t(i) P(O|\lambda) \tag{2.17}$$

$\gamma_t(i)$ est la probabilité d'être à l'état i au temps t, étant donnée l'observation O et le modèle λ.

L'état optimal à un instant t sera donc :

$$q_t = \arg_i \max \left[\gamma_t(i)\right] \tag{2.18}$$

Ce critère d'optimalité maximise le nombre d'états. Cependant, cette méthode peut aboutir à des erreurs. Par exemple, lorsque le modèle de Markov possède des probabilités de transitions égales à zéro, la séquence optimale obtenue pourrait en fait ne pas être une séquence d'états possibles puisque le critère considéré ne tient pas compte des probabilités des changements d'états.Une solution possible est de modifier le critère d'optimalité. On pourrait par exemple chercher la séquence d'états qui maximise les paires d'états (q_t, q_{t+1}) ou même les triplets d'états (q_t, q_{t+1}, q_{t+2}).

Si ces critères sont tout à fait adaptés à certaines applications, le critère le plus utilisé est celui qui cherche la meilleure séquence d'états globale (le meilleur chemin), c'est-à-dire qui maximise $P(Q|O,\lambda)$ ce qui revient à maximiser $P(Q,O|\lambda)$. Une technique formelle existe pour calculer ce chemin optimal, il s'agit de **l'algorithme de Viterbi**.

Pour trouver la meilleure séquence d'états $Q = q_1; q_2; ...; q_T$, connaissant une séquence d'observations $O = O_1; O_2; ...; O_T$, on a besoin de définir la quantité $\delta_t(i)$

$$\delta_t(i) = \max_{q_1,q_2,...,q_T} p\left(q_1, q_2, ..., q_t = i, O_1, O_2, ..., O_t|\lambda\right) \tag{2.19}$$

$\delta_t(i)$ est le meilleur résultat (probabilité la plus grande) selon un simple chemin ; ce chemin se compose des t premières observations et se termine dans l'état i. On peut déterminer les $\delta_t(i)$ de façon itérative. On a en effet :

$$\delta_{t+1}(i) = \max_{1\leq i\leq N} \left[\delta_{t+1}(i)a_{ij}\right] b_j(O_{t+1}) \tag{2.20}$$

Algorithme de Viterbi

Initialisation :

$$\begin{aligned} &\delta_1(i) = \pi_i b_i(o_1) \;\; 1 \leq i \leq N \\ &\psi_1(i) = 0 \end{aligned} \tag{2.21}$$

Récurrence :

$$\begin{aligned} &\delta_t(j) = \max_{1\leq i\leq N} \left[\delta_{t-1}(i)a_{ij}\right] b_j(O_t) \;\; 2 \leq t \leq T \;\; 1 \leq j \leq N \\ &\psi_t(j) = \arg_{1\leq j\leq N} \max \left[\delta_{t-1}(i)a_{ij}\right] \end{aligned} \tag{2.22}$$

Terminaison :

$$\begin{aligned} &P^* = \max_{1\leq j\leq N} \left[\delta_T(i)\right] \\ &\psi_T^* = \arg_{1\leq j\leq N} \max \left[\delta_T(i)\right] \end{aligned} \tag{2.23}$$

Recherche :

$$q_t^* = \psi_{t+1}(q_{t+1}^*) \;\; t = T-1, T-2, ..., 1 \tag{2.24}$$

Pour déterminer la séquence d'états, il est donc nécessaire de garder la trace de l'indice i qui a maximisé la formule précédente, et ceci pour tout t et tout j On réalise ceci par l'intermédiaire d'un tableau $\psi(j)$.

c Apprentissage

Le troisième problème consiste à trouver une méthode pour ajuster les paramètres du modèle $\lambda = (N, A, B, \pi)$ afin de maximiser la probabilité d'une séquence d'observations donnée, sachant le modèle λ. Ce problème n'a pas de solution analytique connue et il n'existe pas de technique optimale pour estimer les paramètres du modèle. On peut cependant choisir $\lambda = (N, A, B, \pi)$ de telle façon que $P(O|\lambda)$ soit localement maximale en utilisant une procédure itérative telle que la méthode de Baum-Welch ou la technique du gradient [16] [17]. Dans ce qui suit nous présentons une procédure itérative basée sur la technique de Baum-Welch.

Pour décrire comment ré-estimer les paramètres du HMM, on définit la probabilité $\xi_t(i, j)$ qui représente la probabilité d'être à l'état i au temps t et de faire une transition à l'état j au temps $t + 1$ étant donnée la séquence d'observations O et le modèle λ [12].

$$\xi_t(i, j) = P(q_t = i, q_{t+1} = j|O, \lambda) \tag{2.25}$$

D'après les définitions des probabilités avant et arrière, $\xi_t(i, j)$ peut s'écrire sous la forme suivante :

$$\xi_t(i, j) = \frac{\alpha_t(i)a_{ij}b_j(O_{t+1})\beta t + 1(j)}{P(O|\lambda)} \tag{2.26}$$

Nous avons défini, précédemment $\gamma_t(i)$ comme étant la probabilité d'être à l'état i au temps t, étant donnée l'observation O et le modèle λ. Ainsi nous pouvons relier $\gamma_t(i)$ à $\xi_t(i, j)$ par une sommation sur j, d'où la relation suivante :

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i, j) \tag{2.27}$$

L'algorithme de Baum-Welch estime les nouveaux paramètres de la chaîne de Markov cachée comme suit :

$$\overline{\pi}_i = \gamma_t(i) \;\; 1 \leq i \leq N \tag{2.28}$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \;\; 1 \leq i \leq N \;\; \& \;\; 1 \leq J \leq N \tag{2.29}$$

$$\overline{b}_j(k) = \frac{\sum_{t=1,o_t=K}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \quad 1 \leq j \leq N \tag{2.30}$$

La ré-estimation de π_i est la probabilité d'être à l'état i au temps $t = 1$. La formule de ré estimation de a_{ij} est le rapport du nombre de transitions de l'état i vers l'état j sur le nombre de transitions partant de l'état i. La ré-estimation de $b_i(k)$ est le rapport du nombre de fois d'être à l'état i en observant k sur le nombre de fois étant dans l'état i.

Nous avons défini le modèle courant $\lambda = (N, A, B, \pi)$, et nous l'avons utilisé pour recalculer ces variables, ainsi nous avons le modèle ré-estimé $\overline{\lambda} = (N, \overline{A}, \overline{B}, \overline{\pi})$. Nous pouvons ainsi affirmer l'une au l'autre de ces propositions :

- le modèle initial λ définit un point critique de la fonction de vraisemblance, dans ce cas $\overline{\lambda} = \lambda$.
- le modèle $\overline{\lambda}$ est meilleur que le modèle λ dans le sens où $P(O|\overline{\lambda}) > P(O|\lambda)$, donc la séquence d'observations O est plus probable avec le nouveau modèle $\overline{\lambda}$.

En se basant sur cette procédure, si nous utilisons itérativement le modèle $\overline{\lambda}$ à la place de λ et si nous répétons l'étape de la ré-estimation des paramètres. Nous pouvons alors améliorer la probabilité que O soit observé sachant le modèle jusqu'à atteindre un certain point limite. Le résultat final de la procédure de ré-estimation est appelé : l'estimation au maximum de vraisemblance du HMM (Maximum Likelihood Estimation : MLE). Il existe d'autres critères d'apprentissage, comme les critères MAP (Maximum A Posteriori) [18] ou MMI (Maximum Mutual Information) [19] [20], mais leur mise en œuvre est généralement plus difficile.

2.3 Alternative des réseaux de neurones aux HMMs

L'une des alternatives à l'utilisation d'HMMs en reconnaissance est le recours à des réseaux neuronaux.

Un réseau de neurones est une interconnexion de cellules simples (neurones). Chaque cellule possède plusieurs entrées et une sortie. Le signal de sortie peut être la somme pondérée (éventuellement seuillée) des signaux collectés en entrée Junqua and Haton, 1995).

L'utilisation de ces réseaux est largement répandue dans les domaines devant résoudre des problèmes de classification et de reconnaissance des formes (traitement d'image, de signature sonar, ...). Les réseaux de neurones (ANN, Artificial Neural Network) possèdent des propriétés très appréciées en RAP : - leur apprentissage est discriminant (ils permettent d'améliorer la reconnaissance d'une classe et simultanément de rejeter les autres classes).

– ils ne nécessitent pas d'hypothèses sur les propriétés statistiques des données en entrée (contrairement aux HMMs qui les modélisent par des PDFs).

Dans le cas des ANNs appliqués à la reconnaissance de la parole (mot ou tout autre unité acoustique), on utilisera le plus souvent des perceptrons multicouches. Plus généralement, on combinera le perceptron avec un algorithme d'alignement de type DTW, les distances locales utilisées lors de la DTW étant les sorties de l'ANN [21]. En plus de leur utilisation dans le problème de reconnaissance, les ANNs peuvent aussi servir à prétraiter le signal de parole et à extraire des paramètres discriminants. En effet, les coefficients de pondération des couches cachés d'un ANN forment une série de paramètres caractérisant l'entrée.

L'architecture d'un réseau de neurones à retard (TDNN, Time-Delay Neural Network) . La particularité d'un neurone de TDNN réside dans le fait que ses entrés à un instant sont constitués de données issues de l'instant présent mais aussi du passé et du futur. L'objectif est d'intégré des schémas temporels dans l'ensemble des données que doit généraliser le réseau de neurones. Un tel réseau combine la robustesse et le pouvoir discriminant des réseaux de neurones avec une architecture invariante par rapport au temps afin de former un identifieur de phonème très performant.

2.4 HTK (pour Hidden Markov Model Toolkit)

HTK (pour Hidden Markov Model Toolkit où "boîte à outils pour modèles de Markov cachés") est un ensemble de librairies et de programmes en langage C développés à l'Université de Cambridge sous la direction de S. Young à partir de 1989, pour faire de la reconnaissance de parole avec des HMM.

HTK permet de faire des systèmes de reconnaissance de parole à petit/moyen vocabulaire, avec une grammaire simple (par automate ou par bigramme). On peut aussi l'utiliser pour autre chose que de la parole (reconnaissance d'écriture manuscrite par HMM, par exemple). L'équipe de Cambridge a obtenu de très bons résultats en reconnaissance de parole sur les tâches " Wall Street Journal " et " Broadcast News ", lors des évaluations américaines DARPA/NIST des années 90 et en transcription de parole spontanée depuis. Un décodeur plus performant permettant l'utilisation de trigrammes a ensuite été intégré [22].

2.4.1 Présentation générale de HTK

HTK est constitué d'une vingtaine de programmes autonomes, dont les paramètres sont transmis par la ligne de commande ou par des variables d'environnement, et qui réalisent

chacun une étape de l'apprentissage ou de la reconnaissance. On peut les enchaîner automatiquement par des scripts (en C-shell, Perl, etc). Le manuel de HTK est très complet, avec un tutorial général, une présentation approfondie et un manuel de référence des différents modules. Le 1er chapitre constitue une bonne introduction à la reconnaissance de la parole par HMM. La documentation est aussi disponible au format HTML pour une consultation en ligne. L'illustration suivante, tirée du manuel, donne l'architecture globale du développement d'un système de reconnaissance :

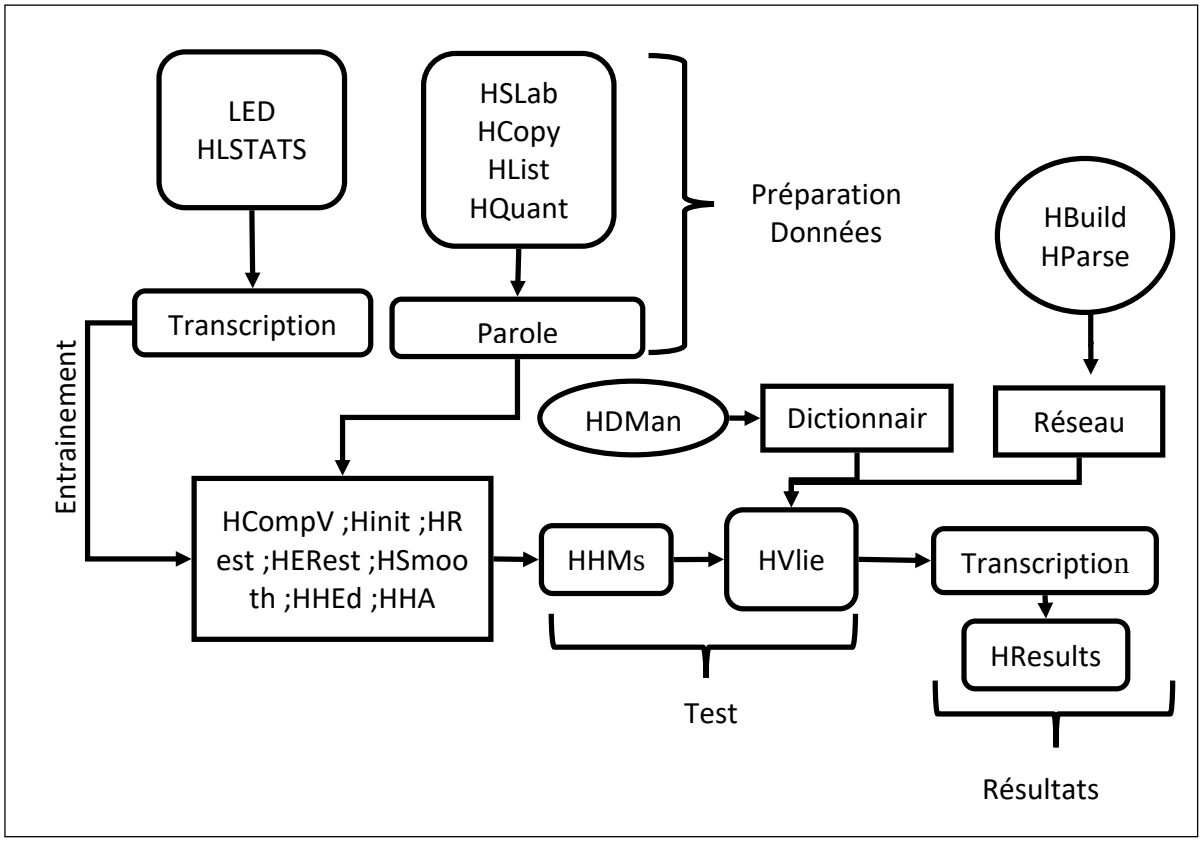

FIGURE 2.2 – Architecture d'un système de reconnaissance avec HTK

2.4.2 Description synthétique des outils de HTK

L'outil HTK nous permet d'implémenter plusieurs fonctionnarisé suivant l'application traitée [22] :

Manipulation du signal et des paramètres

HList : Affiche le contenu d'un fichier (signal ou paramètres).

HCopy : Copie et convertit des fichiers d'un format à un autre (en particulier calcule des paramètres - LPC, MFCC... - à partir d'un signal).

HQuant : Réalise la quantification vectorielle des paramètres d'apprentissage.

HCompV : Calcule la moyenne et la matrice de covariance de fichiers de paramètres (permet d'initialiser les modèles en l'absence de segmentation initiale)

Manipulation des étiquettes

HSLab : Edition interactive de fichiers d'étiquetage avec affichage du signal

HLEd : Edition automatisée des fichiers d'étiquetage (suppression, remplacement, fusion d'étiquettes…)

HLStats : Calcule diverses statistiques sur des fichiers d'étiquetage, en particulier des matrices bigrammes utilisées en reconnaissance

Manipulation de la grammaire et du dictionnaire

HBuild : Convertit une grammaire probabiliste dans un format connu du décodeur.

HParse : Convertit une grammaire déterministe dans un format connu du décodeur.

HSGen : Génère aléatoirement des phrases respectant une grammaire.

HDMan : Crée un dictionnaire de prononciation adapté à la tâche

Apprentissage et manipulation des modèles

HInit : Apprentissage initial d'un modèle en mode isolé par l'algorithme de Viterbi.

HRest : Re-apprentissage d'un modèle en mode isolé par l'algorithme de Baum- Welch (une itération)

HERest : Ré-apprentissage de tous les modèles en mode connecté (sur du signal non segmenté) par l'algorithme de Baum-Welch (une itération).

HHEd : Edition des modèles au moyen d'un script (duplication de modèles, augmentation du nombre de gaussiennes, création de triphones ...).

HEAdapt : Adaptation supervisée des modèles par les techniques MLLR ou MAP.

HSmooth : Lissage de modèles dépendants du contexte par l'algorithme "deleted interpolation" (utilisation très spécifique).

Reconnaissance et évaluation

HVite : Décodage par l'algorithme de Viterbi.

HResults : Calcul du taux d'erreur de la reconnaissance en comparaison avec le fichier de référence

2.5 Quantification vectorielle

La quantification vectorielle a été utilisée, en premier lieu, en reconnaissance de la parole.Soong et Rabiner [23] l'ont utilisé pour la reconnaissance des mots isolés afin de diminuer la complexité du système de reconnaissance de la parole basé sur la programmation dynamique. Aussi, Shore et Burton [24] utilisent la quantification vectorielle et reportent des bonnes performances pour la reconnaissance des mots isoles. Ces performances obtenues sont les principales motivations d'employer la quantification vectorielle dans l'identification automatique du locuteur [25].

La quantification vectorielle est un processus qui permet d'encoder un ensemble des vecteurs d'apprentissage (les vecteurs acoustiques) 'à l'aide d'un ensemble réduit de vecteurs représentatifs (formant un dictionnaire ou un codebook). Une telle représentation permet d'exploiter la corrélation existante entre les composantes d'un vecteur et de diminuer le volume d'information manipule [26]. L'objectif de cette méthode est de trouver le meilleur dictionnaire qui peut représenter l'information source.

Le quantificateur vectoriel est défini par un ensemble fini D de niveaux de quantification et une fonction associe chaque niveau de quantification 'a un vecteur de K échantillons produit par la source. L'ensemble D est appelé le dictionnaire.

Mathématiquement, on peut définir la quantification vectorielle comme une application de l'espace Q de dimension K dans R^K à un sous ensemble Y dans R^K. Soit $x = \{x_1, x_2, ..., x_k\}$ un vecteur de l'ensemble Q, la quantification de x revient à le représenter par un vecteur proche y_i d'un dictionnaire fini $y = \{y_1, y_2, ..., y_M\}$. Le dictionnaire Y est obtenu par partition de R^K en M classes C_i chacune représentée par son vecteur prototype ou Centroide y_i. Tout vecteur $x \in C_i$ sera représenté par y_i.

Cette substitution introduit une distorsion ou erreur de quantification, d'autant plus grande que la distance entre x et y est plus grande. Une façon de réduire l'erreur est d'augmenter la taille du dictionnaire, mais il faut trouver le bon compromis entre la taille du dictionnaire et l'erreur de quantification.

La réalisation d'un système de quantification vectorielle nécessite la résolution de deux problèmes :

Construction du dictionnaire elle est effectuée en générale dans un cadre statistique à partir d'un corpus d'apprentissage important de parole représentatif. Plusieurs algorithmes itératifs sont utilisés, notamment l'algorithme des K-moyennes et une variante,[27].

Accès au dictionnaire il doit être rapide est sûr. Une recherche exhaustive sur l'en-

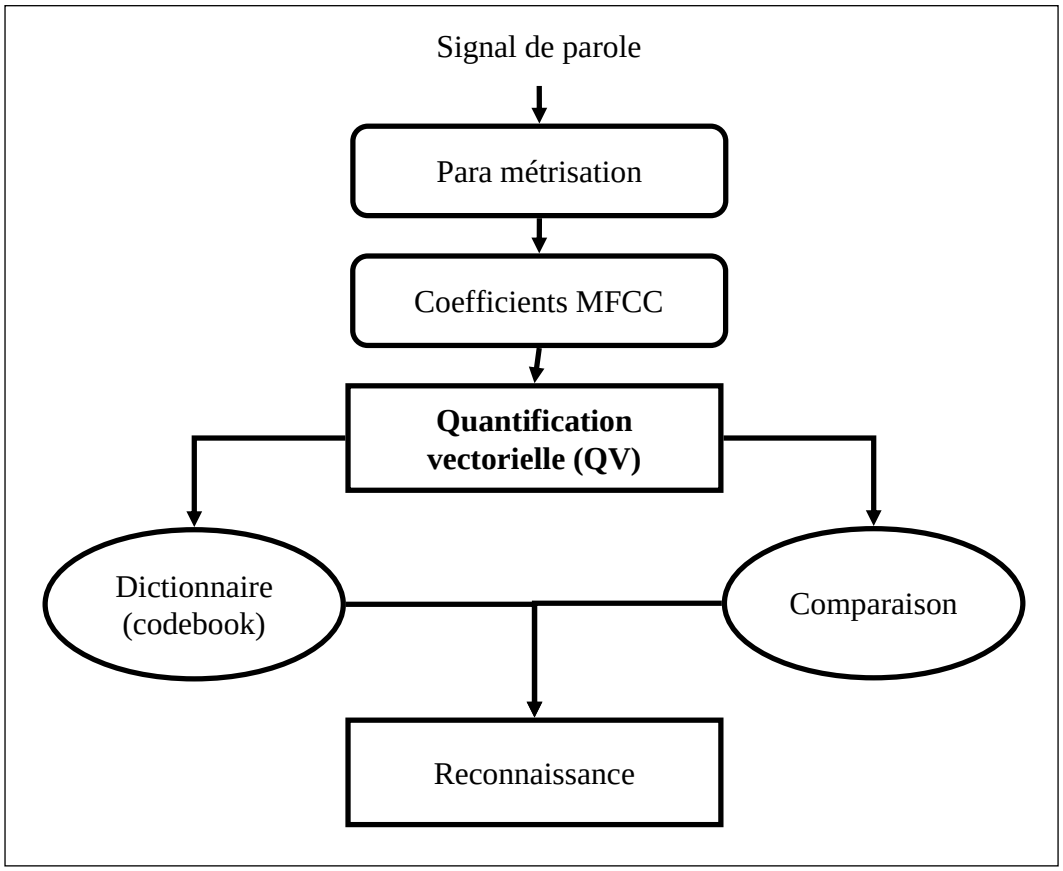

FIGURE 2.3 – Reconnaissance de la parole à base de Quantification vectorielle (QV)

semble du dictionnaire est en général incompatible avec un temps de repose acceptable. Diverses méthodes ont été proposées, notamment avec une définition des structures arborescentes pour les dictionnaires et l'utilisation de quantification hiérarchiques.

La quantification vectorielle est une méthode efficace pour stocker les informations spectrales de la parole. Chaque vecteur de paramètre est alors simplement représenté par le code du vecteur prototype qui lui est le plus proche. La perte en précision de la représentation, due à l'erreur de quantification, est compensée par le gain en volume de codage et en temps de traitement.

2.6 Implémentation du système

La detection de mots clés dans un signal de parole est une tâche très compliquées. Pour aboutir a nous objectifs, nous avons implémenter notre système en se basant sur les HMMs. La figure 2.4 montre l'architecture du système de detection de mots clés proposé.

Le signal de parole venu d'un appel téléphonique sera en premier lieu passé par un module d'extraction des paramètres acoustiques (MFCC ou LPC). Ce module transforme le signal

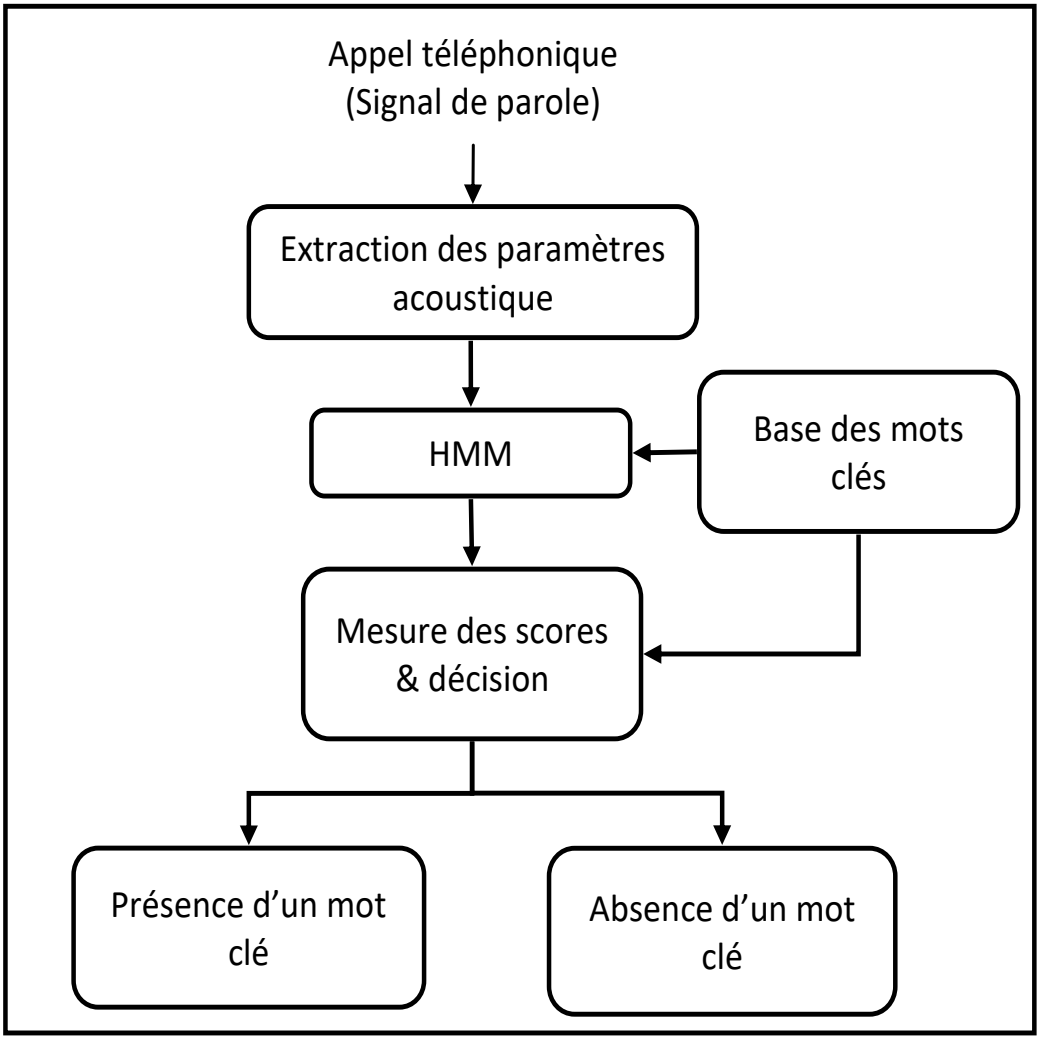

FIGURE 2.4 – Schéma fonctionnel du système de détection des mots clés

de parole vecteur en une matrice composée d'une suite de vecteurs de coefficients acoustiques. Le bloc de décodage et de décision utilise l'ensemble des coefficients acoustiques pour calculer les scores relatifs à chaque mot clés. Ceux-ci sont obtenus durant la phase d'apprentissage et sont stockés dans une base de références. La décision est faite selon le critère de maximum de vraisemblance, en employant l'algorithme de Viterbi.

2.6.1 Apprentissage des modèles

Le système proposé repose sur les HMMs. Le principe de ce système est de détecter la présence d'une menace (diffamation ou sécurité) dans un appel téléphonique. Les mots qui permettent de détecter les menaces cherchées sont enregistrés comme des HMMs dans une base de références. Ces derniers sont obtenus après une opération d'apprentissage.

L'apprentissage des différentes classes (représentés en tant que mot clés) est obtenu grâce à l'algorithme de Baum-Welsh. Pour estimer les paramètres donnant les meilleures

performances, plusieurs configurations possibles sont étudiées en fonction du nombre d'états et du type de paramètres acoustiques. La figure 2.5 montre la phase d'apprentissage par l'algorithme de Baum-Welsh.

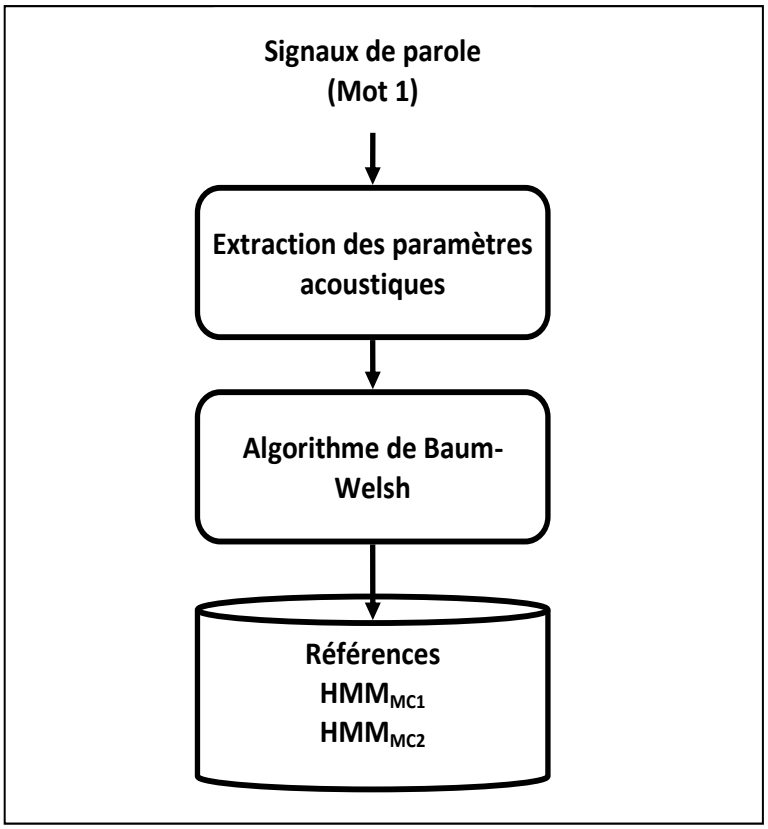

FIGURE 2.5 – Apprentissage du système proposé

On regroupe tous les signaux porteurs du mot clé cherché pour extraire un modèle spécifique à l'aide de l'algorithme de Baum-Welsh. Celui-ci est un processus itérative, sa vitesse de convergence dépend directement du type de paramètre et au modèle initial.

2.6.2 Détection des mots clés

La figure 2.6 montre le schéma fonctionnel pour détecter la présence d'une menace dans un signal de parole. Une fois qu'une personne prononce un mot indiquant un problème de diffamation ou de sécurité, le système utilise le signal de parole résultant et le transforme en une suite de vecteurs acoustiques. Ceux-ci sont utilisés pour calculer les scores relatifs à chaque mot clé enregistré dans la base de références. A l'aide de l'algorithme de Viterbi, et selon le critère de maximum de vraisemblance le système prend la décision de la présence du mot clé cherché.

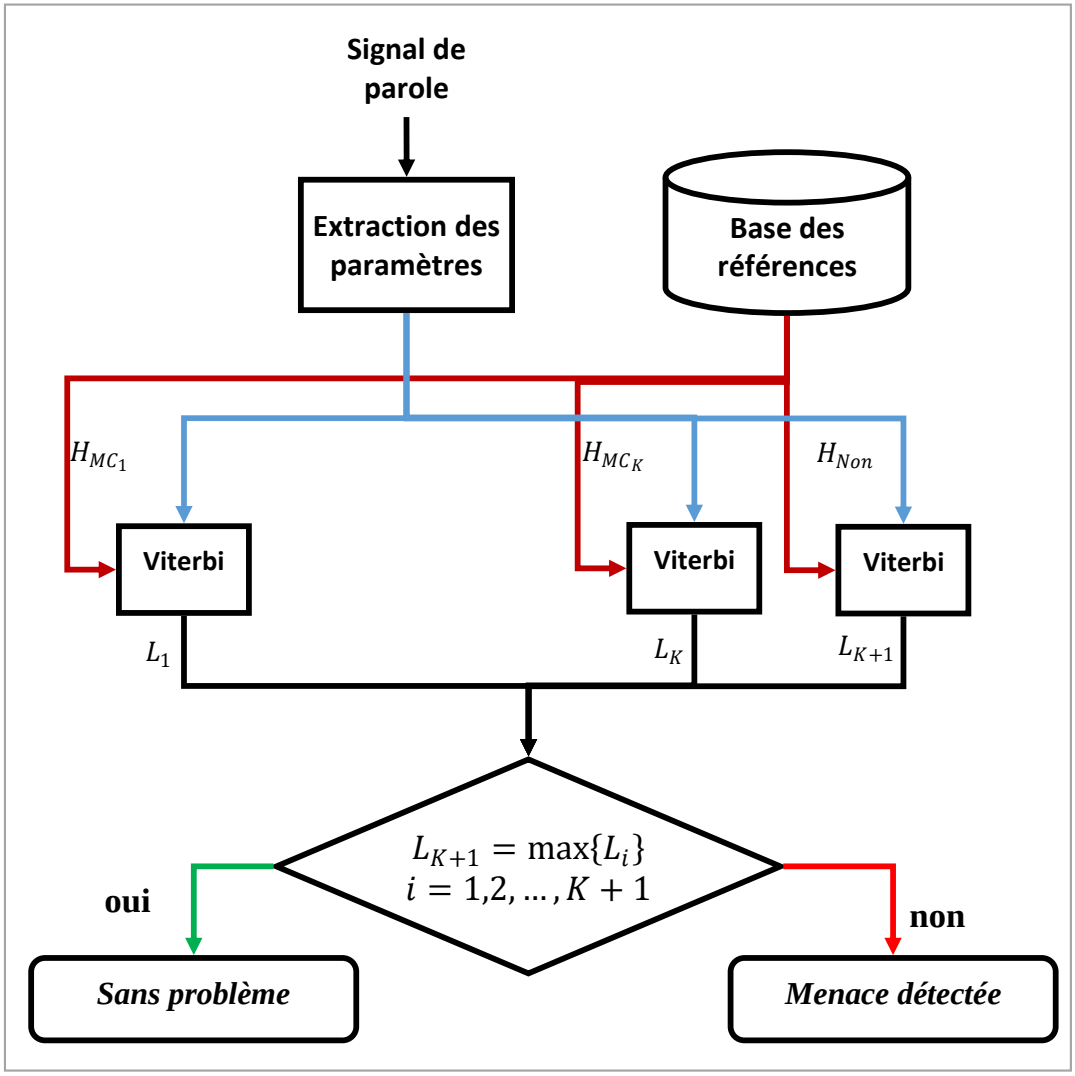

FIGURE 2.6 – Phase de détection du système proposé

2.7 Conclusion

Dans ce chapitre nous avons illustré les différents techniques utilisés pour un système de reconnaissance automatique de parole, nous avons commencé par une généralité sur les HMMs et son problèmes fondamentaux en suite l'alternatives des réseaux neurones aux HMMs ansi que le HTK , et nous avons fini par la quantification vectoriel. Le chapitre suivant est réservé pour l'évaluation expérimentale et la représentation es différents résultats de simulation.

Chapitre 3 Evaluation expérimentale et résultats

3.1 Introduction

Ce chapitre présente le contexte et l'évaluation expérimentale du système de detection des mots clés dans un signal de parole. L'objectif principale est de détecter la présence d'une menace dans un appel téléphonique. Cette menace peut être vu comme une diffamation ou d'un problème de sécurité.

Nous présentons également la procedure suivie pour la construction du corpus de parole ainsi que les différentes étapes pour l'extraction des paramètres acoustiques. Ces derniers sont l'essence des deux phases : l'apprentissage et l'identification. En outre, le chapitre resume tous les résultats obtenus pour les 10 mots clés utilisés. Ces résultats mesurent le taux de classification correct et le taux d'identification. Finalement, une étude des performances globales du système sera donné à la fin du chapitre.

3.2 Procédure expérimentale

Nous présentons ici les différentes étapes suivies avant d'implémenter notre système de la détection des mots clés. Nous allons montrer le corpus de parole, le choix des paramètres et le critère de performances. Vu la non disponibilité des resources pour ce type d'application et surtout pour la langue Arabe, nous avons construit un corpus personnel de mots clés.

3.2.1 Corpus de parole

Tout système de reconnaissance de parole nécessite la presence d'un corpus de parole. Ce corpus est généralement utilisé pour la phase d'apprentissage et en même temps pour la phase de test. Le corpus utilisé est composé de 300 phrases contenant les différents mots clés dans plusieurs contextes possibles.

Le corpus construit, contient 10 mots clés décomposés par deux domaines 5 mots pour

TABLE 3.1 – Mots utilisés pour la diffamation

Diffamation (التشهير)		
[Khaine] الخائن	Au debut Au milieu A la fin	الخائن موجود في بيته هل ذلك الخائن هنا اين ذلك الخائن
[Laiine] اللعين	Au debut Au milieu A la fin	لعين في طبعه وصفاته وصل ذلك اللعين الى منزله هل التقيت بذلك اللعين
[Fadjir] الفاجر	Au debut Au milieu A la fin	فاجر في قوله وعمله ذلك الفاجر سوف يعاقب هل اعتقل ذلك الفاجر
[Moutassalite] المتسلط	Au debut Au milieu A la fin	المتسلط في مكتبه اذا جاءكم ذلك المتسلط اطردوه اين اختفى ذلك المتسلط
[Hakire] الحقير	Au debut Au milieu A la fin	الحقير مر من هنا لقد فر الحقير من هنا من اين لك هذا أيها الحقير

le cas de la diffamation et le reste pour le cas de sécurité. Il occupe une durée totale environ 15 minutes. Chaque mot-clé est placé en 3 positions (au début, au milieu et à la fin) de la phrase. Les tableaux 3.1 et 3.2 resume le corpus de parole utilisé :

TABLE 3.2 – Mots utilisés pour la sécurité

Sécurité (الأمن)		
[silah] السلاح	Au debut Au milieu A la fin	السلاح الأبيض أداة ممنوعة دقة السلاح الروسي عالية القانون يمنع حمل السلاح
[tahribe] تهريب	Au debut Au milieu A la fin	تهريب البنزين في الحدود قضية تهريب المخدرات هل اعتقل بتهمة التهريب
[tazouire] تزوير	Au debut Au milieu A la fin	تزوير العملة الصعبة عملية تزوير الأموال عقوبة قاسية للمتورطين في التزوير
[ikhtiraq] الاختراق	Au debut Au milieu A la fin	اختراق الحساب لم ينجح عملية الاختراق متواصلة المكان جاهز للاختراق
[tafjir] التفجير	Au debut Au milieu A la fin	التفجير يكون بعد منتصف الليل عملية تفجير الثكنة ناجحة تنفيذ الخطة يكون بعد التفجير

Nous avons exploité l'outil PRAAT pour enregistrer les signaux de parole et en même temps pour les segmenter en mot. C'est un outil très important, qui est utilisé pour travailler sur des études phonologiques et phonétiques. Il permet de simplifier un peu la segmentation et l'annotation des données de parole.

Les enregistrements ont été faits avec une fréquence d'échantillonnage de $8000Hz$ pour les adaptés avec la bande passante téléphoniques $(300 - 3400Hz)$.

3.2.2 Choix des paramètres acoustiques

Le signal de parole ne peut directement être transformé en hypothèses de séquences de mots. L'extraction de ses paramètres est une étape importante puisqu'elle doit déterminer les caractéristiques pertinentes du signal. Il est nécessaire de découper le signal audio par trame, en prenant généralement une taille fixe définie aux alentours de 30 ms, afin de rendre le signal quasi-stationnaire. Un vecteur de paramètres est ensuite extrait pour chaque trame.

Dans le travail présent, nous avons utilisé deux types de paramètres : les MFCCs et les LPCs (12 coefficients pour chaque trame). Nous avons ajouté les dérivés premiers et secondaires des MFCCs, pour étudier l'effet de l'information dynamique incluse dans les paramètres acoustiques. Le tableau 3.3 resume la nature de paramétrisation utilisé dans notre travail :

TABLE 3.3 – Caractéristiques des paramètres acoustiques utilisés

Paramètres	Nombre de coefficients
$MFCC$	13 ($12MFCC + 1E$)
LPC	12
$MFCC^{'}$	39 ($12MFCC + 3E + 12\Delta + 12\Delta\Delta$)

a Extraction des coefficients MFCC

L'étude des coefficients MFCC du signal permet d'extraire des caractéristiques de celui-ci autour de la FFT et de la DCT, convertis sur une échelle de Mel. Il s'agit de la méthode la plus utilisée pour représenter un signal en reconnaissance de la parole, car elle est très robuste. Son principal avantage est que les coefficients obtenus sont faiblement corrélés entre eux. Le calcul des coefficients MFCC est réalisé de la manière suivante :

1. Préaccentuation du signal pour ressortir les hautes fréquences avec un filtre passe-haut de la forme :
$$H(z) = 1 - 0.9z^{-1} \tag{3.1}$$
2. Découpage du signal en fenêtre de 30 ms, toutes les 10 ms.
3. Application d'une fenêtre de Hamming sur ces portions de 30 ms.
4. Calcule de l'énergie de claque trame.

5. Application de la transformée de Fourrier sur chacune des portions, pour obtenir le spectre.
6. Faire passer le spectre resultant dans un Banc de filtres triangulaires espacés entre eux selon l'échelle mel.
7. Revenir au domaine temporel par l'application d'une DCT (Discrete Cosinus Transform) sur les portions, on obtient alors les coefficients cepstraux (MFCC).

b Extraction des coefficients LPC

Le codage LPC consiste à synthétiser des échantillons à partir d'un modèle de système de production vocal et d'excitation. Il s'agit d'une méthode très fréquemment utilisée pour l'analyse de la parole, l'encodage de la parole. Elle tire son nom du fait qu'elle permet de prédire une valeur future du signal à partir d'une combinaison des valeurs précédentes.

Pour calculer les coefficients LPC, nous avons effectué les étapes suivantes :

1. Découpage du signal en fenêtre de 30 ms, toutes les 10 ms.
2. Application d'une fenêtre de Hamming sur ces portions de 30 ms.
3. Calcul des coefficients LPC sur ces portions de 30 ms.

3.2.3 Critère de performance

Pour mesurer les performances du système proposé, nous avons utilisé deux critères de performances : le Taux de Classification Correct (TCC) et le Taux d'Identification (TId). Ces deux critères sont donnés par :

$$TCC = \frac{VP + VN}{VP + FP + VN + FN} * 100\% \tag{3.2}$$

$$TId = \frac{VP}{VP + VN} * 100\% \tag{3.3}$$

Avec :

— VP : Vrai Positive (présence du mot clé dans un signal qui le vraiment comporte).
— FP : Faux Positive (présence du mot clé dans un signal qui ne le comporte pas).
— VN : Vrai Négative (absence du mot clé dans un signal qui ne le comporte pas).
— FN : Faux Négative (absence du mot clé dans un signal qui le vraiment comporte).

3.3 Etude des performances du système proposé

Comme vu au précèdent, les performances du système de détection de mots clés sont analysés en calculant le taux d'identification et le taux de classification correct. Trois types de paramètre sont utilisés : $MFCC + E$, $MFCC' = MFCC + E + D + DD$ et LPC.

3.3.1 Cas de la diffamation

Dans cette section nous allons étudier les performance du système de détection pour le cas de la diffamation. Cind mots clés sont choisi pour cette classe : [Fadjir], [Hakire], [Khaine], [Laiine] et [Moutassalite].

a Mot clé [Fadjir]

La figure 3.1 montre le taux de classification correct et le taux d'identification pour le mot [Fadjir] en fonction de nombre d'états pour les trois types de paramètres. Ces résultats montrent que l'augmentation de nombre d'état de 3 à 7 dégrade les performances du système soit pour le TCC ou bien pour le TId. En outre, nous remarquons que le système montre un meilleur TCC de 85.71% obtenu avec une configuration de 3 états en utilisant les paramètres $MFCC'$. La même configuration présent un excelent TId de 100% mais e utilisant les LPC.

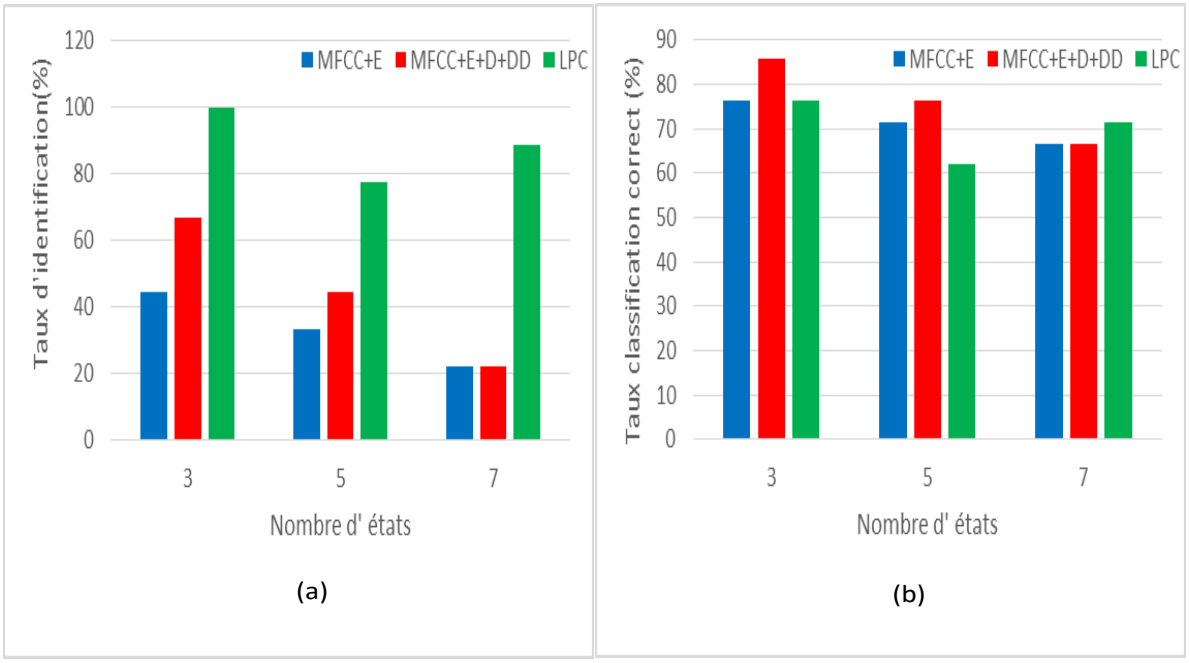

FIGURE 3.1 – Mot clé [Fadjir] (a) : taux d'identification (b) : taux de classification correct

Les résultats obtenus sont résumés au tableau 3.4.

TABLE 3.4 – Performance du système pour le mot clé [Fadjir]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	44.44	33.33	22.22
TCC (%)	76.19	71.43	66.67
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	66.67	44.44	22.22
TCC (%)	85.71	76.19	66.67
LPC			
Nombre d'états	3	5	7
TId (%)	100	77.78	88.89
TCC (%)	76.19	61.90	71.43

b Mot clé [Hakire]

La figure 2.6 illustre les performances du système pour le mot clé [Hakire]. Elle représente le taux d'identification et le taux classification correct pour les trois configurations et les trois types de paramètres acoustiques. Généralement le système donne des résultats importants en terme de TId pour toutes les configurations étudiées avec valeur de 100%. EN outre, ce système montre aussi un TCC élevé de 95.65% en utilisant les $MFCC$.

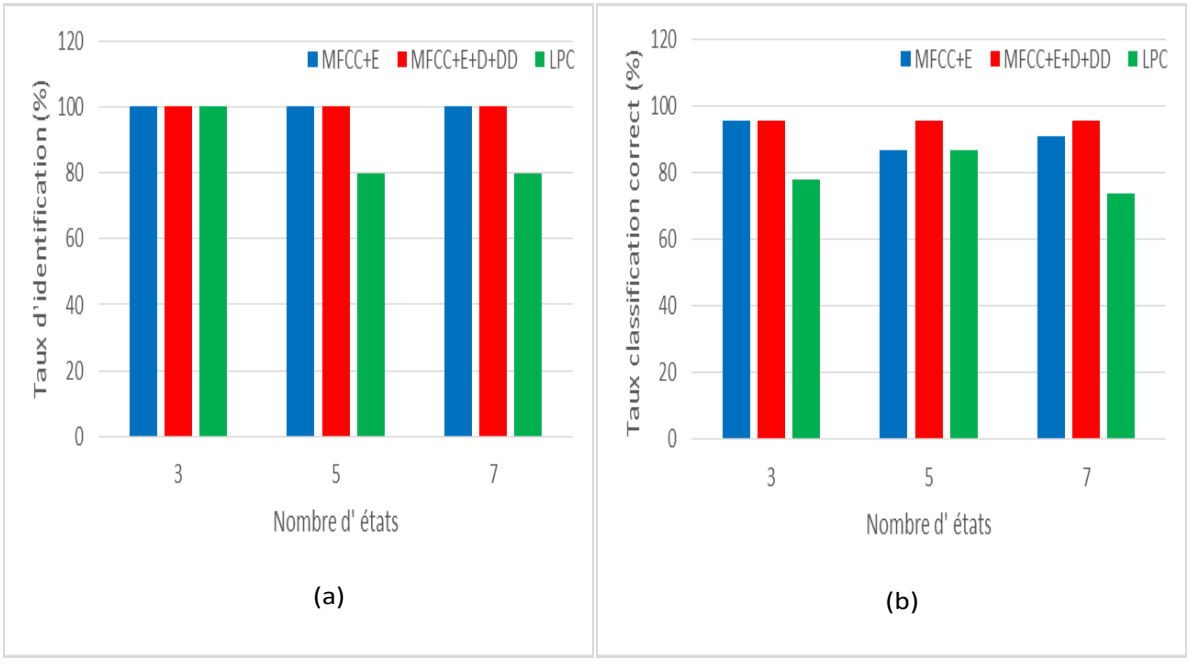

FIGURE 3.2 – Mot clé [Hakire] (a) : taux d'identification (b) : taux de classification correct

Le tableau 3.5 resume les performances de ce système.

TABLE 3.5 – Performance du système pour le mot clé [Hakire]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	100	100	100
TCC (%)	95.65	86.96	91.30
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	100	100	100
TCC (%)	95.65	95.65	95.65
LPC			
Nombre d'états	3	5	7
TId (%)	100	80.00	80.00
TCC (%)	78.26	86.96	73.91

c Mot clé [Khaine]

La figure 3.3 représente l'évaluation des performances du système pour le mot clé [Khaine]. Plusieurs remarques peuvent être tirés à partir de ces résultats. Un modèle de 3 états avec les parcmètres $MFCC$ est suffisant pour obtenir un TId complete de 100% et un TTC maximale de 89.47%. En outre ces performances peuvent être obtenus en utilisant les $MFCC'$ avec le même modèle.

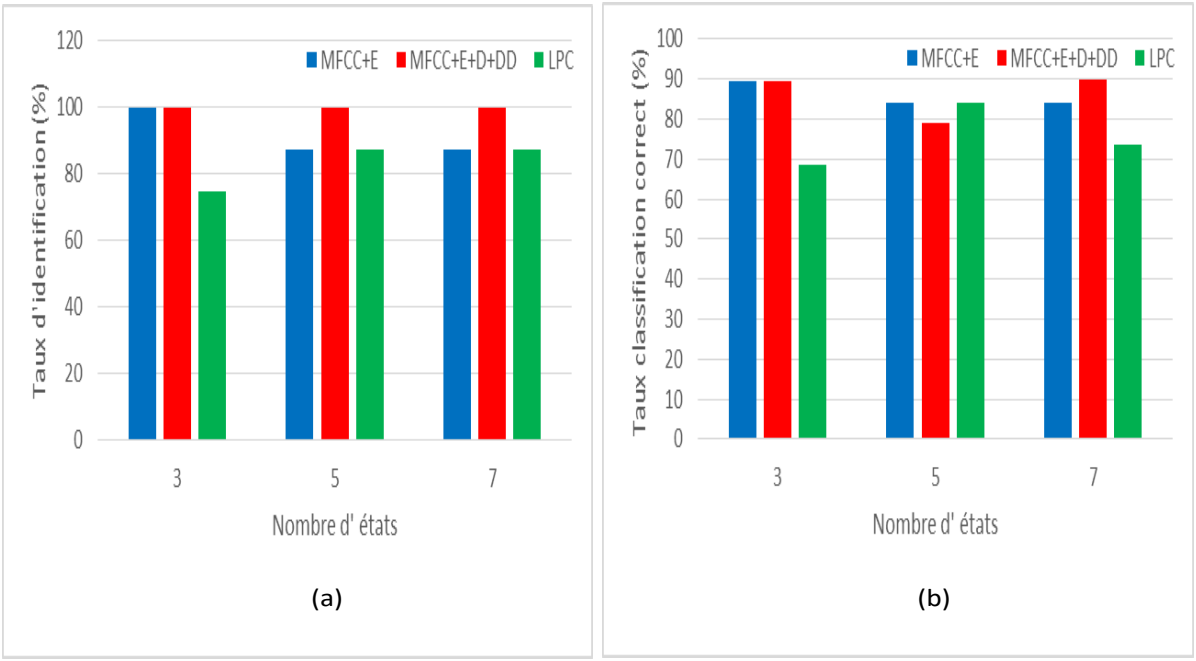

FIGURE 3.3 – Mot clé [Khaine] (a) : taux d'identification (b) : taux de classification correct

Les résultats obtenus sont montrés dans le tableau 3.6.

TABLE 3.6 – Performance du système pour le mot clé [Khaine]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	100	87.50	87.50
TCC (%)	89.47	84.21	84.21
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	100	100	100
TCC (%)	89.47	78.95	89.47
LPC			
Nombre d'états	3	5	7
TId (%)	75.00	87.50	87.50
TCC (%)	68.42	84.21	73.68

d Mot clé [Laiine]

La figure 3.4 représente l'évaluation des performances du système pour le mot clé [Laiine] en utilisant ces différents paramètres pour les trois configurations possibles. Les résultats montrent d'un modèle de 3 états avec les paramètres $MFCC'$ donne des performances excellentes soit en terme de TCC ou bien en terme de TId. En outre l'utilisation des LPC à la place des MFCC dégrade énormément les performances de ce système pour ce mot clé.

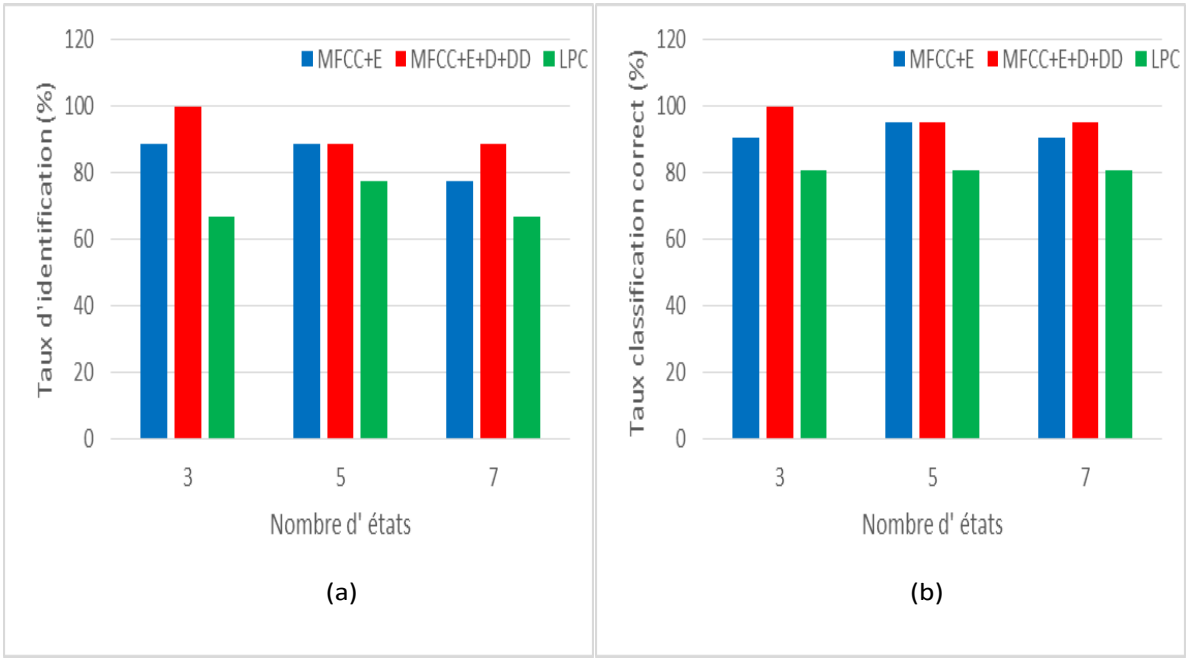

FIGURE 3.4 – Mot clé [Laiine] (a) : taux d'identification (b) : taux de classification correct

Les résultats obtenus sont résumés dans le tableau 3.7.

TABLE 3.7 – Performance du système pour le mot clé [Laiine]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	88.89	88.89	77.78
TCC (%)	90.48	95.24	90.48
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	100	88.89	88.89
TCC (%)	100	95.24	95.24
LPC			
Nombre d'états	3	5	7
TId (%)	66.67	77.78	66.67
TCC (%)	80.95	80.95	80.95

e Mot clé [Moutassalite]

La figure 3.5 montre les résultats obtenus du système proposé pour le mot clé [moutassalite]. Trois configurations sont étudiées en fonction du nombre d'états avec les trois types de paramètres utilisés. Nous remarquons que le système donne des performances adéquates pour toutes les configurations possibles. En plus, le meilleur TCC est obtenu à l'aide d'un modèle de 5 états employant les $MFCC'$ avec une valeur de 95.45%. Le modèle même montre un TId de 90%.

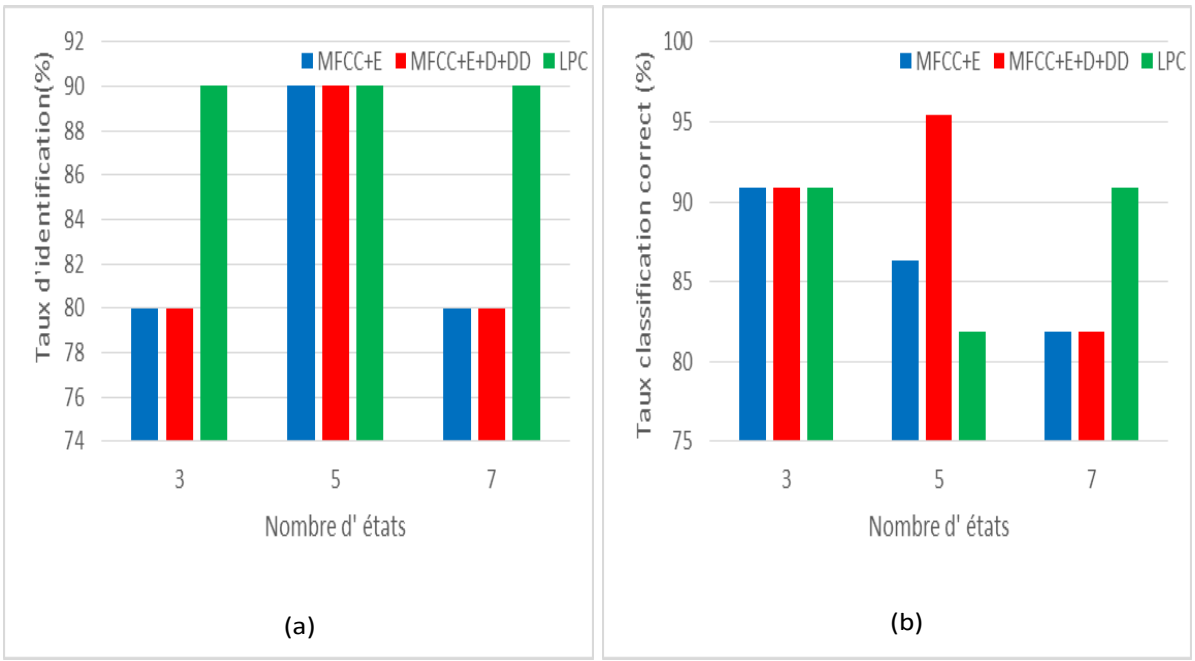

FIGURE 3.5 – Mot clé [moutassalite] (a) : taux d'identification (b) : taux de classification correct

Le tableau 3.8 resume tous ces résultats obtenus pour le ce pot clé.

TABLE 3.8 – Performance du système pour le mot clé [moutassalite]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	80.00	90.00	80.00
TCC (%)	90.91	86.36	81.82
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	80.00	90.00	80.00
TCC (%)	90.91	95.45	81.82
LPC			
Nombre d'états	3	5	7
TId (%)	90.00	90.00	90.00
TCC (%)	90.91	81.82	90.91

3.3.2 Cas de la sécurité

Les mots clés choisis pour la sécurité sont donnés au tableau 3.2.

a Mot clé [silah]

La figure 3.6 montre les résultats obtenus des performances du système pour le mot clé [silah] en fonction défirent configurations possibles. On observe que toutes les configurations étudiées montrent des performances excellentes pour ce mot clé. D'où le TCC et le TId sont obtenus avec une valeur de 100%.

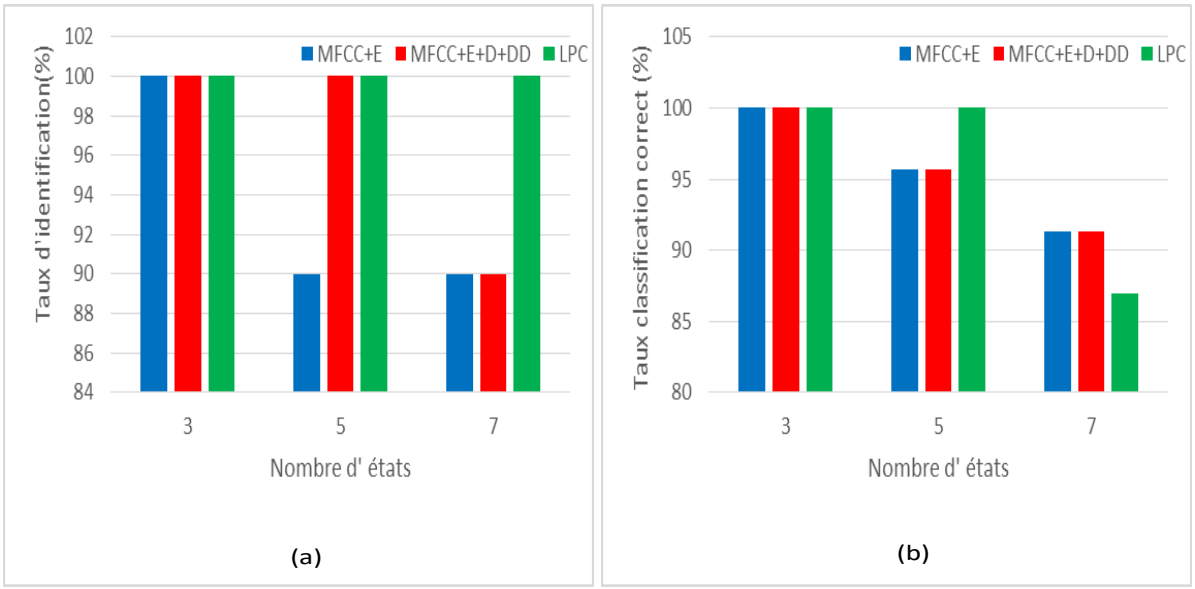

FIGURE 3.6 – Mot clé [silah] (a) : taux d'identification (b) : taux de classification correct

Le tableau 3.9 resume ces bons résultats en fonctions des différentes configurations.

TABLE 3.9 – Performance du système pour le mot clé [silah]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	100	90.00	90.00
TCC (%)	100	95.65	91.30
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	100	100	90.00
TCC (%)	100	95.65	91.30
LPC			
Nombre d'états	3	5	7
TId (%)	100	100	100
TCC (%)	100	100	86.96

b Mot clé [ikhtiraq]

La figure 3.7 résumé les résultats obtenus des performances du système pour le mot clé [ikhtiraq]'. Différentes configurations sont étudiées en fonction de nombre d'états et de type de paramètres. La figure montre qu'un HMM de 3 états avec les $MFCC$ est pour obtenir une meilleur performance avec un TCC de 100% et un TId de 95.24%. Par contre l'augmentation du nombre d'états et le changement de type de paramètres dégrade énormément les performances de ce système pour ce mot clé.

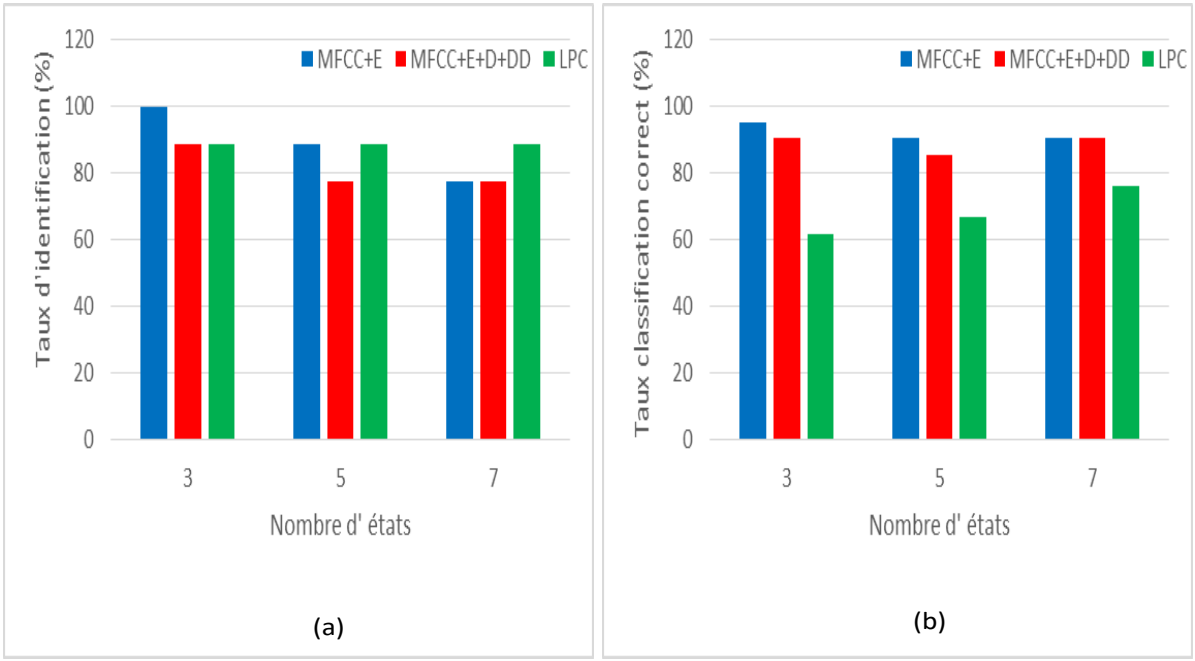

FIGURE 3.7 – Mot clé [ikhtiraq] (a) : taux d'identification (b) : taux de classification correct

Les résultats obtenus sont donnés au tableau 3.6.

TABLE 3.10 – Performance du système pour le mot clé [ikhtiraq]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	100	88.89	77.78
TCC (%)	95.24	90.48	90.48
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	88.89	77.78	77.78
TCC (%)	90.48	85.71	90.48
LPC			
Nombre d'états	3	5	7
TId (%)	88.89	88.89	88.89
TCC (%)	61.90	66.67	76.19

c Mot clé [tafjir]

La figure 3.8 représente les résultats obtenus des performances du système pour le mot clé [tafjir]. Nous calculons ces performances en terme de TCC et de TId en fonction du nombre d'états pour les trois types de paramètres acoustiques. D'une manière générale les paramètres $MFCC$ et sa variante $MFCC'$ montre de meilleurs performances avec un pourcentage de 100% pour les deux performances : le TCC et le TId. Par contre l'utilisation des LPC dégrade beaucoup les performances de ce système pour ce mot clé.

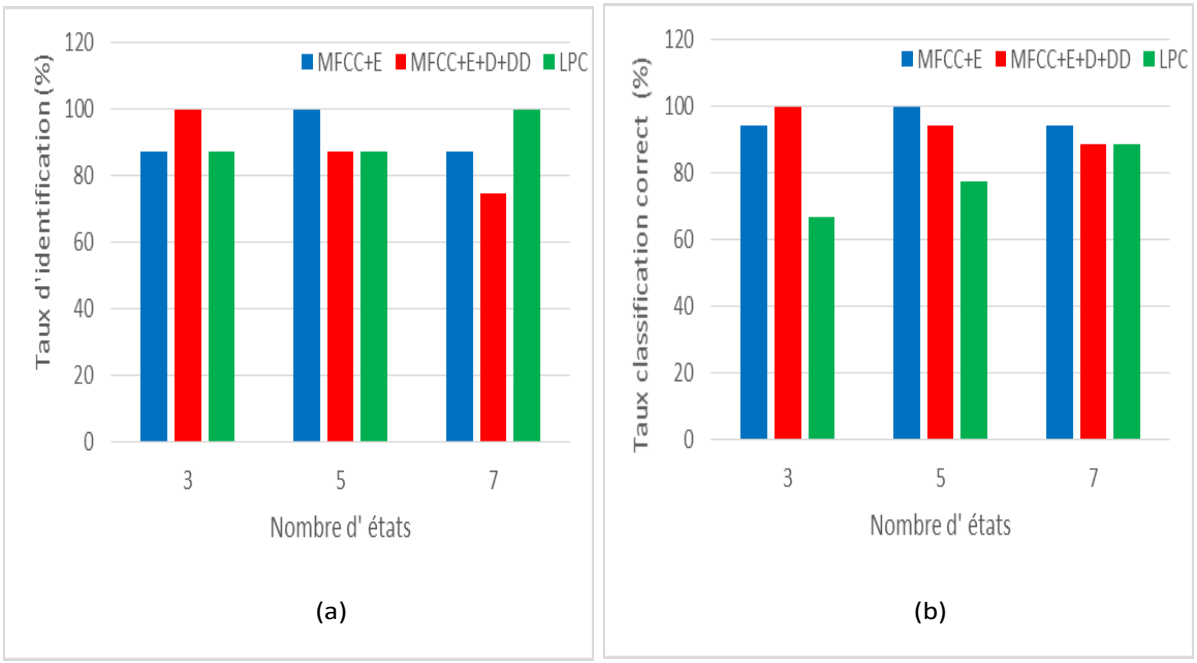

FIGURE 3.8 – Mot clé [tafjir] (a) : taux d'identification (b) : taux de classification correct

Le tableau 3.11 regroupe touts les résultats obtenus pour ce mot clé.

TABLE 3.11 – Performance du système pour le mot clé [tafjir]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	87.50	100	87.50
TCC (%)	94.44	100	94.44
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	100	87.50	75.00
TCC (%)	100	94.44	88.89
LPC			
Nombre d'états	3	5	7
TId (%)	87.50	87.50	100
TCC (%)	66.67	77.78	88.89

d Mot clé [tahribe]

La figure 3.9 montre l'évaluation des performances du système de détection du mot clé [tahribe]. Trois configurations du HMM sont étudiées en fonction du nombre d'états pour les trois types de paramètres acoustiques.

D'après cette figure, nous remarquons que l'augmentation du nombres d'états diminue le TCC et le TId. En outre un HMM de 3 états donne des performances excellentes avec une pourcentage complète pour les deux cas : le TCC et le TId. En outre, les $MFCC$ et les $MFCC'$ sont meilleurs que les LPC dans la plus part des configurations.

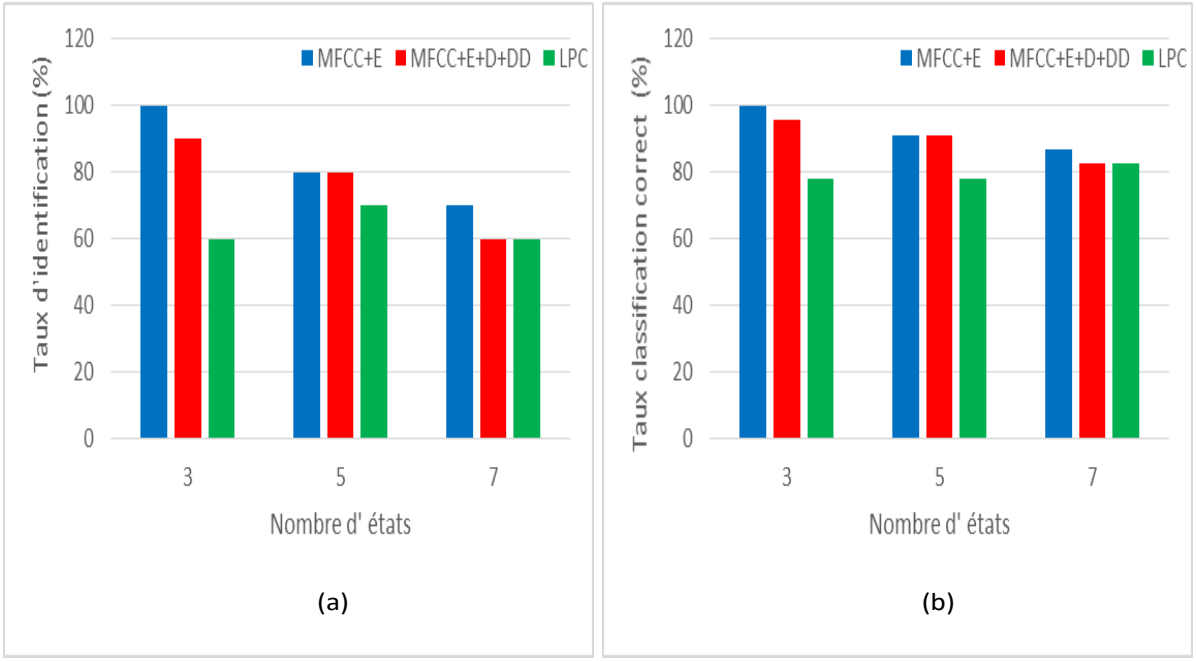

FIGURE 3.9 – Mot clé [tahribe] (a) : taux d'identification (b) : taux de classification correct

Les résultats obtenus sont résumés dans le tableau 3.12.

TABLE 3.12 – Performance du système pour le mot clé [tahribe]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	100	80.00	70.00
TCC (%)	100	91.30	86.96
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	90.00	80.00	60.00
TCC (%)	95.65	91.30	82.61
LPC			
Nombre d'états	3	5	7
TId (%)	60.00	70.00	70.00
TCC (%)	78.26	78.26	82.61

e Mot clé [tazouire]

La figure 3.10 illustre les performances du système pour le mot clé [tazouire]. Les résultats obtenus mesurent le taux de classification correct et le taux d'identification en fonction du nombre d'états. Nous avons utilisé trois types de paramètres pour représenter les coefficient acoustiques. Ces résultats montre l'efficacité du système proposé pour ce mot clé, avec une pourcentage complete soit pour le TCC soit pour le TId.

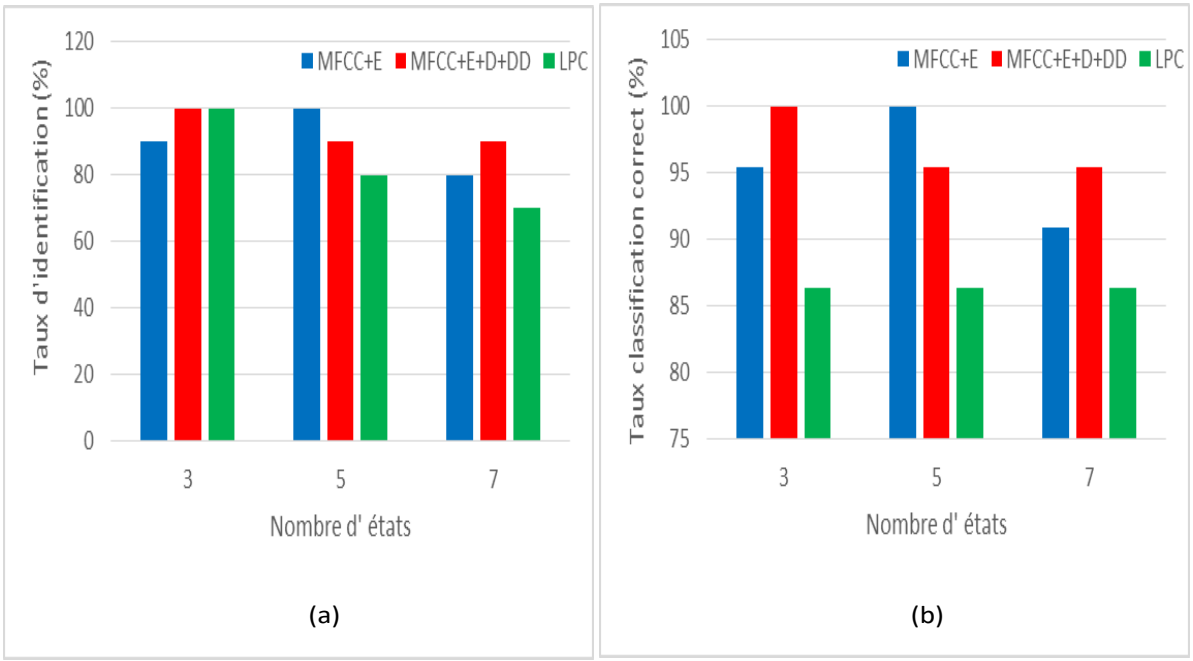

FIGURE 3.10 – Mot clé [tazouire] (a) : taux d'identification (b) : taux de classification correct

Le tableau 3.13 resume touts ces résultats pour le mot clé [tazouire].

TABLE 3.13 – Performance du système pour le mot clé [tazouire]

MFCC + E			
Nombre d'états	3	5	7
TId (%)	90.00	100	80.00
TCC (%)	95.45	100	90.91
MFCC+E+D+DD			
Nombre d'états	3	5	7
TId (%)	100	90.00	90.00
TCC (%)	100	95.45	95.45
LPC			
Nombre d'états	3	5	7
TId (%)	100	80.00	70.00
TCC (%)	86.36	86.36	86.36

3.3.3 Performances globales du système

Le tableau 3.14 illustre les meilleurs configurations possibles pour chaque mot clé selon le taux d'identification obtenu.

TABLE 3.14 – Performances globales du système proposé

Classe	Mot clé	Nombre d'états	Paramètres	TId $(\%)$
Diffamation	[Fadjir]	3	LPC	100
	[Hakire]	3	LPC	100
	[Khaine]	3	MFCC	100
	[Laiine]	3	MFCC'	100
	[Moutassalite]	3	LPC	90
Sécurité	[silah]	3	LPC	100
	[ikhtiraq]	3	MFCC	100
	[tafjir]	3	MFCC'	100
	[tahribe]	3	MFCC	100
	[tazouire]	3	LPC	100

3.4 Conclusion

Dans ce chapitre, nous avons étudié les performances de système de détection de mots clés dans un signal de parole. Nous avons tout d'abord présenté la procedure suivie pour l'implémentation de notre système concernant : la construction du corpus de parole, l'apprentissage du modèle et le processus de détection. Les performances du système sont étudiés pour touts les mot clés proposés. Ces résultats montrent l'efficacité des HMM pur ce type d'application.

Conclusion Générale

La détection de mots clés, le sujet de ce mémoire est sans doute une étape clé pour plusieurs applications interactives utilisant la parole comme moyen de communication. Elle consiste à améliorer les performances de telles applications en les aidant à détecter le sens des énoncés émis et ce en dévoilant seulement les mots porteur de sens pour l'application en question.

La liste des mots clés les plus significatifs pour l'application est déterminée dans une étape antérieure : dans notre travail, par exemple, nous avons défini un ensemble de 10 mots clés. Dont 5 mots clés représentent le problème de la diffamation et les autres pour le problème de la sécurité.

Ce travail s'inscrit dans le cadre de la reconnaissance automatique de mots isolés. L'objectif principal est de construire un système permettant la détection des menaces relatives à la diffamation ou à la sécurité dans un appel téléphonique. Le noyau principale du système en cours repose sur les modèles de Markov cachés avec une représentation cepstral et paramétrique des signaux de parole.

Pour aboutir à nos objectifs, nous avons étudié les performances de notre système pour les 10 mots sélectionnés. Cette étude est effectue en fonction du nombre d'états dans le modèle HMM, et le type de paramètres utilisés. L'implémentation du ce système est réalisé sous le logiciel Matlab. Le performances sont donnés en terme du taux de classification correct et du taux d'identification correct.

Nous avons montré qu'un HMM de 3 états permet de donner des pourcentages complètes pour la majorité des mots clés. Ce qui confirme l'efficacité du système proposé pour ce type d'application.

Comme perspective nous proposons de faire une étude linguistique sur les mots qui indiquent les différentes menaces comme la diffamation, les crimes, l'enlèvement d'enfants et la sécurité. En outre la construction d'un corpus de mots clés est une phase très importante.

Bibliographie

[1] Boite R. et Kunt M., *Traitement de la Parole*, Press Polytechniques Romandes, 1987.

[2] Calliope, *Traitementautomatique de la parole*, Masson, 1989.

[3] Koreman J., Andreeva B., et Strik H., *Acoustic parameters versus phonetic features in ASR*, In Proceedings of the International Congress of Phonetic Sciences, pages 549-553, 1999.

[4] Haton J. P., Pierrel J. M., Perennou G., Caelen J., et Gauvain J. L., *Reconnaissance automatique de la Parole*, Dunod, 1991.

[5] Deroo O., *Modèlesdépendants du contexteetméthodes de fusion de donnés à la reconnaissance de la parole par modèleshybrides HMM/MLP*,Thèse de Doctorat, Université Mons France, 1998.

[6] Lindgren N., 'Machine Recognition of Human Language Part I. Automatic Speech Recognition'. In : IEEE Spectrum, vol. 2, no. 3, pp. 114-136, 1965.

[7] Dreyfus-Graf J., *Sonograph and Sound Mechanics*, Journal of the Acoustical Society of America, vol. 22, no. 6, pp. 731-739, 1950.

[8] Davis K. H., Biddulph R., et Balashek S., *Automatic Recognition of Spoken Digits*', Journal of the Acoustical Society of America, vol. 24, no. 6, pp. 637-642, 1952.

[9] Benati N., *Détection des mots clés dans un flux de parole basée sur une approche perceptuelle*, Magister en iformatique, Univ Guelma, 2010.

[10] Richard G., *Eléments de Reconnaissance de la Parole pour PACT*, Extraits du polycopié de cours de l'UE SI340, Télécom Paris Tech, Paris, France, 2012.

[11] Fréjus A. A. Laleye, *Contributions à l'étude et à la reconnaissance automatique de la parole en Fongbe*, Thèse de Doctorat, Université d'Abomey-Calavi et Université du Littoral Côte Opale, France, 2016.

[12] Ben Ayed Y., *Détection de mots clés dans un flux de parole*, Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 2003.

[13] Rabiner L. et Juang B. H., *A tutorial on hiddenmarkovmodels and selected application in speech recognition*, Proceeding of IEEE, vol. 77, pp. 257-285, 1989.

[14] Roxane L., *Au sujet des algorithmes de recherche des systèmes de reconnaissance de la parole à grands vocabulaires*, Thèse de Doctorat, Computer Science Université McGill, Montréal,1995.

[15] Bakis R., *Continuous speech word recognition via centisecond acoustic states*, In 91^{st} Meeting of the Acoustical Society of America, 1976.

[16] Juang B. H., *Maximum likelihood estimation for mixture multivariate stochastic observations of markovchains*, ATT Technical Journal, vol. 64, pp. 1235-1246, 1985.

[17] Bahl L. R., Brown P. F., Suza P. V., et Mercer L. R., *Maximum mutual information estimation of hidden Markov model parameters for speech recognition*, In Proceeding of the International Conference on Acoustics, Speech and Signal Processing, pp. 49-52, 1986.

[18] Gauvain J. L. et Lee C., *Maximum a-posteriori estimation for multivariate gaussian mixture observations of markov chains*, IEEE Transactions on Speech and Audio Processing, vol. 2, no. 2, pp. 291-298, 1994.

[19] Cardin R., Normandin Y., et De Mori R., *High performance connected digit recognition using maximal mutual information estimation*, In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 533-536, 1991.

[20] Kapadia S., Valtchev V., et Young S. J., *MMI training for continuousphoneme recognition on the TIMIT database*, In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 491-494, 1993.

[21] Bourlard H., D'hoore B., et Boite J. M., *Optimizing recognition and rejection performance in wordspotting system*, In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pp. 373-376, 1994.

[22] 26-28, Barras C., *Reconnaissance et interactive vocale*, Cours Master 2, Univ Paris-Saclay, 2018.

[23] Pan, K. C., Soong, F. K. and Rabiner, L. R., 'A Vector Quantization Based Preprocessor for Speaker-Independent Isolted Word Recognition', submitted for publication.

[24] Shore, J. E. et Burton, D. K, *Discrete Utterance Speech Recognition Without Time Alignment*, IEEE Trans. On Inform. Theory, Vol. IT-24, No. 4, pp. 473-491, July 1983.

[25] Soong F.K., Rosenberg A.E., Rabiner L.R., B. H. Zhuang, *A vector quantization approach to speaker identification*, In : Proceedings of International Conference on Acoustics, Speech, and Signal Processing, Tampa, FL, pp. 387-390, 1985.

[26] William H. E, *A new Vector Quantization Clustering Algorithm*, IEEE Tran, Vol. 37, No. 10, pp. 1568-1575, October 1989.

[27] Linde Y., Buzo A., and Gray R.M., *An algorithm for vector quantizer desing*, IEEE Transactions on communications, vol. 28, pp 702-710, 1980.