REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de 08 Mai 45Guelma
Faculté des sciences et de l'Ingénierie

Département de l'informatique

Ecole Doctorale de l'Informatique de l'Est (EDIEST)
Option : Intelligence Artificielle

Mémoire présenté pour l'obtention du diplôme de
Magister en Informatique

Détection de mots clefs dans un flux de parole basée sur une approche perceptuelle

Présenté par Benati Nadia

Devant le Jury :

Président :	Pr Seridi Hamid	Professeur	Université 8 Mai 45 Guelma
Rapporteur :	Dr Bahi-Abidet Halima	M.C	Université Badji Mokhtar Annaba
Examinateurs :	Dr Khadir Tarek	M.C	Université Badji Mokhtar Annaba
	Dr Souici-Meslati Labiba	M.C	Université Badji Mokhtar Annaba

Remerciements

.áãÚâ ÇÐå ãÇãÊ? ÉæÞâÈíäãÅæ íäÞý íÐâ ìâÚÊæ åâíÈÓ?Ç ÑßÔ

Je tiens à remercier mon amie Dr Bahi-Abidet Halima, mon encadreur, maitre de conférences à l'université Badji Mokhtar Annaba, pour ses précieux conseils, ses encouragements, sa gentillesse, sa disponibilité et surtout sa patience avec moi. Qu'elle trouve ici l'expression de ma profonde gratitude.

Je remercie chaleureusement Pr Seridi Hamid, Professeur à l'université 8 mai 45 guelma, d'avoir accepté de présider mon jury. Je le remercie également d'avoir toujours été à notre écoute ainsi que pour tous les efforts qu'il a fournis pour que notre formation se déroule dans des conditions favorables.

Un remerciement particulier au Dr Souici-Meslati Labiba, maitre de conférences à l'université Badji Mokhtar Annaba, d'avoir accepté d'être l'examinatrice de mon travail ainsi que pour ses encouragements.

Je remercie Dr Khadir Tarek, maitre de conférences à l'université Badji Mokhtar Annaba, d'avoir accepté d'être l'examinateur de ce mémoire et pour toute l'aide qu'il m'a fournie.

Je remercie particulièrement Pr Ziou Djemel de l'université de Sherbrooke, pour ses conseils, son aide précieuse et sa disponibilité.

Mes remerciements vont également à tous les enseignants de notre année théoriques pour leurs encouragements et leur compréhension.

Un grand merci à tous mes collègues de l'Ecole Doctorale en Informatique de l'Est spécialement Houda, Samia, Naima, Anissa, Samira, Nouzha, Nourredine, Ali, Mohamed Salah, Rachid et Akram pour leurs encouragements.

Je remercie mes collègues de travail et mes responsables pour leur aide et leur compréhension.

Je remercie vivement tous les membres de ma famille de m'avoir encouragée à étudier et de m'avoir soutenue durant ces années.

Je dis merci à toute personne ayant contribué de prés ou de loin à l'élaboration de ce mémoire.

Dédicaces

A mes parents
A tous ceux qui me sont chers

Résumé

Les premières approches de la recherche d'information dans un contenu parlé ont d'abord utilisé des techniques similaires à celle développées pour les documents textuels, appliquées à la transcription automatique du flux de parole. Un système de reconnaissance automatique de la parole est appliqué à un ensemble de fichiers audio et génère une transcription de la parole. Cette transcription est alors indexée et une recherche est alors lancée par un système de recherche. Le résultat fourni pour une requête est une liste de pointeurs vers des fichiers audio.

En détection de mots clés, le but est de reconnaître les mots clés dans un flux de parole continue, indépendamment du locuteur. La détection de mots clés résout quelques problèmes liés à la reconnaissance de la parole continue comme les hésitations, les faux départs, les phrases grammaticalement incorrectes, les phrases tronquées, etc.

Dans ce mémoire, nous avons proposé une approche qui peut être considérée comme une autre voie que la transcription textuelle des discours en vue de l'exploitation des bases de données audio, en offrant une méthode qui se fonde principalement sur le signal acoustique avant sa transcription phonétique.

Nous avons proposé une méthode qui tente de localiser un mot clef dans un flux audio en se basant sur la détection de certaine de ses caractéristiques. Ainsi, la détection d'un mot clef est réalisée par la détection d'un son particulier dans le mot c'est-à-dire d'une unité infra-lexicale, et à partir de cet « ilot de confiance », on peut chercher dans l'intervalle considéré le mot à reconnaître. Lorsque ces unités sont détectées dans le discours un appariement avec les modèles des mots clefs qui contiennent ces unités est effectué.

L'évaluation de cette approche est faite sur la langue arabe, langue où peu de travaux ont été menés aussi bien en reconnaissance de la parole qu'en recherche d'information.

Abstract

The first approaches in spoken document retrieval used techniques similar to those developed for text documents, applied to the automatic transcription of speech flow. A system for automatic speech recognition is applied to a set of audio files and generates a transcript of the speech. The transcript is then indexed and the search is then launched by a search system. The result for a given query is a list of pointers to audio files.

In keyword spotting, the goal is to recognize keywords in a stream of continuous speech, regardless of the speaker. Detection keyword solves some problems related to recognition of continuous speech as hesitations, false starts, the sentences grammatically incorrect, truncated phrases, etc

In this work, we proposed an approach that can be seen as another way to text transcription of speeches for the use of audio databases, providing a method that is based primarily on the acoustic signal prior to its phonetic transcription.

We proposed a method that attempts to locate a keyword in an audio stream, based on the detection of some of its characteristics. Thus, detection of a keyword is performed by detecting a particular sound in the word ie a sub-lexical unit, and from this "island of reliability" can be search in the interval considered the word to recognize. When these units are detected in the speech matching models with keywords that contain these units is made.

The evaluation of this approach is made on the Arabic language, the language in which little work has been done both in speech recognition in information retrieval.

ملخص

f lDžKzuŕĂ ŐkfllDžkĐžžKGžkDžPk[k łdĂT !fKGŽĂdžfLJz łchufŐ5JNNGdžfýĂD
ìáÚ ã?ßá ìáÚ íÆÞâá 2ƵkfDŽLjsŀƵõ.ã?ßá ÞŸÊ! ŐJŽŽkfq[LJf lDž GfjŐƵkĂ ÜŐJŽGĂ!f
TpkDžLdžĂGfK[ƵnƷDžő łP[LjlJNjŲGŽP!fLJz łŐP łP[LjpĂŐĂ łdĂT !fKGŽžfLdž łzĂlždž
KGŽžfIDžz KƵTdžfLdžÉãÆP íå äÚã Ñ;ÓÉÊ? ÉÌÊa.DŽLJf JNjsŽŐLJz łchułŽžz
. łdĂT !f

ã?ßá ääs5JkŐ5 łő[ƵC!fKGŽžf lDž 2ƵkfĂNj2fnJfÜő[ƵC!fKGŽžf!fLJz 2T+!fŐ5
2ƵkfGu łDžkžfý+Gdžfýoŏłő[ƵC!fKGŽžf!fLJz 2T+!f.DžžkžfLJz aULJfU'ƶuÜý[ƵĂkžf
¡ÉÑæÊa áãÌ ¡;æÍäÉÍíÍÕ ÑÛ ã;ßÍ?;æ ¡ÉÝÆÒ Ê;ÍÌÈ Üƶdžkžfƒƶ2kfýdžã?ßá ìáÚ íÆÞâá
. llRƵŐĞĂ

ýnDLdžKGGPDŽŐ[LJfq[LJfLJz ăƵPDÐžĂ lƵGžfLJužŐGnljjGj2kfÜƶ+kžf lJNjÕ5
ÎÓâá áÈÉêæÕâ Ê;Ñ;Ô?; ìáÚ ;ÓÍÕ ãæÞÊ íêá ÉÞÑØ Ñý¼êæ ¡ łdĂT !fKGGbfƒƶZĂpá?ÛÊÕ
.ŐUÐžf

LdžU'ƶuLJz 2T+!fƒGžZGÜŐkĂT !fƵGkfŐ5 łő[ƵC łdžž ÍDž ƵĂvz!fýĂGŕk łƵŐ Gj2kf
ïDłdžž!fŐ5Ldžž KĂT 2T+ s5ŐLJzLJĂ+k łő[ƵC łdžž 2T+ÜƒNĂ. łdžž!flJNjƒ JE[P
.GŽDž 2ƵkfƒƵdžf łdžžDžƒƶdžfýGŽdžfŐ5 łchufLJuž łlVfƵrĺ lJNjLdžĂ. łŐkƒƵdžKƒƶĂ
ÍDž ïĂvrkŐkf łő[ƵC!fKGŽžf!fUłŽŽjydžNžhGŽžDžŐÜ;GŽP!fŐ5KƒƶĂ!flJNj2T+k GŽhzĂ
. KƒƶĂ!flJNj

ÍDž ŐJŽŽkf2ƵkfŐ5CĂ\ƒƵŐ+ý;k[kDžŐkf łƷžfÜłŐƵz!f łƷžfDžPk[GDžkNhljfJNjDŽŐk
. KGĂždžfLJz łchufŐ5Ãã?ßá

Tables des matières

Partie 0 :Introduction 10
1 Introduction générale 12
2 Position du problème 13
3 Solution proposée 14
4 Contribution 14
5 Organisation du mémoire 14
Chapitre 1 : La recherche d'information audio 16
1 Introduction 16
2 Systèmes de recherche d'information 16
3 Composants principaux d'un modèle de RI 18
4 La recherche d'information audio 19
5 Conclusion 22
Chapitre2 : Reconnaissance Automatique de la Parole 23
1 Introduction 23
2 Reconnaissance automatique de la parole 24
2.1 Le signal de la parole 24
2.2 Caractéristiques d'un système de RAP 25
2.3 Différentes étapes de la RAP 26
3 Analyse acoustique 26
3.1 La mise en forme du signal de parole 26
3.2 Calcul des coefficients 29
3.2.l Analyse spectrale (Coefficients MFCC) 29
3.2.2 Analyse par prédiction linéaire (coefficients LPCC) 32
4 Décodage des informations acoustiques 32
4.1 Approche analytique 33
4.2 Approche globale 33
4.3 Approche statistique 34
5 Les modèles de Markov cachés 34
5.1 Définition d'un HMM 34
5.2 Mise en œuvre 36

5.2.1 Hypothèses simplificatrices 36
5.2.2 Topologie du modèle 37
5.2.3 Apprentissage 39
5.2.4 Décodage 39
5.3 Limitation des HMM 40
6 Conclusion 41
Chapitre 3 : La détection de mots clés 42
1 Introduction 42
2 Les systèmes de détections de mots clés 42
3 Modèle poubelle 43
4 Mesure de confiance 44
4.1 Programmation dynamique 45
4.2 Algorithme de Viterbi et de Baum-Welch 46
4.3 Utilisation des traces d'alignements 47
4.4 Apprentissage discriminant 47
4.5 Seuil sur les scores de reconnaissance 48
4.6 Méthodes d'adaptation 48
4.7 Connaissances acoustiques et linguistiques 49
4.8 Réseaux de neurones 50
4.9 Transformations et algorithmes 51
5 Les applications 51
6 Conclusion 52
Chapitre 4 : Détection des mots clefs par les ilots de confiance 54
1 Introduction 54
2 Traitement des fichiers son 55
2.1 Analyse du signal 55
2.2 Calcul des coefficients 56
2.3 La quantification vectorielle 57
3 Définition et caractérisation des mots clefs 58
3.1 Caractérisation acoustique 58
3.2 Construction des modèles de référence : 59
3.3 Topologie du modèle 60
4 La détection 61
5 Conclusion 62

Chapitre 5 : Evaluation de l'approche proposée.....63
Présentation du chapitre.....63
1 Introduction.....63
2 Extraction des caractéristiques.....64
2. 1 Echantillonnage.....64
2. 2 Pré-accentuation.....65
2. 3 Fenêtrage.....66
2. 4 Fenêtrage de Hamming.....67
2. 5 Analyse MFCC.....68
3 La quantification vectorielle.....69
3. 2 Etablissement des classes par la méthode de LLOYD généralisée (K-means method).....69
3.3 Construction du dictionnaire.....71
4 Caractérisation d'un mot clef.....72
5 Recherche d'un mot clef.....73
5.1 Corpus utilisé.....73
5.3 La détection.....73
6 Evaluation de performance.....74
6.1 Corpus de test.....74
6. 2 Taux de détection.....74
6. 3 Taux d'erreur.....75
7 La reconnaissance.....75
8 Conclusion.....76
Conclusion et perspectives.....77
Références bibliographiques.....79

Liste des figures

Figure 1.1 : Schéma général d'un modèle de recherche d'information……...17
Figure 2.1 : Représentation d'un signal de parole……...22
Figure 2.2 : Les différentes tâches d'un système RAP……24
Figure 2.3 : L'échantillonnage et l'interpolation d'un signal……..25
Figure 2.4 : La mise en forme d'un signal……...…27
Figure 2.5 : Répartition fréquentielle des filtres triangulaires suivant l'échelle de Mel…….29
Figure 2.6 : Calcul des coefficients MFCC…….30
Figure 2.7 : Exemple d'un modèle de markov caché à trois états……...34
Figure 2.8 : HMM gauche-droite à 3 états……...36
Figure 3.1 : Description du système de détection de mots clef basé sur l'utilisation d'un réseau de mots clef et de mots poubelles……..…42
Figure 4.1 : Etapes de construction du dictionnaire……..…56
Figure 4.2 : Les étapes de caractérisation d'un mot clef……..57
Figure 4.3 : Transcription en phonèmes du chiffre cinq……..58
Figure 4.4 : Modèle du chiffre cinq……..…58
Figure 4.5 : Etapes du processus de détection…….59
Figure 5.1 : Signal de la phrase narrative1……...63

Listes des tables

Table 5.1 : Variation de la fréquence selon l'application….……...62
Table 5.2 : Taux de détection…….73

Partie 0 : Introduction

1 Introduction générale

L'exploitation des outils de recherche d'information dans le flux audio est indispensable devant la masse importante du signal acoustique disponible. Beaucoup des travaux ont réussi à indexer, rechercher et parcourir le signal acoustique, en utilisant les dernières avancées du domaine de reconnaissance automatique de la parole. Les dernière avancées technologiques ont permis de rendre l'acoustique moins « opaque», c'est-à-dire, fournir de la perspicacité dans le contenu d'un fichier audio, et peut-être des voies de l'utiliser autrement qu'un bloc homogène de données numériques.

La recherche d'information est basée sur la recherche des occurrences d'un mot (lisible par la machine) dans des documents texte. Dans ce contexte, il existe plusieurs algorithmes puissants qui permettent de calculer la fréquence d'apparition, ces méthodes utilisent des approches probabilistes, des approches statistiques ou des approches hybrides. Ainsi, ces résultats sont classés selon leurs importances et degrés de pertinences. Tandis que, ces techniques sont loin d'être appliquées pour le signal audio, vu que ces méthodes sont basées sur la notion d'appariement des mots de la requête dans les documents cibles. Et comme, on n'a pas des représentations normalisées des mots acoustiques (représentation phonétique), la tâche de recherche devient extrêmement compliquée. De plus, la tâche se complique, si on considère que le signal acoustique contient paroles, musiques, silences et éventuels bruits. Un autre problème se pose lors de l'exploitation des documents est sa linéarité, pour cela il est nécessaire d'intégrer les métaphores qui correspondent au notions : mot, phrase, paragraphe…etc.

Initialement issue du domaine de la reconnaissance de la parole, la détection de mots clefs dans un flux de parole s'impose comme une orientation nouvelle des recherches pour laquelle des méthodes et des techniques ne cessent de se développer. Elle permet de résoudre certains

problèmes tels que les hésitations, les faux départs, les phrases grammaticalement incorrectes…etc. De plus elle permet une indépendance vis-à-vis du locuteur et du canal de transmission.

Les recherches réalisées dans le domaine de la détection de mots clefs dans un flux de parole visent généralement à faciliter l'interaction entre l'homme et la machine en détectant les mots les plus intéressants pour l'interprétation sémantique de ce qui a été prononcé. Les applications possibles dans ce domaine sont nombreuses, nous pouvons citer, les opérateurs de service automatique, les systèmes de routage téléphonique ou de classification des documents parlés etc.

Les systèmes de reconnaissance équipés par des processus de détection de mots clés permettent aux utilisateurs de parler librement et naturellement, sans besoin de s'exprimer avec un format rigide. Un système de reconnaissance idéal doit accepter la parole spontanée et générer une transcription précise de la phrase d'entrée en temps réel.

2 Position du problème

Bien que les premiers travaux aient été basés sur la détection de début et de la fin d'un mot clef, et ensuite sur la comparaison du segment obtenu avec un modèle de référence. Les recherches dans ce domaine se sont plutôt orienté vers la construction de système de détection à l'image des systèmes classiques de reconnaissance de la parole, où des modèles spécifiques sont entraînés pour reconnaître les mots clefs et des approches particulières ne cessent de se développer pour la modélisation des mots non clefs ; modèles qu'on appelle : modèles poubelles.

Nous remarquerons alors que les performances du système de détection sont fortement reliées à la qualité de modélisation des mots poubelles or cette vision des choses peut sembler paradoxale, vu que c'est de la qualité de modélisation des mots clefs que doit dépendre le système final. Toutefois, la tendance actuelle dans les systèmes de détection de mots clefs dans les documents audio demeure centrée autour des modèles clefs et des modèles poubelles.

Notre souci dans ce travail est de présenter une alternative à cette vision des choses en s'inspirant d'une philosophie perceptuelle ; où dans un flux de parole, on détecte un mot particulier que l'on recherche au travers de certaines de ces caractéristiques. Cette première

alerte peut aboutir où échouer en prospectant plus loin le voisinage de l'élément déclencheur. En effet, au niveau phonétique, il y'a des sons particuliers qui nous interpellent dans un mot, ainsi si nous recherche le mot [mukbilun], c'est le son [k] qui semble guider notre recherche, transposons cette approche au niveau acoustique, et essayons de caractériser les mots clefs de manière à définir des caractéristiques discriminantes qui serviront à les localiser.

3 Solution proposée

La solution proposée se veut une alternative à la fois aux modèles poubelles et à la détection du début et la fin d'un mot. Notre proposition inspirée de la démarche perceptuelle lorsque l'on entend un discours, nous pouvons lui trouver un parallèle dans la transcription des discours, où des mots particuliers servent de zones de confiance « ilots de confiance », ces mots sont des blocs de base autour desquels des trous seront progressivement remplis.

Toutefois dans notre proposition ce sont des unités infra-lexicales qui font office d'ilots de confiance. Lorsque ces unités sont détectées dans le discours un appariement avec les modèles des mots clefs qui contiennent ces unités est effectué.

4 Contribution

Les technologies actuelles en recherche d'information audio se basent énormément sur les avancées du domaine de la recherche d'information textuelle. Notre travail se veut un enrichissement dans ce contexte, en proposant une autre voie que la transcription textuelle des discours en vue de l'exploitation des bases de données audio, en offrant une méthode qui se fonde principalement sur le signal acoustique avant sa transcription phonétique. En plus, en trouvant une similitude de notre proposition avec l'approche par « ilots de confiance » utilisée en transcription nous pourvoyons notre proposition d'une assise théorique déjà établie.

5 Organisation du mémoire

Le mémoire est organisé en deux parties comme suit :

Une première partie intitulée état de l'art est constituée des trois premiers chapitres. Nous avons tout d'abord présenté les systèmes de recherche d'information, leurs composants, les modèles existants dans la littérature, ainsi que les méthodes utilisées pour l'information audio.

Le deuxième chapitre se divise en deux parties. La première est consacrée au signal de la parole et les caractéristiques d'un système de reconnaissance automatique de la parole ainsi les différentes étapes d'une reconnaissance automatique de la parole(RAP).

Dans la deuxième partie du chapitre, un outil très utilisé dans la reconnaissance automatique de la parole; ce sont les modèles de markov cachés (HMM). Une définition d'un HMM est donnée, ensuite sont décrites les différentes étapes de la mise en œuvre d'un système RAP avec les HMM.

Le troisième chapitre est dédié à la présentation des méthodes utilisées dans la construction des systèmes de détections des mots clés. Ensuite, sont exposées les techniques de constructions des modèles poubelles, ainsi que les différentes mesures de confiances. Enfin quelques applications dans le domaine de la détection des mots clés sont exposées

La deuxième partie est consacrée à la détection des mots clefs dans un flux de parole par les ilots de confiance. La méthodologie adoptée ainsi que l'approche proposée sont exposées. Les différentes phases des expérimentations menées afin de concrétiser cette approche sont aussi décrites. Cette partie comprend les chapitres quatre et cinq.

Le quatrième chapitre décrit les différentes phases de la méthodologie adoptée. Il présente la préparation des fichiers son, ensuite la définition et la caractérisation des mots clefs et enfin la phase qui décrit l'approche proposée pour la détection de mots clefs.

Le cinquième chapitre présente le plan d'évaluation de l'approche proposée, les choix d'implémentation effectués pour le mettre en œuvre ainsi que les résultats obtenus.

Nous terminons ce mémoire par une conclusion générale dans laquelle nous rappellerons les résultats obtenus et nous évoquerons les perspectives à ce mémoire.

Chapitre 1 :
La recherche d'information audio

1 Introduction

La grande quantité d'informations disponibles sous des différents formats (numériques, textes, images…) produites par des sources d'informations distribuées est un des résultats de l'apparition et le développement des Nouvelles Technologies de l'Information et de la Communication (NTIC) et du réseau Internet. Il s'agit aujourd'hui de développer des outils automatisés de traitement et de recherche d'information pour gérer ces masses d'informations et permettre à un utilisateur d'accéder à l'information dont il a besoin.

La Recherche d'Information (RI) est une branche en informatique qui s'intéresse à l'acquisition, l'organisation, le stockage et la recherche des informations. Elle propose des outils, appelés Systèmes de Recherche d'Information (SRI), dont l'objectif est de capitaliser un volume important d'information et d'offrir des moyens permettant de localiser les informations pertinentes relatives à un besoin en information d'un utilisateur exprimé à travers une requête.

Dans ce chapitre, nous présentons les systèmes de recherche d'information, leur composants, les modèles de SRI existants dans la littératures et nous mettrons le point sur les méthodes utilisées pour l'information audio.

2 Systèmes de recherche d'information

Un Système de Recherche d'Information (SRI) est un système informatique qui facilite l'accès à un ensemble de documents (corpus), pour permettre de retrouver ceux dont le contenu correspond le mieux à un besoin d'information d'un utilisateur [Géry 02].

Les SRI se basent donc sur trois concepts essentiels : le document, le besoin et la correspondance. Les documents, atomiques et indépendants, doivent correspondre avec la représentation du besoin de l'utilisateur : la requête.

On distingue les deux tâches principales d'un SRI :
L'indexation automatique, c'est-à-dire l'extraction et le stockage du contenu sémantique des documents du corpus. Cette phase nécessite un modèle de représentation de ce contenu sémantique, appelé *modèle de documents*.

L'interrogation, c'est-à-dire l'expression du besoin d'information de l'utilisateur sous la forme d'une requête, la recherche dans le corpus, et la présentation des résultats. Cette phase nécessite un modèle de représentation du besoin de l'utilisateur, appelé *modèle de requête*, ainsi qu'une *fonction de correspondance* qui doit évaluer la pertinence des documents par rapport à la requête. Dans ce contexte, pour évaluer la pertinence d'un document vis à vis d'un besoin exprimé par une requête, la majorité des SRI mesure un score de pertinence entre un document, représenté par ses mots clés et la requête. Cette mesure de pertinence, élément fondamental de tout SRI, est souvent formalisée à travers la notion de modèle de recherche d'information.

La réponse du système est un ensemble de références à des documents qui obtiennent une valeur de correspondance élevée. Cet ensemble est généralement présenté sous la forme d'une liste ordonnée suivant la valeur de correspondance. D'autres paramètres peuvent être considérés : le nombre de documents à présenter, la quantité d'information à fournir pour chaque document, le format de présentation utilisé, etc.

Éventuellement, le système propose un mécanisme de retour de pertinence ("relevance feedback" [Rocchio71], [Salton et al.90]) : quand le résultat de la recherche n'est pas satisfaisant, le système reformule automatiquement la requête, en fonction du jugement de pertinence de l'utilisateur sur les documents déjà proposés. Il y a alors un apprentissage par étapes du besoin de l'utilisateur. Cette méthode permet à l'utilisateur de s'abstraire en partie des problèmes de formulation : syntaxe et complexité de la requête.

3 Composants principaux d'un modèle de RI

La problématique d'un SRI est de modéliser ce processus de recherche d'information. Pour cela, on distingue quatre composants principaux (figure 2.1), qui utilisent le même formalisme de représentation des connaissances. Ce formalisme peut être très simple, comme par exemple des mots-clés, ou plus complexe, comme par exemple des graphes conceptuels.

Modèle de documents : correspond à la modélisation du contenu sémantique des documents, dans le formalisme de représentation de connaissances. Le choix du formalisme utilisé est crucial, mais il est toujours difficile, d'obtenir une modélisation exprimant parfaitement l'idée initiale de l'auteur.

Modèle de requête : correspond à la modélisation du besoin d'information de l'utilisateur, dans le formalisme de représentation de connaissances. Ce formalisme limite souvent la précision de définition du besoin. De plus, la "qualité" de la requête exprimée par l'utilisateur varie considérablement avec sa connaissance du domaine et avec son aptitude à définir son besoin. Souvent, il y a une importante perte d'information entre le besoin et son expression.

Fonction de correspondance : le système évalue la pertinence (la valeur de correspondance) des documents par rapport à la requête. La fonction de correspondance est un élément clé d'un SRI, car la qualité des résultats dépend de l'aptitude du système à calculer une pertinence des documents la plus proche possible du jugement de l'utilisateur.

Base de connaissances : un thésaurus, composé de concepts apparaissant dans le corpus, reliés entre eux par diverses relations (spécificité/généricité, synonymie, ...etc.). En considérant par exemple les relations de synonymie entre les concepts, il est ainsi possible de retrouver, pour une requête composée du terme "voiture", des documents traitant de voiture ou d'automobile.

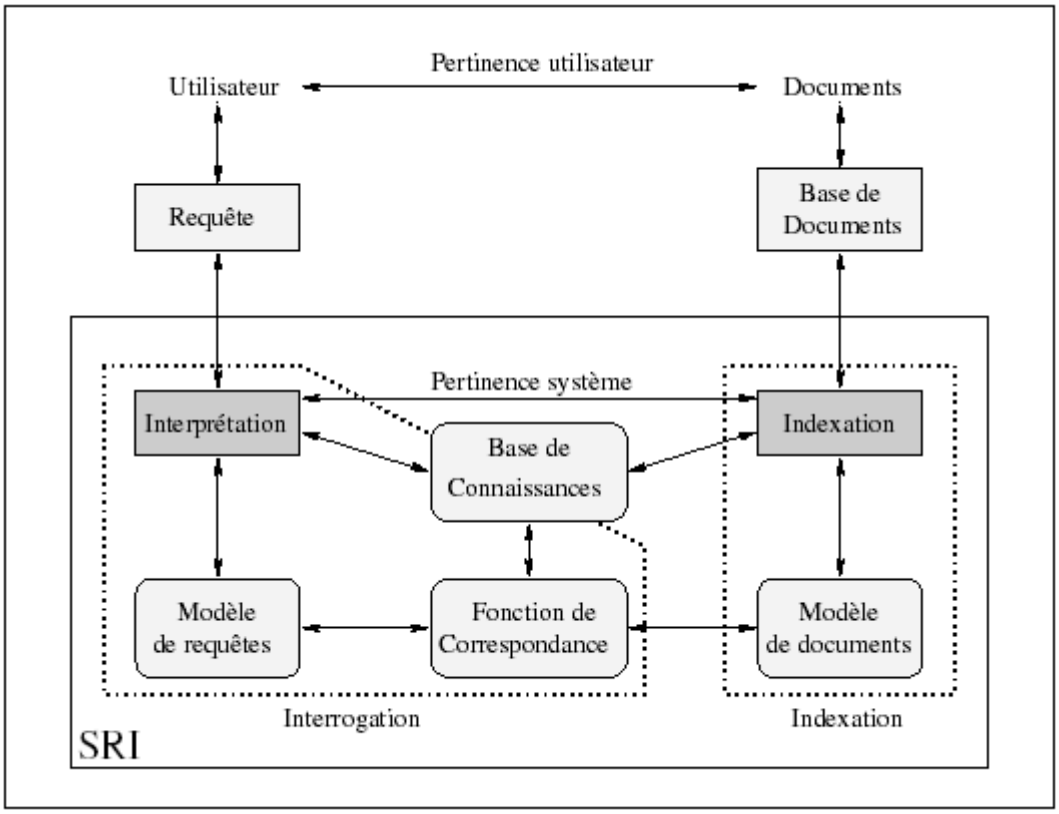

Figure 1.1 : Schéma général d'un modèle de Recherche d'Information [extrait Géry 02]

Ces quatre éléments permettent de modéliser un processus de recherche d'information : ils forment ce qu'on appelle un **modèle de RI**.

4 La recherche d'information audio

Les premières approches de la recherche d'information dans un contenu parlé ont d'abord utilisé des techniques similaires à celle développées pour les documents textuels, appliquées à la transcription automatique du flux de parole.

La recherche documentaire audio (Spoken Document Retrieval, SDR) est la première formalisation de la tâche au travers de la campagne TREC7 (Text REtrieval Conference).

En pratique, la SDR est réalisée par la combinaison des techniques de la reconaissance automatique de la parole (RAP) et de la recherche d'information (RI). Un système de reconnaissance automatique de la parole est appliqué à un ensemble de fichiers audio et génère une transcription de la parole. Cette transcription est alors indéxée et une recherche

est alors lancée par un système de recherche. Le résultat fourni pour une requête est une liste de pointeurs vers des fichiers audio[graffolo 00].

La tâche SDR de TREC [Garofolo et al. 99] propose d'indexer les transcriptions de documents contenant de la parole journalistique (500 heures d'émissions radio). Les transcriptions sont réalisées par des systèmes de reconnaissance automatique de la parole et contiennent une part d'erreurs qui fait diminuer les performances de la recherche documentaire textuelle classique. La qualité des transcriptions est mesurée par le taux d'erreurs sur les mots, Word Error Rate (WER). L'information recherchée dans les documents audio est exprimée sous la forme d'une requête textuelle.

Il faut remarquer que la plupart des systèmes de recherche documentaire fonctionnent soit sur du texte soit sur l'audio, mais ne mélangent pas les deux modalités. Dans [Sanderson et Shou 02], [Favre 03] les auteurs soulignent qu'en général ce mélange défavorise l'audio et qu'aucune technique permettant de réduire cet écart n'a été proposée à ce jour.

Les évaluations TREC montrent que le taux d'erreur des mots est linéairement corrélé aux performances en recherche documentaire et qu'un taux d'erreur inférieur à 40% permet d'obtenir des résultats acceptables par l'utilisateur [Garofolo et al. 99]. Cette bonne réussite s'explique d'abord par la longueur des requêtes TREC et la quantité d'informations qu'elles contiennent (environ 10 mots porteurs de sens, à comparer à des requêtes WEB de moins de 2 mots en moyenne). L'impact du taux d'erreur peut être limité à 10% des performances sur la transcription manuelle en utilisant des techniques d'expansion de requête et de document.

Les techniques de recherche d'information audio tentent maintenant d'aller plus loin que la parole des flux radio, en se focalisant sur la parole spontanée et sur les applications temps réel. E.W Brown s'attache à annoter des flux télévisuels et des conférences avec des informations susceptibles d'intéresser le spectateur[Brown et al. 01]. Pour ce qui est de la parole spontanée, Byrne et son équipe [Byrne et al. 04] ont annoté un corpus de 10000 heures d'interview puisées dans les enregistrements de la Shoah Visual History Foundation. Ce corpus est à ce jour le plus grand corpus de parole spontanée réunissant de nombreux locuteurs sur le même thème ; ce corpus permettra certainement de mieux tester les approches de recherche d'information et de segmentation que les corpus téléphoniques de Switchboard [Godfrey et al. 92].

Le taux d'erreur de mots n'est pas le seul problème lié à la transcription automatique du contenu parlé, les systèmes de transcription ont en effet un vocabulaire limité aux mots les plus fréquents (dans le but de minimiser le taux d'erreur de mots, tout en limitant les ressources nécessaires). Les mots les moins fréquents sont considérés comme des mots hors vocabulaire (Out of Vocabulary, OOV) et ignorés lors du décodage du signal de parole. Ils ne pourront être retrouvés et paradoxalement, ce sont justement les événements peu fréquents et inattendus qui sont le plus susceptibles de sélectionner les documents pertinents. En effet, le moteur de recherche SpeechBot [Thong et al. 00] a offert pendant plusieurs années l'accès à du contenu parlé transcrit automatiquement sur le web et il a été observé que plus de 12% des mots utilisés dans les requêtes étaient hors vocabulaire. Le problème est aussi lié aux modèles de langages nécessairement mal estimés pour les langues à ressources minoritaires comme les langues africaines [Abdillahi et al. 06].

Des techniques basées sur l'utilisation de sous-parties des mots comme les phonèmes ou les radicaux sont apparues pour essayer de remédier au problème des mots hors vocabulaire [Wechsler et al. 98]. Ces approches demandent une phonétisation de la requête, puis la comparaison de cette séquence de phonèmes avec les hypothèses de transcription phonétique du système de transcription. Une mesure de confiance basée sur l'adéquation entre la modélisation phonétique et le contenu acoustique est utilisée afin de ne rapporter que des séquences proches de la meilleure hypothèse (probabilité a posteriori du sous-graphe d'hypothèses passant par le chemin étudié). L'utilisation du treillis de phonèmes apporte un gain intéressant en rappel au détriment de la précision car de nombreux passages ont une transcription phonétique similaire à la requête sans pour autant impliquer la présence des mêmes mots.

Yu et Seide intègrent la recherche dans le treillis de phonèmes avec une recherche dans le treillis de mots afin de profiter de l'augmentation à la fois du rappel et de la précision. Face à un taux d'erreur de mots de l'ordre de 43% à 60% selon les conditions, ils observent un gain de 10% en performance sur la détection de mots (word spotting) par rapport à l'utilisation d'une des deux méthodes isolément. Les mots hors vocabulaire ont des effets de bord sur la qualité de la transcription, car ils sont remplacés par une séquence de mots acoustiquement proches, mais qui diverge du contenu réel et provoque des erreurs autour du mot inconnu [Yu et Seide 04].

5 Conclusion

L'objectif de la recherche d'information audio est de retrouver les documents parlés satisfaisant un utilisateur. L'information parlée introduit de nombreux challenges comparée à la recherche d'information textuelle. La variabilité de la parole est le premier obstacle à l'extraction des descripteurs sémantiques d'un document. En outre, les nombreuses techniques d'analyse des phénomènes linguistiques sur le texte peuvent être appliquées à la transcription automatique du message parlé, au détriment d'erreurs proportionnelles au taux d'erreur de mots. L'impact de ces erreurs est lourd lorsque des structures plus petites que le document, comme la phrase, doivent être employées. Intégrer les spécificités de la parole à la recherche d'information parlée représente un verrou à lever.

Comme nous venons de le constater la totalité des systèmes de recherche d'information audio opèrent sur des documents textuels obtenus par transcription des documents audio originels, ceci suppose l'existence de systèmes de reconnaissance de la parole robuste ; nous présentons dans le chapitre suivant une introduction à ces systèmes et aux outils utilisés à cette fin.

Chapitre2 : Reconnaissance Automatique de la Parole

1 Introduction

De nos jours, les ordinateurs sont largement utilisés pour communiquer par l'intermédiaire de la parole. La manipulation automatique des documents audio repose sur l'utilisation d'un Système de Reconnaissance Automatique de la Parole (SRAP) permettant le décodage automatique du signal de parole.

Le signal de parole est caractérisé par de nombreux paramètres. Il possède une grande variabilité. Il est différent d'un locuteur à un autre et même un locuteur ne prononce jamais un mot deux fois de la même façon. Les différences d'âge, de sexe, d'accent, d'émotivité entre locuteurs rendent complexes l'extraction d'informations pertinentes concernant le signal, cette extraction se voulant être indépendante du locuteur.

L'acoustique du milieu ambiant lors de la prise de son (bruits extérieurs, bruits de bouche, respirations, éternuements...) ainsi que la qualité de l'enregistrement génèrent encore des difficultés que le SRAP doit surmonter. La segmentation du signal en mots s'avère également un processus complexe à réaliser pour un système de RAP. En effet, pour un SRAP, le signal de parole est un flux continu et il n'a pas la capacité d'interpréter ce signal comme étant une suite de mots.

Nous présentons dans la première partie de ce chapitre, le signal de la parole et les caractéristiques d'un SRAP. Ensuite nous exposons les différentes étapes de la reconnaissance automatique de la parole.

Dans la deuxième partie du chapitre, un outil très utilisé dans la RAP est présenté ; il s'agit des modèles de markov cachés (HMM). Nous définissons un HMM, ensuite nous décrivons les différentes étapes de la mise en œuvre d'un système RAP avec les HMM.

2 Reconnaissance automatique de la parole

2.1 Le signal de la parole

Le signal de parole est une onde acoustique modulée par l'appareil phonatoire en fréquence et en amplitude. Cette onde est généralement présentée sous la forme d'une courbe (**Figure2.1**) représentant les variations d'amplitude du signal au cours du temps.

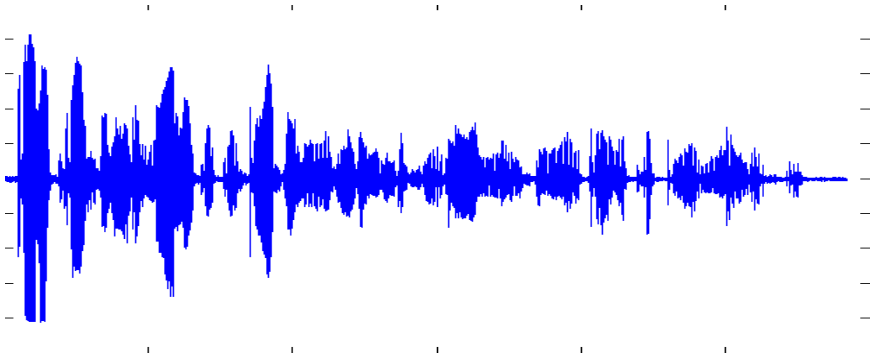

Figure 2.1 : Représentation du signal de parole correspondant à la phrase : « kanat riHu ecchamali tatadzadalou u:a chamsa fi ai:in minhouma kanat aku:a mina el oxra »

Le signal de parole est une concaténation de réalisations acoustiques élémentaires. Ces réalisations sont plus connues sous le nom de *phonèmes.* Un phonème est une entité abstraite définie 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 construits. 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.

2.2 Caractéristiques d'un système de RAP

Un système de RAP est caractérisé par son mode de fonctionnement, le mode d'élocution, le type de vocabulaire et la syntaxe du langage.

Système dépendant ou indépendant du locuteur : les systèmes de RAP peuvent être regroupés en trois classes suivant le nombre de locuteurs qui utilisent ces systèmes :

- monolocuteur : le système de RAP est adapté à la voix d'un seul locuteur.
- multilocuteur : le système est utilisé par un groupe de personnes qui sont connus par le système dès l'apprentissage.
- indépendant du locuteur : tout locuteur peut utiliser le système.

Mode d'élocution : les systèmes de reconnaissance diffèrent suivant le mode d'élocution utilisé, en général on trouve trois types de systèmes de reconnaissance de mots :

- système de reconnaissance de mots isolés : chaque mot est prononcé isolement en marquant des pauses entre les mots (élimination de certains problèmes de coarticulation).
- système de reconnaissance de mots connectés : le système peut reconnaître une suite de mots sans marquer des pauses entre les mots (chiffres connectés).
- système de reconnaissance de la parole continue : la parole continue est le discours usuel, dans ce cas un modèle de langage est introduit.

Vocabulaire : c'est l'ensemble de mots que le système est capable de reconnaître, il est caractérisé par :

- sa taille qui peut varier de quelques mots à plusieurs dizaines de milliers de mots.
- sa nature : par exemple si on prend un vocabulaire constitué de mots phonétiquement proches, alors il sera très difficile de les distinguer.

Syntaxe du langage : la syntaxe spécifie les contraintes imposées sur la suite de mots prononcés. Le but de la syntaxe est de faciliter la tâche du système pendant la reconnaissance, en limitant le nombre de mots candidats, par exemple après la phrase "trente et" on est obligé de mettre le mot "un".

2.3 Différentes étapes de la RAP

Le but de la RAP est d'identifier à partir du signal vocal le message linguistique le plus vraisemblable, cette identification se fait suivant deux grandes phases :

- une phase d'analyse acoustique ou paramétrisation, son rôle est d'extraire du signal les informations pertinentes et d'éliminer la redondance.
- une phase de comparaison ou de décodage, son rôle est la comparaison des informations issues de la première phase, aux données de références.

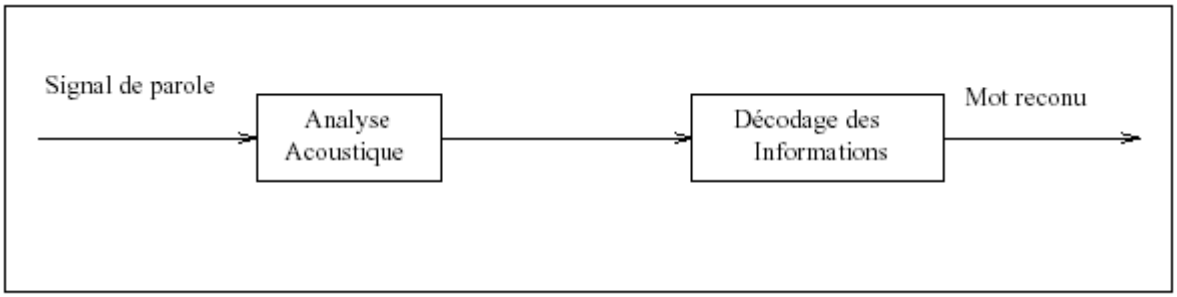

Figure 2.2 : Les principales tâches d'un système RAP

3 Analyse acoustique

En acoustique, un son se définit classiquement au moyen de son amplitude, de sa durée et de son timbre [Calliope 89]. 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 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.

3.1 La mise en forme du signal de parole

Avant de commencer les calculs des coefficients, il est nécessaire de faire le calcul préalable suivant :

• Filtrage analogique en sortie du microphone :

Les informations acoustiques du signal de parole se situent dans la bande fréquentielle [50Hz,8khz], le rôle principal de ce filtrage est d'éliminer toute information hors de cette bande.

• Conversion analogique/numérique (échantillonnage) :

Afin d'utiliser ou de traiter les signaux continus, sortant d'un microphone ou d'un appareil électronique, par des calculateurs, il est nécessaire de numériser ou de discrétiser ce signal.

Cette opération de discrétisation s'appelle l'échantillonnage du signal et l'opération inverse s'appelle l'interpolation (**Fig 1.3**). Si on note par *x(t)* un signal continu, l'échantillonnage de *x(t)* est l'application qui fait correspondre au signal *x(t)* un signal discret $(x_1,x_2,..,x_n,..)$ avec :

$$x_n = x(t_n) \qquad (2.1)$$

Lorsque $t_n - t_{n-1} = T$ est constante pour tout n on note par :

$$fe = \frac{1}{T} \qquad (2.2)$$

T est le pas d'échantillonnage du signal $x(t)$.

f_e est appelée fréquence d'échantillonnage.

Pour avoir une perte d'information presque nulle entre le signal continu et le signal échantillonné suivant une fréquence d'échantillonnage f_e , il faut et il suffit que f_e soit au moins supérieure au double de la fréquence la plus élevée f_m de ce signal (théorème de shannon), c-à-d :

$$f_m \leq \frac{f_e}{2} \qquad (2.3)$$

A titre d'exemple, Si les enregistrements sont effectués à travers les lignes téléphoniques on a f_m = 3.3*khz* ce qui implique que :

$$f_e > 6.6khz$$

Si les enregistrements sont effectués dans le laboratoire dans ce cas $f_m = 8khz$ ce qui donne

$$f_e > 16khz$$

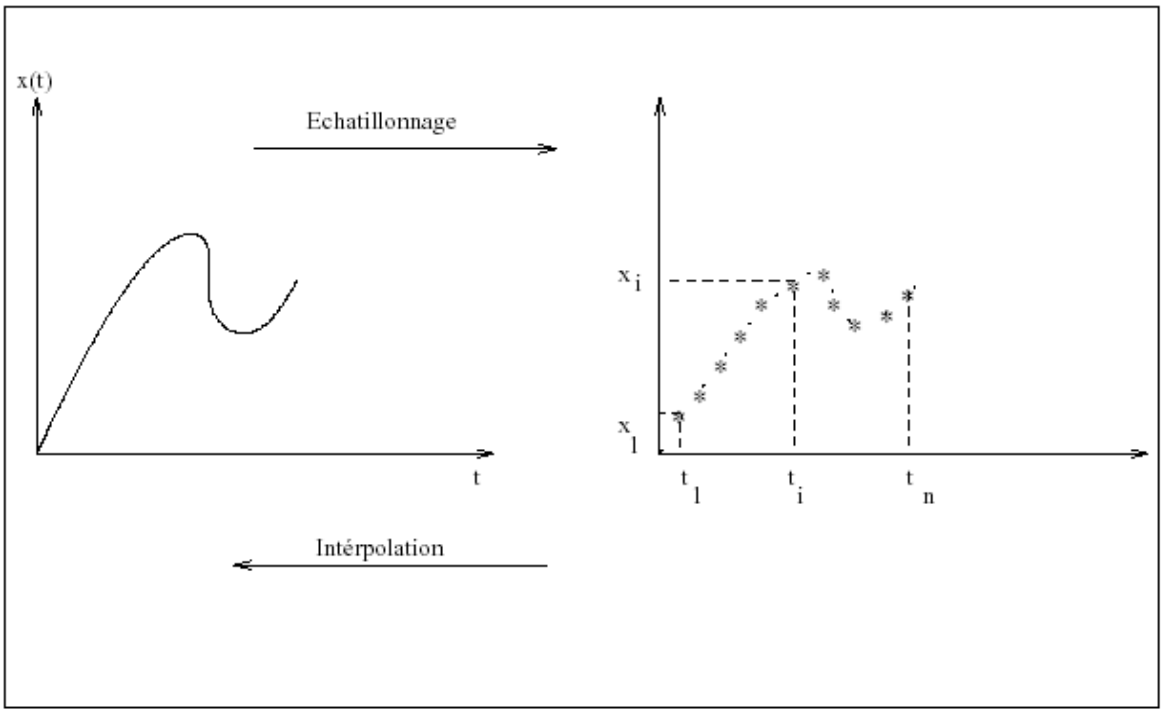

Figure 2.3 : L'échantillonnage et l'interpolation d'un signal

• Pré-accentuation du signal :

Cette opération est effectuée afin de relever les hautes fréquences qui sont moins énergétiques que les basses fréquences. Si on note par x_n le signal échantillonné , alors la pré-accentuation de x_n est :

$$x_n = x_n - a\, x_{n-1} \qquad 0.9 < a < 1 \qquad (2.4)$$

• La segmentation du signal :

La plupart des méthodes d'analyse acoustique utilisent l'hypothèse de la stationnarité du signal, ce qui n'est pas vrai pour le cas de la parole, une analyse sur des segments à court terme sur lesquels le signal est supposé quasi-stationnaire est nécessaire.

Deux types de segmentation sont utilisés, soit une segmentation de signal en trames de longueur variable et qui s'appuie sur un algorithme de segmentation automatique, qui isole les zones homogènes du signal [Obrecht 88], soit une segmentation du signal en trames de longueur fixe qui se recouvrent entre eux. La longueur de la trame varie entre 20ms et 40ms. Si cette longueur est égale à 32ms et $f_e = 16kHZ$, le nombre d'échantillons par trame est de 512 échantillons.

• Fenêtrage du signal :

Pour appliquer la segmentation du signal, ce dernier est multiplié par une fenêtre rectangulaire, l'application de ce type de fenêtrage crée des oscillations importantes dans le domaine (fréquence,spectre), de plus il produit une discontinuité aux frontières des trames. Pour réduire ces effets le signal est multiplié par une fenêtre $\omega(n)$ dont la transformée de Fourrier s'approche d'une impulsion de Dirac. Le nouveau signal devient :

$$x_n = x_n.w(n) \qquad (2.5)$$

II existe plusieurs types de fenêtrage :

- Fenêtrage de Hamming, son équation est :

$$\omega(n) = 0.54 - 0.46.\cos(\tfrac{2.\pi.n}{N})$$

- Fenêtrage de Hanning défini par :

$$\omega(n) = 0.5(1 - \cos(\tfrac{2.\pi.n}{N}))$$

- Fenêtrage de Blackman son équation est :

$$\omega(n) = 0.42 - 0.5.\cos(\tfrac{2.\pi.n}{N}) + 0.08.\cos(\tfrac{4.\pi.n}{N})$$

- Fenêtrage de Kaiser [Guy 1993].

$$\omega(n) = \begin{cases} \dfrac{I_0(2\alpha\sqrt{\frac{n}{L-1} - (\frac{n}{L-1})^2})}{I_0(\alpha)} & si\ 0 \le n \le L-1 \\ 0\ ailleurs \end{cases}$$

Où $I_0(...)$ est la fonction de Bessel modifiée d'ordre 0 de première espèce.

$$I_0(x) = 1 + \frac{(x/2)^2}{(1!)^2} + \frac{(x/2)^4}{(2!)^2} + \cdots$$

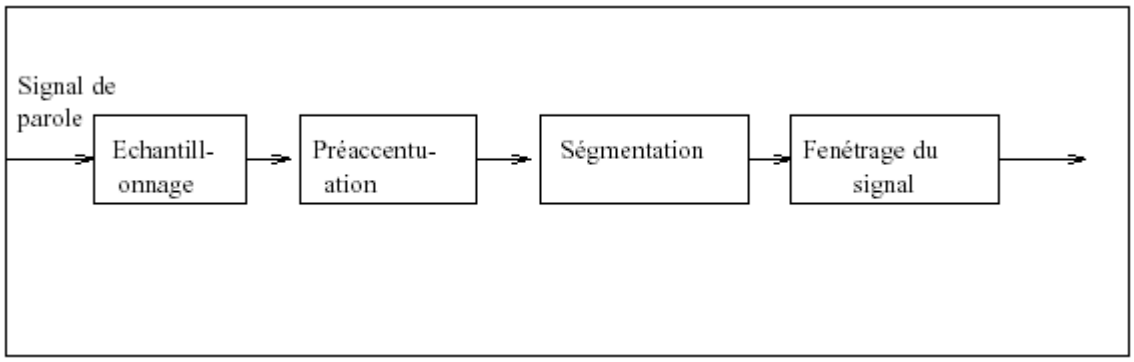

Figure 2.4 : La mise en forme d'un signal de parole

3.2 Calcul des coefficients

Une fois que le signal a subi ces transformations, les méthodes de calcul des coefficients le traite par bloc. Parmi les méthodes les plus utilisées en analyse acoustique nous citons :

- la méthode d'analyse spectrale qui donne les coefficients MFCC (Mel Frequency Cepstral Coefficients).
- la méthode de prédiction linéaire qui donne les coefficients LPCC (Linear Prédiction Cepstral Coefficient).

3.2.l Analyse spectrale (Coefficients MFCC)

Les coefficients MFCC sont les plus utilisés en RAP, ils sont introduits pour la première fois en 1980 par Davis et Mermelstein [Davis 80]. Pour calculer ces coefficients, nous appliquons sur le signal les transformations suivantes :

- **Application de la transformée de Fourrier :**

L'analyse cepstrale, présente l'intérêt de séparer la contribution de la source de celle du conduit vocal. Cette séparation est réalisée par un homomorphisme qui transforme le produit de

convolution entre la source glottique g_n et la réponse impulsionnelle du conduit vocal h_n en une addition dans le domaine cepstral. Si x_n est le signal observé alors :

$$x_n = g_n * h_n \qquad (2.6)$$

La transformée de Fourrier sur le produit de convolution :

$$X_w = TF(x_n) = TF(g_n).TF(h_n) \qquad (2.7)$$

Ensuite

$$Log|X_\omega| = Log|TF(g_n)| + Log|TF(h_n)| \qquad (2.8)$$

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

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

Pour réduire le taux de calcul il est préférable d'utiliser l'algorithme FFT (Fast Fourrier Transform (Transformée de Fourrier Rapide)).

La FFT est appliquée sur le bloc $\{x_{i1},\ldots x_{iN}\}$ pour obtenir les spectres $\{|x_{i1}|,\ldots|x_{iN}|\}$

- **Application du filtrage triangulaire :**

La période fondamentale des sons voisés produit de nombreuses harmoniques sur le spectre obtenu par la FFT, pour diminuer ces phénomènes nous effectuons des lissages sur ces spectres en appliquant une suite de filtres triangulaires, réparties sur la bande passante [100Hz,7.5khz] suivant une échelle de Bark ou de Mel et cela pour se rapprocher de l'oreille humaine (**Fig 2.5**).

L'équation de l'échelle de Bark est :

$$B = 6\text{Arcsinh}(F/600) \qquad (2.9)$$

L'équation de l'échelle de Mel :

$$M = \frac{1000}{Log\,2}.Log(1+\frac{F}{1000}) \qquad (2.10)$$

Ensuite, les bornes de ces filtres sont exprimées en échelle Mel.

On note par F_n et F_{n+1} les bornes inférieures des deux filtres triangulaires numéro n et n+1.

L'équation du n^{ieme} filtre est donnée par :

$$I_n = \begin{cases} \dfrac{f - F_n}{F_{n+1} - F_n} si F_n \le f \le F_{n+1} \\ 1 - \dfrac{f - F_n}{F_{n+1} - F_n} si F_{n+1} \le f \le F_{n+2} \end{cases} \qquad (2.11)$$

Ce filtrage est appliqué sur le bloc $\{|x_{i1}|,\ldots.|x_{iN}|\}$ obtenu par la FFT et qui correspond à la suite des fréquences $\{f_{i1},\ldots,f_{iN}\}$. Ensuite l'énergie sortant du n^{eme} filtre est donnée par :

$$e_n = \sum_{j=1}^{N} \left| X_{ij} \right|^2 I_n \qquad (2.12)$$

A la fin le bloc de signal $\{|x_{i1}|,\ldots.|x_{iN}|\}$ va être représenté seulement par les F énergies $\{e_i, ..., e_F\}$. F est le nombre de filtres triangulaires.

- **Application de la transformée de Fourrier inverse :**

Ensuite la transformée de Fourrier inverse est appliquée sur le bloc $\{e_i, ..., e_F\}$ et seulement la partie réelle de cette transformation est prise. La formule de calcul de cette transformée donne les coefficients cepstraux Ck et qui sont notés MFCC.

$$C_k = \sum_{i=1}^{F} Log(e_i).\cos(\frac{\pi k(i-0.5)}{F}) \qquad k=1,\ldots,d \qquad (2.13)$$

d est le nombre de coefficients cepstraux.

Une dizaine de ces coefficients est généralement jugée suffisante pour présenter une trame du signal.

Chaque trame du signal est représentée par un vecteur de coefficients :

$$X=(C_1,..,C_d) \qquad (2.14)$$

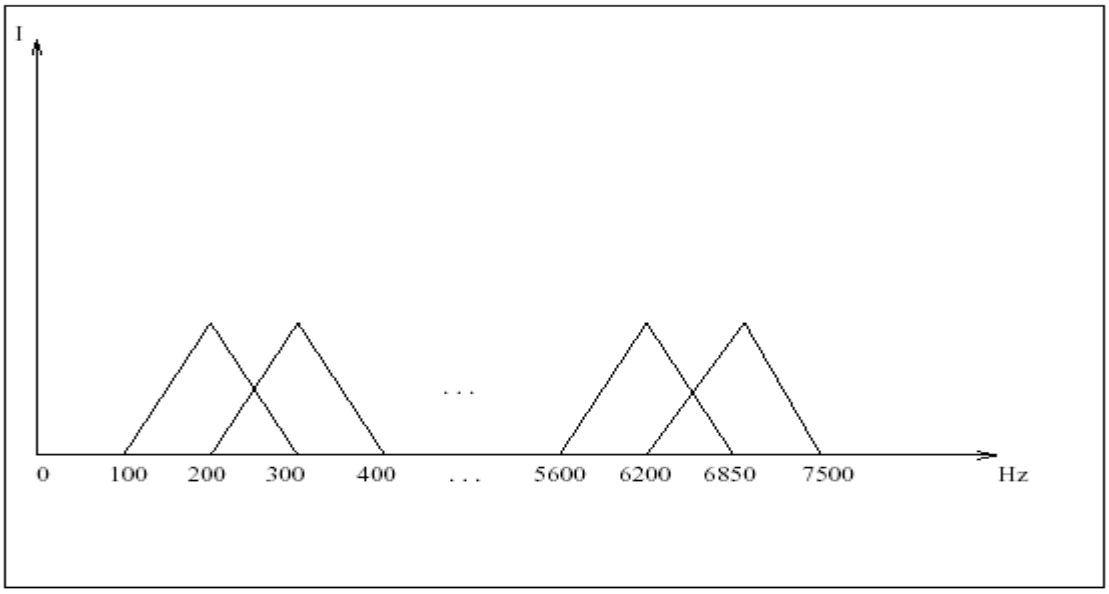

Figure 2.5 : Répartition fréquentielle des filtres triangulaires suivant l'échelle de MEL

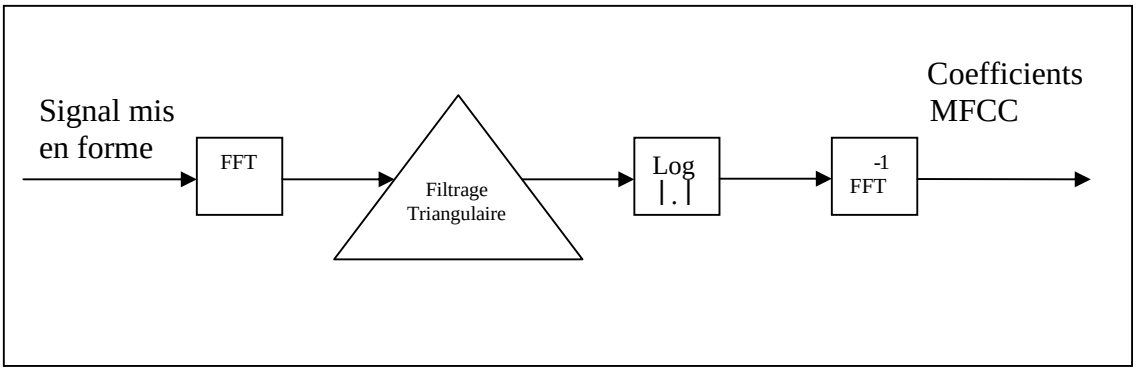

Figure 2.6 : Calcul des coefficients MFCC

3.2.2 Analyse par prédiction linéaire (coefficients LPCC)

Après l'étape de la mise en forme du signal, on calcule les coefficients cepstraux du signal de parole. Cette analyse suppose que le signal vocal peut être considéré comme un signal auto-régressif décrit par le modèle :

$$x_n - \sum_{i=1}^{p} a_i x_{n-i} = \varepsilon_n \qquad \text{n=1,….,T} \qquad (2.15)$$

- ε_n est un bruit blanc gaussien de variance σ^2.

Pour calculer les a_i *et* σ^2 on est amené à résoudre le problème de minimisation donné par la méthode des moindres carrées :

$$\min_{a_1,\ldots,a_p} \sum_{n=1}^{T} (x_n - \sum_{i=1}^{p} a_i x_{n-i})^2 \qquad (2.16)$$

Après les calculs on trouve :

$$\begin{cases} \sum_{i=0}^{p} a_i . R_{ij} = 0 \quad j = 1,\ldots, p \\ \sum_{i=0}^{p} - a_i . R_{i0} = \sigma^2 \end{cases} \qquad (2.17)$$

avec :

$a_0 = -1 \, et \, R_{ij} = \sum_{n=1}^{T} (x_{n-i} . x_{n-j})$

Après le calcul des coefficients a_i on calcule les coefficients cepstraux C_k, qui sont nommés coefficients LPCC (Linear Predection Cepstral Coefficient). Pour k=l,..,d on a :

$$C_k = a_k - \sum_{i=1}^{k-1} \frac{i}{k} . a_{k-1} . C_i \qquad (2.18)$$

4 Décodage des informations acoustiques

Pour décoder les informations issues de l'analyse acoustique, il existe trois approches : l'approche analytique, l'approche globale, et l'approche statistique .

4.1 Approche analytique

Cette approche est réalisée suivant les étapes suivantes :

- Segmentation du signal obtenu par l'analyse acoustique, en unité de taille phonétique (phonème, syllabe, ...). Cette segmentation est faite à l'aide de différents critères (énergie, stabilité,...).
- Identification phonétique des segments par comparaison des données acoustiques avec les formes de références.
- Exploitation de la suite phonétique identifiée par certains analyseurs (lexical, syntaxique) pour déterminer au juste le mot ou la phrase prononcée [Haton et Al 91].

4.2 Approche globale

Cette approche consiste à identifier un mot ou une phrase comme des entités élémentaires en effectuant des comparaisons avec des références enregistrées. Cette comparaison est réalisée en utilisant le principe de l'alignement temporel dynamique (Dynamic Time Warping DTW), cet alignement est réalisé à l'aide du principe d'optimalité de Bellman [Bellman 57].

Si on note par $V = \{\omega, \ldots \omega_L\}$ le vocabulaire du système, où chacun des mots ω_i est représenté par une ou plusieurs formes acoustiques de références notée $R\omega_i$ (par exemple les paramètres spectraux issues de l'analyse acoustique et qui sont calculés de manière périodique). Notons par O_ω la suite de formes associée à un mot inconnu ω, l'identification du mot ω est réalisée suivant le critère :

$$\omega^* = \arg(\min_{\omega_i \in V} D(O_\omega, R_{w_i}) \qquad (2.19)$$

Le problème de calcul de cette distance est que la durée du mot ω_i est différente de celle de ω. La solution, est de calculer cette distance par un alignement temporel qui rapproche le mieux les deux formes $R\omega_i$ et O_w. La construction de cet alignement est réalisée récursivement sur l'indice temporel en exploitant le fait que le chemin optimal est l'extension d'un chemin partiel optimal. Les premières applications de cet algorithme en RAP sont développées en URSS en 1968 [Vintsujk 68], [Velichko 70], puis au Japon à partir de 1970 [Sakoe et Chiba 71]. Cette méthode est efficace, elle a donné des résultats meilleurs pour les systèmes mono-locuteur à petit vocabulaire et en mots isolés.

Des extensions de DTW ont été proposées par :

- [Rabiner et al 79] pour les systèmes indépendants du locuteur.
- [Sakoe 79],[Myers et Rabiner 81],[Bridle et al 82] pour les systèmes de mots connectés.

4.3 Approche statistique

F.Jelinek a proposé une formalisation statistique simple issue de la théorie de l'information et qui consiste à décomposer le problème de la RAP [Jelinek 76]. Etant donnée une suite d'observations $Y_1,\ldots,Y_T$ associée à une suite de mots prononcés ω, l'approche statistique consiste à trouver la suite de mots ω^* la plus probable connaissant la suite d'observations $Y_1,\ldots,Y_T$.

$$\omega^* = \arg\max_{\omega \in V} \Pr(\omega | Y_1,...,Y_T) \qquad (2.20)$$

la règle de Bayes nous donne :

$$\Pr(\omega | Y_1,...,Y_2) = \frac{\Pr(Y_1,...,Y_T | \omega).\Pr(\omega)}{\Pr(Y_1,...,Y_T)} \qquad (2.21)$$

• $\Pr(Y_1,\ldots,Y_T \mid \omega)$ représente la probabilité d'observer la suite $Y_1,\ldots,Y_T$ sachant la suite de mots prononcés *w,* cette probabilité est estimée par une modélisation acoustique.

• $\Pr(\omega)$ la probabilité a priori que la suite de mots ω soit prononcée. Elle est estimée par un modèle de langage. Dans le cas de la reconnaissance des mots isolés on suppose que tous les mots ont la même probabilité d'être prononcés ($\Pr(\omega)=1$).

Puisque $\Pr(Y_1,\ldots,Y_T)$ ne dépend pas de ω, l'équation précédente devient :

$$\omega^* = \arg\max_{\omega \in V} \Pr(Y_1,...,Y_T | \omega) \qquad (2.22)$$

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).

5 Les modèles de Markov cachés

Les modèles de Markov cachés (HMM : Hidden Markov Model) ont été proposés pour la première fois dans le cadre de la reconnaissance automatique de la parole en 1975[Baker 75a, Baker 75b] et se sont imposés depuis comme modèles de référence dans ce domaine.

5.1 Définition d'un HMM

Un HMM est un cas particulier des modèles stochastiques graphiques, et peut être vu comme un automate probabiliste. Il est généralement caractérisé par un quadruplet (S, Π, A, B) :

- $S = \{S_0, \ldots, S_i, \ldots, S_k\}$ est l'ensemble des états de l'automate.
- $\Pi = \{\Pi_0, \ldots, \Pi_I, \ldots, \Pi_K\}$, avec Π_I étant la probabilité que *Si* soit l'état initial.
- A est l'ensemble des probabilités de transition d'un état vers un autre. A est caractérisé par une matrice $k * k$ d'éléments *aij* avec i *et* $j \in [0, k]$ et k le nombre d'états. Tout élément *aij* de cette matrice est la probabilité d'atteindre l'état S_j au temps t sachant que nous étions dans l'état S_i au temps t — 1.
- B est un ensemble de lois de probabilité *bi(o)* donnant la probabilité *P(o|Si)* que l'état S_i ait généré l'observation *o*. Cette probabilité est la vraisemblance de l'observation au regard de S_i.

Un HMM étant un automate (probabiliste), les contraintes suivantes doivent être respectées :

1. La somme des probabilités des états initiaux doit être égale à 1 :
$$\sum_i \pi_i = 1$$
2. La somme des probabilités des transitions sortant d'un état doit être égale à 1 :
$$\forall i \ \sum_j a_{ij} = 1$$
3. La somme des probabilités des émissions d'un état doit être égale à 1 :
$$\forall i \quad \sum_o b_i(o) = 1 \qquad \text{dans le cas d'observations discrètes.}$$
$$\forall i \quad \int_o b_i(o)do = 1 \qquad \text{dans le cas d'observations continues.}$$

Un HMM représente un objet par deux suites de variables aléatoires : l'une dite *cachée* et l'autre *observable*. La suite observable correspond à la suite d'observations $o_1, o_2, \ldots, o_T$ où les o_i sont des vecteurs d'observations du signal à reconnaître.

La suite cachée correspond à une suite d'états $q_1, q_2, \ldots, q_T$ où les q_i prennent leurs valeurs dans l'ensemble des N états du modèle $\{S_1, S_2, \ldots, S_N\}$.

La suite observable est définie comme une réalisation particulière de la suite cachée. L'objectif est de déterminer la meilleure séquence d'états $Q^* = q_1^*, q_2^*, \ldots, q_T^*)$ à partir de la séquence d'observations $O = (o_1, o_2, \ldots, o_T)$. Le meilleur chemin Q^* est celui qui maximise la probabilité a *posteriori P(Q|O)*.

$$Q^* = \arg\max_Q P(O|Q)\ P(Q) \qquad (2.23)$$

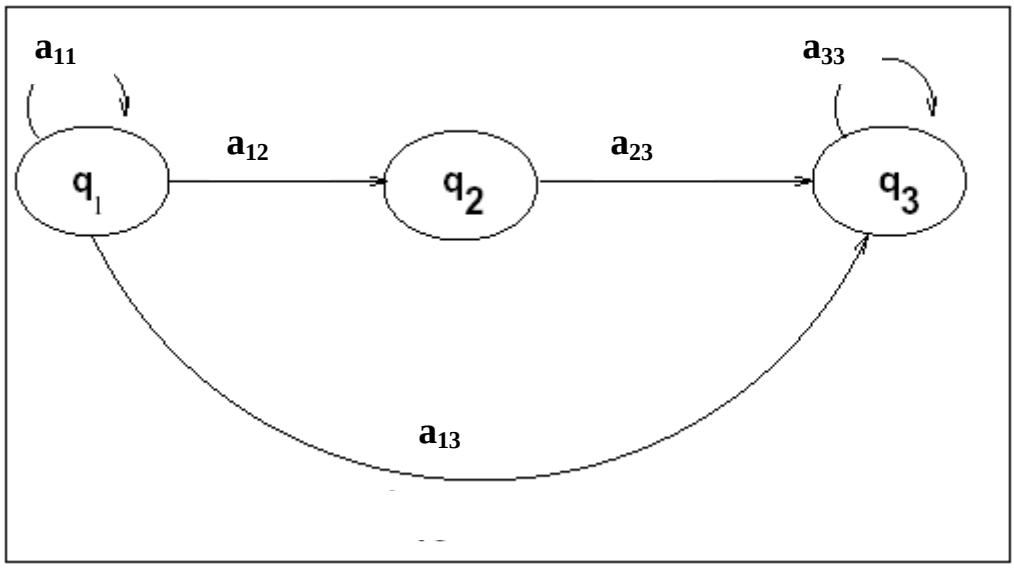

Figure 2.7 : Exemple d'un modèle de markov caché à trois états

Un HMM présente plusieurs avantages : il s'inscrit dans un formalisme mathématique bien établi, il bénéficie de méthodes d'apprentissage automatique pour la détermination de ses paramètres et il est particulièrement bien adapté à la modélisation de processus à évolution temporelle.

5.2 Mise en œuvre

La mise en œuvre d'un système de reconnaissance de la parole à partir de HMM nécessite de formuler quelques hypothèses simplificatrices dans le but d'adapter le cadre théorique des HMM à la RAP mais aussi d'en simplifier le formalisme mathématique et ainsi proposer des algorithmes d'apprentissage et de classification optimaux sous ces hypothèses.

5.2.1 Hypothèses simplificatrices

Soit $O = O = (o_1, o_2, \ldots, o_T)$ une suite de T observations. Soit $Q = (q_1, q_2, \ldots, q_T)$ une séquence d'états alignée avec la suite d'observations ; au temps t le HMM est dans l'état q_t engendrant l'observation o_t.

Hypothèse n° 1

La probabilité qu'une observation o_t soit émise au temps t ne dépend pas des observations antérieures.

$$P(o_t \mid q_t, q_{t-1}, \ldots, q_1, o_{t-1}, o_{t-2}, \ldots, o_1) = p(o_t \mid q_t, q_{t-1}, \ldots, q_1) \qquad (2.24)$$

Hypothèse n° 2

La probabilité qu'une observation soit émise au temps *t* ne dépend pas des états précédemment visités, mais seulement de l'état courant.

$$P(o_t | q_t, q_{t-1}, ..., q_1) = P(o_t | q_t) \qquad (2.25)$$

Hypothèse n° 3

La probabilité que le HMM soit dans l'état q_t à l'instant *t* ne dépend que de l'état dans lequel il se trouvait à l'instant *t*-1.

$$P(q_t | q_{t-1}, q_{t-2}, ..., q_1) = P(q_t | q_{t-1}) \qquad (2.26)$$

Un modèle respectant cette dernière hypothèse est appelé modèle de Markov du premier ordre.

5.2.2 Topologie du modèle

Le nombre d'états d'un HMM dépend de l'entité acoustique qu'il modélise. L'entité la plus répandue est le phonème, mais il est possible de considérer des entités plus grandes (supra-phonétique), comme la syllabe ou le mot. Cependant construire un système possédant un modèle pour chaque mot d'une langue n'est pas envisageable pour des raisons de temps et d'espace de calcul mais aussi pour des raisons de taille de la base d'apprentissage devant contenir suffisamment d'exemples de chaque mot pour obtenir des modèles fiables. Une telle modélisation est alors inconcevable pour des systèmes grand vocabulaire permettant de reconnaître plusieurs dizaines de milliers de mots différents. Néanmoins sous certaines contraintes comme l'utilisation d'un vocabulaire restreint cette modélisation peut s'avérer avantageuse notamment pour la modélisation des phénomènes de coarticulation.

Un phonème est généralement décomposé en 3 parties : un début, une partie stable et une fin. Une topologie à 3 états est par conséquent utilisée. Le second état correspondant à la partie stable est l'état caractérisant le mieux le phonème alors que le premier et dernier état modélisent les effets de la co-articulation, c'est à dire les transitions entre phonèmes. Ceux-ci correspondent donc aux parties instables du phonème car elles sont influencées par le contexte gauche et droit. Dans le but de restituer l'évolution temporelle du signal de la parole une topologie gauche-droite est adoptée dans la grande majorité des cas. Ceci veut dire qu'aucun retour en arrière n'est possible.

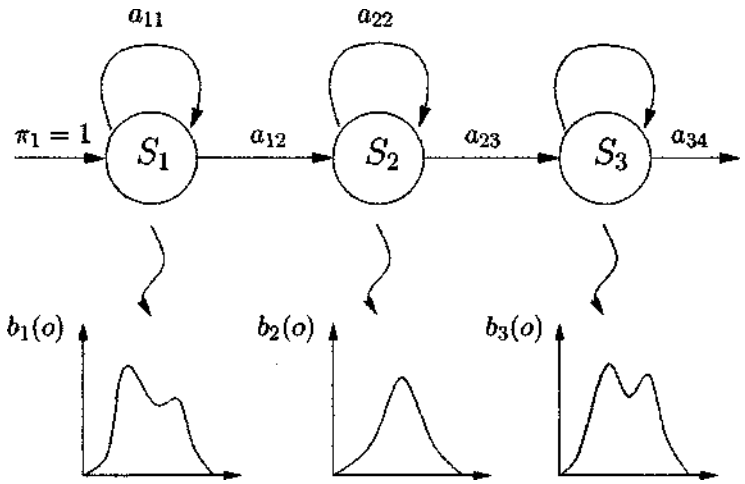

FIGURE 2.8 : *HMM gauche-droite à 3 états usuellement utilisé pour la modélisation de phonèmes. Les lois de probabilité bi(o) fournissant les probabiltés qu'une observation o ait été générée par un état Si sont modélisées par des modèles à mélange de*

Chaque état S_i d'un HMM renvoie pour une observation o la probabilité que o ait été générée par S_i. Le calcul de cette probabilité appelée également vraisemblance de l'observation s'appuie sur une fonction de densité de probabilités $b_i(0)$. Cette fonction $b_i(0)$ est un modèle paramétrique de l'ensemble des observations pouvant être générées par l'état S_i. La plupart des systèmes s'appuient sur des densités de probabilités continues modélisées par un mélange de lois normales (distribution gaussienne des observations). La vraisemblance d'une observation o est donc donnée par :

$$b_i(o) = \sum_{j=1}^{N_\lambda} \lambda_j N(o, \mu_j, \Sigma_j) \qquad (2.27)$$

Avec

$$N(o, \mu_j, \Sigma_j) = \frac{1}{\sqrt{(2\pi)^M |\Sigma_j|}} \exp(-\frac{1}{2}(o-\mu_j)' \Sigma_j^{-1} (o-\mu_j)$$

N_λ est le nombre de gaussiennes, λ_j est le poids le la j^{ieme} gaussienne, μ_i et Σ_j sont respectivement le vecteur moyen et la matrice de covariance de la j^{teme} gaussienne et M la dimension du vecteur d'observations. La figure 1.9 présente un HMM gauche-droite à 3 états utilisé pour la modélisation de phonèmes.

5.2.3 Apprentissage

Considérons un ensemble de HMM M_j et un ensemble de T observations O_j. Apprendre les paramètres des HMM revient à chercher le meilleur ensemble de paramètres $A_j^* = (\mu_i^*, \Sigma_j^*)$ tel que la probabilité que O_j ait été générée par M_j soit maximale (critère du maximum de vraisemblance) .

$$A_j^* = \arg\max_{A_j} \prod_{t=1}^{T} P(O_j(t) \mid M_j, A_j) \qquad (2.28)$$

Idéalement, c'est $P(M_j|O_j,A_j)$ qui devrait être maximisée. L'apprentissage serait alors plus discriminant : lorsque la vraisemblance du modèle *j* augmente pour les exemples correspondant au modèle *j*, les vraisemblances des autres modèles devraient diminuer pour ces mêmes exemples. Les HMM devraient donc être entraînés, non seulement pour maximiser la probabilité de générer les exemples de sa propre classe, mais aussi pour les discriminer par rapport aux autres classes (critère du maximum a *posteriori).* Parce qu'il n'existe pas de méthode permettant de maximiser directement $P(O_j \backslash M_j, A_j)$, les paramètres des modèles sont obtenus en maximisant l'équation par la méthode itérative de Baum et Welch [Baum 72], qui est un cas particulier de l'algorithme EM (Expectation Maximisation) [Dempster 77].

5.2.4 Décodage

Le décodage de la parole par des modèles HMM revient à déterminer la meilleure séquence d'états $Q^* = (q_1^*, q_2^*, \ldots, q_T^*)$ à partir de la séquence d'observations $O = (o_1, o_2, \ldots, o_T)$:

$$\begin{aligned} Q^* &= \arg\max_{Q} P(O \mid Q) \\ &= \arg\max_{Q} \pi_0 \prod_{t=1}^{T} a_{q_{t-1}q_t} . b_{q_t}(o_t) \end{aligned} \qquad (2.29)$$

Une solution évidente est de calculer la probabilité $P(O|Q)$ de toutes les séquences d'états *Q* possibles et de ne retenir que la meilleure. Ceci peut se faire en construisant un arbre. A chaque temps *t* une couche de nœuds internes est ajoutée à l'arbre. Chaque nœud interne représente un état particulier des modèles et contient la probabilité de se trouver dans cet état à l'instant *t.* Les probabilités des différentes hypothèses de reconnaissance sont contenues dans les feuilles de cet arbre. Cependant une telle solution est en pratique inapplicable car le nombre d'hypothèses est très grand.

L'algorithme de Viterbi, variante stochastique de la programmation dynamique, propose de simplifier l'arbre au fur et à mesure de sa construction. En effet, lors de son déroulement on se trouve rapidement avec des branches proposant les mêmes substitutions, mais avec des

probabilités différentes. Plusieurs hypothèses peuvent se retrouver dans le même état au même instant. L'algorithme de Viterbi stipule qu'il n'est pas nécessaire de dérouler les hypothèses de plus faible probabilité car elles ne peuvent plus être candidates pour décrire le message le plus probable.

La mise en œuvre de cet algorithme consiste à construire de façon itérative la meilleure séquence d'états à partir d'un tableau T * *N (T :* nombre d'observations, N : nombre d'états total des modèles) appelé *treillis des hypothèses* où chacun de ses nœuds *(t,i)* contient la vraisemblance $\delta_i(o_t)$ du meilleur chemin passant par l'état *i* à l'instant *t.* La vraisemblance $\delta_i(o_T)$du meilleur chemin qui finit à l'état *i* au temps T est alors calculée par récurrence :

1. **Initialisation :** $\delta_i(o_1)= \pi_i$
2. **Récursion** : pour se trouver dans l'état *i* à l'instant t, le processus markovien se trouvait forcément dans un état *j* à l'instant $t - 1$ pour lequel une transition vers l'état *i* est possible : $a_{ji}>0$.

   D'après le principe d'optimalité de Bellman, $\delta_i(o_t) = \max_j (\delta_i(o_{t-1}).a_{ji}). bi(o_t).$
3. **Terminaison** : La vraisemblance des observations correspondant à la meilleure hypothèse est obtenue en recherchant l'état *i* qui maximise la valeur $\delta_i(o_T)$ à la dernière observation o_T :

$$P(O|Q^*) = \max_i(\delta_i(o_T))$$

5.3 Limitation des HMM

L'utilisation des HMM en reconnaissance automatique de la parole repose sur plusieurs hypothèses simplificatrices. Celles-ci sont, certes, nécessaires, mais elles constituent également des points faibles des HMM.

La modélisation de la durée des phonèmes n'est qu'implicitement contenue au travers des probabilités de transitions entre les états. Une modélisation explicite de celle-ci a cependant été proposée avec succès [Russel 85], [Levinson 86].

L'hypothèse d'indépendance conditionnelle des observations est irréaliste. Une solution efficace et largement répandue consiste à prendre en compte les dérivées premières et secondes des paramètres. Une deuxième solution est de modéliser explicitement la corrélation entre les vecteurs d'observations successifs [Russell 93], [Gales 93b].

6 Conclusion

La reconnaissance automatique de la parole pose de nombreux problèmes d'un point de vue théorique. Leur complexité fait que seuls des sous-problèmes ont pu être résolus. Ces solutions partielles correspondent à des contraintes plus ou moins fortes, et les systèmes existants supposent une coopération plus ou moins grande des utilisateurs.

Ainsi, les informations contenues dans un message parlé se traduisent en paramètres et en phénomènes mesurables. La voix est un signal acoustique résultant du passage de l'air par le conduit vocal (cordes vocales du larynx, pharynx, cavité buccale et cavité nasale), de sa modification par un environnement (écho, bruit, canal...), et de sa perception par un capteur (oreille, micro). Ce signal est tout d'abord caractérisé par des paramètres perceptifs comme par exemple la hauteur, le timbre, le rythme, la vitesse, la clarté ou le tremblement. Au niveau physique, ces paramètres peuvent être observés par étude du spectre, une analyse fréquentielle du signal par rapport au temps.

Pouvoir accéder au contenu lexical (et peut être sémantique) du message, véhiculé par la voix, suppose plusieurs étapes de décodage et d'analyse.

La recherche d'information audio consiste donc en une analyse des propriétés acoustiques du signal audio pour déterminer ce qui pourrait intéresser l'utilisateur. La reconnaissance de la parole étant un domaine clef dans les systèmes de recherche d'information audio.

Toutefois, en recherche d'information le but ultime est non pas de transcrire un message audio, mais de retrouver des documents qui répondent à un certain nombre de critères. Parmi ces critères de recherche, on retrouve les mots clés où étant donné un ensemble de documents, on retient ceux qui contiennent les mots clés définis. Dans le chapitre suivant nous expliquons comment se fait la détection de mots clés dans un document audio.

Chapitre 3 :
La détection de mots clés

1 Introduction

Parmi les applications les plus importantes de la reconnaissance automatique de la parole, on peut citer les accès automatiques à des services d'information telle que la commande vocale et les systèmes de dialogue.

Les recherches se sont orientées vers le développement des systèmes qui ont comme entrée la parole spontanée prononcée par différents locuteurs et dans des environnements divers [Cole et al 95]. Ceci pose de nouveaux problèmes, comme l'introduction des mots Hors Vocabulaire (HV), les faux départs, les hésitations, etc. Dans ces situations, le système de reconnaissance doit être capable de détecter et de reconnaître les mots clés de l'application et de rejeter les mots Hors Vocabulaire.

Les systèmes de reconnaissance équipés par des processus de détection de mots clés permettent aux utilisateurs de parler librement et naturellement, sans besoin de s'exprimer avec un format rigide. Un système de reconnaissance idéal doit accepter la parole spontanée et générer une transcription précise de la phrase d'entrée en temps réel.

Dans ce chapitre nous présentons les méthodes utilisés dans la construction des systèmes de détections des mots clés. Nous exposons les techniques de constructions des modèles poubelles, ensuite nous abordons les différentes mesures de confiances. Enfin nous citons quelques applications dans le domaine de la détection des mots clés.

2 Les systèmes de détections de mots clés

La détection de mots clés est la tâche d'identification des occurrences de certains mots recherchés dans un signal de parole arbitraire.

Les systèmes de détection de mots clés ont fourni une solution aux processus de la parole spontanée [Wilpon et al 90] [Rose 95]. En effet, le but des systèmes de reconnaissance automatique de la parole est de trouver une transcription exacte de tous les mots prononcés dans une phrase, alors que les systèmes de détection de mots clés essayent de détecter seulement les mots qui ont une importance pour l'interprétation sémantique de la phrase et qui sont définis précédemment dans le vocabulaire des mots clés.

Il suffit donc que les mots clés précédemment définis dans le vocabulaire soient détectés s'ils sont prononcés dans la phrase.

Le problème de détection de mots clés dans un flux de parole a été traité suivant deux approches qui sont les modèles poubelles et les mesures de confiance.

3 Modèle poubelle

La structure générale d'un système de détection de mots clés peut être décrite comme une combinaison de mots clés et de séquences de parole (ou éventuellement de bruit) constituées de mots non-clés. Un système de détection a pour rôle d'une part la modélisation des mots non-clés afin de réduire les fausses acceptations et d'autre part la modélisation des mots clés pour améliorer leur détection. La performance d'un tel système de détection de mots clés dépend essentiellement de la manière avec laquelle il peut accomplir ces deux tâches.

Dans un système de détection de mots clés, il est souhaitable de réaliser un taux élevé de détection, avec une minimisation du nombre de fausses acceptations. Dans ce cas, il n'est pas suffisant de modéliser seulement les mots clés, mais la modélisation des mots HV est aussi nécessaire. Une approche commune pour la tâche de détection de mots clés est l'emploi d'un ensemble de modèles poubelles pour concurrencer les modèles de mots clés. Les modèles poubelles doivent être adaptés aux mots HV (mots non-clés), aux bruits, aux sons non parole, aux faux départs, etc. Le rôle de ces modèles poubelles est d'absorber tous les mots HV.

L'un des premiers systèmes de détection des mots clés utilisant les modèles HMM pour la reconnaissance de la parole continue a été proposé par Rose [Rose et Paul 90]. Le principe de ce système repose sur la séparation entre mots clés et mots poubelles. Il a construit un réseau constitué de N mots clés et M mots poubelles (**figure 3.1**). Le point d'opération du système peut être ajusté par la disposition des poids des transitions w_1, w_2, …, w_N pour les mots clés et

f_1, f_2,…,f_M pour les mots poubelles. Dans ce contexte, ce point d'opération réfère à un compromis entre le nombre de faux rejets et celui de fausses acceptations.

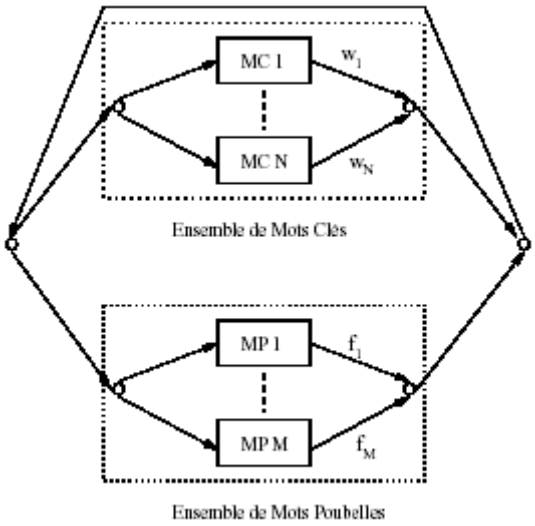

Figure 3.1 : Description du système de détection de mots clés basé sur l'utilisation d'un réseau de mots clés et de mots poubelles.

Le modèle poubelle en ligne est introduit par [Boite et al 93] et [Bourlard et al 94] pour décrire la création d'un modèle poubelle local comme étant la moyenne des n-meilleurs vraisemblances de chaque trame dans la séquence d'observations.

Lleida [Lleida et al 93] a mené une étude comparative sur différentes formes de modèles poubelles en introduisant des modèles poubelles aux niveaux phonème, syllabe et mot. En premier lieu, les modèles poubelles sont représentées par des phonèmes indépendants du contexte. En deuxième lieu, il a groupé les syllabes en quatre classes, chaque classe regroupe les syllabes les plus ressemblantes. Ensuite il a fait toutes les combinaisons possibles de ces quatre classes de syllabes afin d'extraire 60 modèles poubelles syllabiques. En troisième lieu il a construit trois modèles poubelles à base de mots. Le premier modèle est pour les mots monosyllabiques, le deuxième est pour les mots bisyllabiques et le troisième est pour les mots qui contiennent trois syllabes ou plus.

4 Mesure de confiance

Le système de détection de mots clés risque toujours de produire des insertions, des substitutions et des omissions, ce qui déclenche parfois des fausses acceptations, tout en présentant des non détections. La régulation de ces opérations se fait par la génération des

hypothèses de mots et l'association d'un score à chaque mot. Pour éviter les erreurs de détection, il faut être capable de générer suffisamment d'hypothèses et donc de résoudre le problème d'omissions de mots clés. Ensuite, vient l'étape de la vérification qui permet d'accepter seulement les mots clés en se basant sur une mesure de confiance pour rejeter les mots incorrectement reconnus. Dans des situations particulières où la parole prononcée ne contient pas de mots clés ou il y a une grande confusion entre les mots clés, un grand taux de substitution peut être observé. Pour remédier à ce problème, le système de reconnaissance doit être capable de reconnaître les mots clés incorporés dans la parole et de les vérifier ensuite afin de rejeter les mots qui ont un score de confiance faible.

La sortie du système de reconnaissance doit être modifiée afin d'extraire différents scores qui peuvent être utilisés pour générer des vecteurs caractéristiques qui seront à leurs tour utilisés dans le processus de vérification. Chacun de ces scores représente le niveau de confiance assigné à chacun des mots et il est capable de faire la discrimination entre les mots clés corrects et ceux insérés et donc de réduire le nombre de fausses acceptations.

Plusieurs chercheurs ont travaillé sur la détection des mots clés en utilisant des mesures de confiance [Rivlin et al 96] [Weintraub et al 97] [Moreau et Jouvet 00] [Zhang et Rudnicky01]. Le processus de vérification devient un problème de classification, dans lequel chaque mot clé doit être classé comme correct s'il correspond à un mot clé correctement détecté ou incorrect s'il correspond à une fausse alarme. Le processus de classification des mots clés reconnus se base sur l'utilisation d'un score discriminant associé à chaque mot et la définition d'un seuil d'acceptation pour les mots clés.

4.1 Programmation dynamique

L'approche historique de la détection de mots clés est basée sur l'algorithme de recalage temporel Dynamic Time Warping (DTW). Il s'agit d'un algorithme de reconnaissance, mais il convient bien à la détection de mots clés grâce à son estimation de distance. On peut l'appliquer en relâchant les contraintes sur les régions de commencement et de fin de l'échantillon de parole [Myers et al 80] [Rabiner et Schmidt 80] [Myers et Rabiner 81]. La méthode consiste à faire coïncider la séquence de parole à l'entrée du système, avec une séquence de mots de référence concaténés entre eux [Higgins et Wohlford 85]. Une alarme se déclenche lorsque la séquence qui correspond le mieux au signal d'entrée contient un mot clé. Pour calculer le meilleur alignement, on utilise généralement l'un de ces trois algorithmes

récursifs : one-pass, level-building ou two-level, qui sont presque équivalents dans leur version synchronisée temporellement [Godin et Lockwood 89] [Bezie et Lockwood 93]. Ces algorithmes récursifs stockent à tout moment les correspondances entre chaque référence et chaque hypothèse de commencement pour les suites de vecteurs d'échantillons.

La programmation dynamique, en tant qu'algorithme d'alignement, sert notamment à la détection de mots clés à base de treillis de phonèmes. [James et Young 94] a utilisé un système de reconnaissance HMM basé sur un algorithme de Viterbi modifié afin d'obtenir une décomposition intermédiaire compacte de chaque expression. Cette forme intermédiaire est un treillis de phonèmes dans lequel plusieurs hypothèses de phonèmes sont stockées pour chaque point à travers la parole. L'étape de détection devient une programmation dynamique symétrique qui essaye de faire la correspondance entre les prononciations des mots clés et le treillis avec des pénalités pour les phénomènes d'insertion, d'omission et de substitution.

Silaghi et Bourlard [Silaghi et Bourlard 00] proposent une nouvelle approche pour la détection de mots clés sans modélisation explicite des mots non clés. Cette approche est basée sur la technique de mesure de confiance en utilisant les probabilités a posteriori locales. Elle cherche le segment de parole qui maximise l'observation moyenne a posteriori calculée sur le chemin le plus probable. Ce problème est généralement résolu avec un processus de programmation dynamique très complexe. Les auteurs proposent une nouvelle forme itérative de l'algorithme de décodage de Viterbi (appelé IVD pour Iterating Viterbi Decoding) afin de résoudre ce problème d'optimisation avec un simple processus de programmation dynamique.

4.2 Algorithme de Viterbi et de Baum-Welch

Le modèle HMM peut être utilisé pour réaliser la reconnaissance ou la détection de mots clés. La seule différence revient dans le choix du critère de décision. Dans le cas de la reconnaissance de la parole, le but est de trouver la séquence d'observations la plus proche d'un modèle donné. Ceci est approximé par la recherche de la meilleure séquence d'états pour trouver le mot correspondant en utilisant l'algorithme de Viterbi. Pour la détection de mots clés, le but est de déterminer la probabilité qu'un mot clé soit présent à un instant donné.

4.3 Utilisation des traces d'alignements

D'autres travaux ont été réalisés en se basant sur l'utilisation des traces d'alignements. La trace du modèle sur le signal est constituée des informations qui proviennent des statistiques sur l'utilisation des fonctions de densité de probabilité le long du chemin optimal de Viterbi. Parmi ces travaux [Mathan 91] a étudié une technique de rejet basée sur la classification des traces d'alignement (post-traitement). Les traces utilisées contiennent le nombre et la valeur moyenne des trames observées par gaussienne le long du chemin optimal, le score obtenu par le meilleur modèle et la différence de score entre les deux meilleurs modèles. La décision d'accepter ou de rejeter la trace est réalisée par un réseau connexionniste classifieur binaire (PMC : Perceptron Multi-Couches) à deux sorties, l'une pour l'acceptation, l'autre pour le rejet.

Manuuary avec une modélisation à base d'allophones, a également testé l'utilisation de traces, limitées aux durées de réalisation de chaque phonème (c'est-à-dire le nombre de trames mises en correspondance avec les gaussiennes de chacun des phonèmes). La technique s'est avérée moins efficace sur les données d'exploitation, cependant elle a réalisé de bonnes performances de rejet en effectuant une classification des traces par l'algorithme des plus proches voisins[Mauuary 94].

4.4 Apprentissage discriminant

Sukar et Wilpon introduisent un système hybride qui consiste en une analyse discriminante à deux étapes. La première étape utilise un apprentissage basé sur la descente probabiliste généralisée (GPD). La seconde étape réalise une combinaison linéaire, de la sortie de la première étape avec les scores obtenus par les HMM. Ainsi, le pouvoir de discrimination du HMM est combiné avec celui de la GPD, ceci sera utilisé pour réaliser la classification mot clé/mot non-clé [Sukkar et Wilpon 93].

Dans la formalisation originale, la GPD a été développée pour minimiser les erreurs de reconnaissance en optimisant les paramètres du HMM [Chou et al 92]. La GPD définit en premier lieu une fonction de distance et en second lieu, une fonction de perte (fonction de la fonction distance) qui doit être minimisée. L'intérêt de cette fonction est d'accentuer la séparation de classes dans le cas où ces dernières sont fortement corrélées.

4.5 Seuil sur les scores de reconnaissance

Certains travaux sont basés sur un seuil et dans lesquels le rejet ou l'acceptation se fait en comparant le score d'un mot à ce seuil. [Mazor et Feng 93] décrit l'application d'un seuil aux diverses quantités, en incluant la vraisemblance des hypothèses des mots clés et la différence entre la vraisemblance de la meilleure hypothèse de mots clés et de celle qui la suit dans la tâche de vérification.

Rivlin est le premier à utiliser une approche basée sur l'estimation de la probabilité a posteriori du phonème. Il utilise le logarithme de ces probabilités a posteriori sur un intervalle d'hypothèses de phonèmes afin de calculer une mesure de confiance au niveau phonème. Une mesure de confiance au niveau du mot est créée en utilisant les mesures des phonèmes qu'ils constituent[Rivlin et al 96].

[Bayya 00] propose d'effectuer le rejet des mots HV en utilisant une mesure de confiance qui est une fonction de la distance entre le meilleur score obtenu et les K-meilleurs scores suivants.

[Moreau et al 00] propose une méthode qui consiste en un post-traitement des hypothèses de reconnaissance par le calcul d'une mesure de confiance pour chaque hypothèse. Cette mesure est basée sur le rapport de vraisemblance au niveau le plus élémentaire qui est le niveau des trames acoustiques.

4.6 Méthodes d'adaptation

Pour réaliser la tâche de la détection de mots clés, quelques chercheurs ont travaillé sur la stratégie d'adaptation. [Gupta et Soong 98] montre qu'un seuil adaptatif, basé sur la longueur des expressions, fournit une amélioration par rapport au seuil statique pour une tâche de reconnaissance de chiffres.

[Moreau et al 00] résout le problème de rejet des données incorrectes dans une tâche de grand vocabulaire par l'utilisation d'un algorithme d'adaptation incrémentale pour adapter les modèles des mots et ceux des poubelles. L'approche qu'il a proposée utilise un modèle poubelle pour capturer les mots hors-vocabulaire, il étudie l'impact de l'adaptation des modèles des mots et des modèles poubelles aux données du domaine qui sont collectées durant la phase d'exploitation du système. Il suppose que l'utilisation des données spécifiques

à la tâche améliore les performances du système de reconnaissance et particulièrement sa capacité de rejeter les mots incorrects. La technique proposée est basée sur l'algorithme EM de segmentation incrémentale.

L'adaptation incrémentale est utilisée dans ce cas afin d'adapter un modèle HMM aux données du domaine. Cette adaptation consiste en l'estimation de nouveaux paramètres des fonctions de densités de probabilités par une procédure itérative à deux étapes. La première consiste à aligner les trames d'apprentissage sur le modèle donnant la séquence optimale. La deuxième calcule les nouveaux paramètres du modèle en utilisant la base complète (la base initiale X(Init) et la base d'apprentissage X(Adapt)).

4.7 Connaissances acoustiques et linguistiques

En plus de l'utilisation des méthodes basées uniquement sur des connaissances acoustiques, certains travaux combinent les deux types de connaissances, acoustiques et linguistiques. En effet, la partie linguistique présente aussi une information significative et il arrive parfois qu'un mot ayant un score acoustique faible soit correctement reconnu grâce au modèle de langage utilisé. Les mesures de confiance acoustiques s'avèrent donc, dans de tels cas, insuffisantes. Rose [Rose et al 98] propose une nouvelle méthode qui combine les deux types de connaissances : linguistiques et acoustiques. Dans cette approche, on incorpore la notion de mesure de confiance acoustique dans l'automate stochastique utilisé pour décrire le modèle de langage n-gram du système de reconnaissance [Riccardi et al 96]. Dans un cas simple, un état de cet automate peut correspondre au contexte du mot w_i et le poids d'un arc peut correspondre à la probabilité de produire w_i sachant le mot précédent. La méthode proposée étend la notion d'état pour inclure non seulement le contexte, mais aussi une représentation discrète de la confiance acoustique c_i correspondante à l'histoire du mot w_i. On ajoute donc un état qui correspond à la confiance acoustique étendant ainsi l'espace des états de l'automate stochastique considéré.

[Hernandez-Abrego et Marino 00] explore l'influence des informations contextuelles sur les mesures de confiance pour les résultats de la reconnaissance de la parole continue. Il a proposé une approche à trois étapes. Tout d'abord, il effectue l'extraction de trois mesures de confiance acoustiques à la sortie des résultats de reconnaissance. Ensuite, ces mesures sont compilées à l'aide d'un système d'inférence flou qui prend en entrée ces trois types de mesures de confiance et fournit en sortie une seule mesure de confiance acoustique floue comprise

entre 0 et 1. Les paramètres de ce moteur sont estimés directement à partir des exemples avec une stratégie d'évolution. Enfin, au niveau du modèle de post-traitement, on intègre l'information linguistique qui va être utilisée pour ré-estimer la mesure de confiance de chaque mot w_i. Cette nouvelle mesure de confiance est calculée comme étant le produit de la mesure acoustique fournie par le moteur d'inférence et un coefficient de proportionnalité S(wi)

4.8 Réseaux de neurones

L'une des méthodes les plus importantes en reconnaissance automatique de la parole est l'utilisation les réseaux de neurones artificiels (RNA). Ces derniers ont été utilisés aussi dans la détection de mots clés en se basant sur les Réseaux de Neurones Artificiels. [Morgan et al 91] introduit l'utilisation des RNA dans la recherche de mots clés ; il utilise un système standard pour détecter les mots clés et obtenir les régions susceptibles de contenir un mot clé. Ensuite, il utilise un réseau de neurones pour la décision. [Clary et Hansen 92] utilise un RNA pour combiner les vecteurs acoustiques successifs afin de créer un nouveau vecteur, qui sera utilisé par la suite dans un modèle de Markov semi-continu pour la détection des mots clés. [Zeppenfeld et Waibel 92] utilise un RNA temporel (TDNN) avec les méthodes classiques de programmation dynamique. Il emploie pour chaque mot clé un RNA dont les sorties sont fournies à un algorithme de programmation dynamique pour faire la détection de mots clés. La quantification vectorielle et le mélange de gaussiennes peuvent très bien s'appliquer à la détection de mots clés en les utilisant comme facteurs de complément et d'optimisation [Tadj 95].

[Bernardis et Bourlard 98] utilise un système hybride HMM/RNA. Il montre que l'utilisation de la probabilité a posteriori locale accumulée (obtenue à partir de la sortie d'un Perceptron multi-couches) normalisée par le nombre de trames contenus dans le segment du mot a réalisé de bonnes mesures de confiance et de bons scores pour la ré-estimation des N-meilleurs hypothèses. Ceci est confirmé par [Williams et Renals 97] qui utilise le même type de mesures de confiance en les comparant avec plusieurs autres approches.

[Charlet et al 01] présente une technique de combinaison de mesures de confiance utilisant un Percepton Multi-Couches (PMC). Dans cette approche, on suppose que chaque segment de parole composant une hypothèse de reconnaissance w (un mot ou une séquence de mots) est décrit par un ensemble de caractéristiques phonétiques comme voisé/non-voisé,

voyelle/consonne etc. On calcule alors un score représentant chacune de ces caractéristiques qu'on combine après en utilisant un PMC afin d'extraire un score global pour cette hypothèse de reconnaissance.

Les paramètres du RNA sont appris en utilisant l'algorithme de rétro-propagation afin de minimiser l'erreur quadratique moyenne. Durant l'apprentissage, la sortie est soit 0 soit 1, selon la présence ou l'absence théoriques de la caractéristique phonétique pour le phonème associé au segment phonétique.

4.9 Transformations et algorithmes

D'autres études incluent l'exécution de quelques transformations linéaires et de quelques algorithmes pour l'extraction du vecteur caractéristique et pour la détection de mots clés.

[Kamppari et Hazen 00] présente une technique de calcul de la mesure confiance au niveau du mot fondée sur une combinaison de plusieurs caractéristiques, elles-mêmes basées seulement sur les informations acoustiques extraites d'un classifieur phonétique. Il utilise l'analyse discriminante linéaire de Fisher afin de fusionner l'ensemble des caractéristiques acoustiques en un simple score de confiance en utilisant une projection linéaire.

[Vergyri 00] décrit un processus de post-traitement qui traite les caractéristiques au niveau des mots comme sources de connaissances indépendantes et les combine dans un seul modèle logarithmique linéaire pour calculer la probabilité a posteriori de la séquence de mots. Ce modèle est utilisé pour calculer le score de l'hypothèse. Les paramètres de ce modèle sont optimisés à l'aide d'une approche de combinaison de modèles discriminants. Cette méthode utilise elle même une méthode d'optimisation simplex afin de minimiser la fonction du taux d'erreur empirique sur la base d'apprentissage.

5 Les applications

Les recherches réalisées dans le domaine de la détection de mots clés dans un flux de parole visent généralement à faciliter l'interaction entre l'homme et la machine en détectant les mots les plus intéressants pour l'interprétation sémantique de ce qui a été prononcé. Les applications dans ce domaine sont nombreuses.

Par exemple les opérateurs de service automatique [Sukkar et Wilpon 93] où l'utilisateur demande le type de service qu'il souhaite (apprendre les prix des actions du marché ou

interroger une base de donnée téléphonique : adresse et numéro de téléphone d'une personne, hôtel etc.), les systèmes de routage téléphonique ou de classification des documents parlés etc. [Wilpon et al 90] [Gorin et al 97].

Une autre application est l'indexation audio. Dans cette tâche, le système doit classer les vidéos, les enregistrements audio et les événements vidéo par leurs contenus [Jones et al 95] [Gelin et Wellekens 96]. La classification est réalisée en se basant sur des occurrences suffisantes de mots clés qui caractérisent correctement le domaine de l'enregistrement audio. Ainsi, l'utilisateur est capable de faire un balayage sur une très grande base audio et d'extraire l'information appropriée sans des connaissances explicites de tout le contenu de la base de données.

Une troisième application est la surveillance des conversations téléphoniques. Dans ce cas, la tâche principale consiste à chercher des mots clés importants pour comprendre le contenu de la conversation [Yapanel 97].

L'initialisation de l'interaction homme-machine est une autre application des systèmes de détection de mots clés. Par exemple, pour des gens handicapés on peut fournir quelques appareils électroménagers qui peuvent répondre à des mots spécifiques, donc le système détecte ces mots clés en négligeant tous les autres entrées acoustiques et répond à la demande [Pols 97].

6 Conclusion

En détection de mots clés, le but est de reconnaître les mots clés dans un flux de parole continue, indépendamment du locuteur. La détection de mots clés résout quelques problèmes liés à la reconnaissance de la parole continue comme les hésitations, les faux départs, les phrases grammaticalement incorrectes, les phrases tronquées, etc. Cependant, elle introduit de nouveaux types de problèmes notamment ceux de fausses acceptations et de faux rejets.

Le problème de détection de mots clés peut être aussi traité en utilisant une mesure de confiance. La notion de mesure de confiance a été employée par plusieurs chercheurs afin de réaliser des systèmes de détection de mots clés. Cette mesure peut être calculée en utilisant différents types d'informations telles que les informations acoustiques et linguistiques. Dans la

littérature, plusieurs méthodes ont été proposées pour calculer différentes mesures de confiance.

Cette grande diversité de méthodes prouve l'importance de la technique de la mesure de confiance permettant de prendre la décision d'accepter ou de rejeter un mot reconnu.

Chapitre 4 : Détection de mots clefs par les îlots de confiance

1 Introduction

Avec les méthodes classiques, nous remarquons que les performances du système de détection sont fortement reliées à la qualité de modélisation des mots poubelles or cette vision des choses peut sembler paradoxale, vu que c'est de la qualité de modélisation des mots clefs que doit dépendre le système final.

La solution proposée se veut une alternative à la fois aux modèles poubelles et à la détection du début et la fin d'un mot. Au lieu de rechercher les frontières d'un mot, nous partons de régions particulières que nous appelons « îlots de confiances ». Ces régions dont la reconnaissance est quasi-certaine représentent des blocs autour desquels des trous seront remplis.

En effet, au niveau phonétique, il y'a des sons particuliers qui nous interpellent dans un mot, ainsi si nous recherchons le mot [musafirun], c'est le son [s] qui semble guider notre recherche, transposant cette approche au niveau acoustique, et essayant de caractériser les mots clefs de manière à définir des caractéristiques discriminantes qui serviront à les localiser.

Ainsi, la détection d'un mot clef est réalisée par la détection d'un son particulier dans le mot c'est-à-dire d'une unité infra-lexicale, et à partir de ce « puit d'ancrage », on peut chercher dans l'intervalle considéré le mot à reconnaître. Dans notre proposition ce sont ces unités infra-lexicales qui font office d'ilots de confiance. Lorsque ces unités sont détectées dans le discours un appariement avec les modèles des mots clefs qui contiennent ces unités est effectué.

La méthodologie que nous avons adoptée pour arriver aux résultats discutés dans le prochain chapitre, s'articule autour de trois phases :

- Le traitement des fichiers son
- La définition et la caractérisation des mots clefs
- La détection des mots clefs

2 Traitement des fichiers son

Pour pouvoir évaluer notre approche, nous avons besoin d'enregistrements de paroles effectués dans différents environnements pour construire un corpus de test. Pour cela nous avons utilisé les fichiers audio disponibles par l'association internationale de phonétique [Web01], ainsi qu'un ensemble d'enregistrement effectués localement à différentes périodes, ce qui devrait garantir la diversité des sources, des locuteurs et des conditions d'enregistrement; c'est la base BAp.

Comme nous utilisons des fichiers son, ces derniers étant des signaux acoustiques nous devons en premier lieu faire une analyse de ces signaux.

2.1 Analyse du signal

Le but de l'analyse d'un signal est de quantifier certaines grandeurs qui caractérisent un son, à savoir l'amplitude, la durée et le timbre. Cette analyse permet de représenter le signal par des vecteurs de coefficients qui sont calculés sur des intervalles de temps.

a) Echantillonnage

Le but de l'échantillonnage est la numérisation ou la discrétisation d'un signal continu. En effet, les informations acoustiques d'un signal de parole se situent dans la bande passante [50hz, 8khz]. Notre signal doit être échantillonné à une fréquence supérieure ou égale à 16khz d'après le théorème de Shanon, parce que les enregistrements utilisés ont été faits dans le laboratoire.

L'échantillonnage est réalisé par une segmentation du signal de la parole en trames de longueur fixe qui se recouvrent entre elles.

Dans le cas de notre étude, nous avons échantillonné par une segmentation en trames de 512 échantillons avec un recouvrement de 100 échantillons.

b) Préaccentuation

Cette opération est effectuée pour relever les hautes fréquences qui ont une amplitude moindre que les basses fréquences. Le paramètre de préaccentuation utilisé prends sa valeur dans l'intervalle]0.9,1[.

c) Fenêtrage

Le signal est multiplié par une fenêtre rectangulaire afin de réduire les oscillations et les discontinuités créées par la segmentation en trames.

Il existe plusieurs méthodes de fenêtrage comme nous l'avons présenté dans le chapitre reconnaissance de la parole. Nous avons utilisé le fenêtrage de Hamming.

2.2 Calcul des coefficients

Une fois que le signal a été mis en forme, l'étape suivante dans l'analyse est le calcul des coefficients cepstraux. Il existe dans la littérature deux méthodes utilisées dans la reconnaissance automatique de la parole : les coefficients Mel frequency cepstral coefficients (MFCC) et les coefficients Linear Prediction cepstral coefficients (LPCC).

Pour notre travail nous avons préféré la méthode de Mel Frequency Cepstral Coefficients MFCC car nous voyons deux avantages à son emploi. La première qualité de la méthode MFCC est sa résistance reconnue au bruit. Certaines études ont ainsi montré que les MFCC étaient plus résistants que les coefficients log-spectraux avec lesquels ils partagent pourtant quelques caractéristiques computationnelles . La deuxième qualité majeure de la méthode MFCC est sa plausibilité biologique puisqu'elle utilise une échelle psycho acoustique des fréquences similaires à celle de l'oreille

Pour chaque trame du signal, on calcule la transformée de Fourrier rapide (FFT), puis un lissage est effectué par l'application d'un filtre triangulaire suivant l'échelle de MEL. Enfin la transformée de Fourrier inverse est appliquée pour obtenir un vecteur des coefficients cepstraux.

Après le calcul des coefficients, chaque trame est représentée par un vecteur de coefficients. Nous avons retenu les douze premiers coefficients ce qui est suffisant pour représenter une

trame d'un signal. Puis nous avons ajouté l'énergie ce qui constitue un vecteurs de treize coefficients par trame.

2.3 La quantification vectorielle

Vu le nombre important de trames obtenu à la phase d'échantillonnage du signal traité et du fait que chaque trame est représenté par un vecteur de coefficients, nous obtenons autant de vecteurs que de trames. Ceci va constituer une lourde tâche pendant la phase de détection.
Pour cela, nous avons procédé à un partitionnement des vecteurs en classes acoustiques par la méthode de la quantification vectorielle

Définition :

La quantification vectorielle (QV), est une méthode puissante qui consiste à utiliser les propriétés statistiques des sons dans leur espace de représentation. Cette technique rentre dans la problématique plus générale de la classification automatique et connaît actuellement de nombreux développements dans le domaine de la parole. Elle part du postulat que deux formes proches dans leurs espaces de représentation sont proches en soi.
Soit $x=[x_1, x_2, \ldots, x_k]$ un vecteur réel à valeurs continues dans R^k. La quantification de x consiste à lui substituer un vecteur voisin $y_i \in R^k$ (i=1,2,..,M) choisi parmi un ensemble fini de M vecteurs. Construire un système de QV, c'est opérer une partition de R^k en classes C_i ; dans chacune d'entre-elles, on distingue un vecteur particulier y_i appelé prototype ou centroïde (code-word). Chaque vecteur $x \in C_i$ sera représenté par le centroïde associé y_i. L'ensemble des centroïdes constitue un dictionnaire (code-book).
La substitution de x par le vecteur y provoque une distorsion notée d(x,y). Pour le codage des formes d'onde, on utilise le plus souvent l'erreur quadratique:

$$d(x,y)=\left[\sum_i (x_i-y_i)^2\right]/K \qquad (4.1)$$

Le but poursuivi dans l'établissement d'un système de codage est de minimiser la distorsion moyenne intra-classe et de maximiser la distance inter-classe.

Pour notre étude nous avons utilisé l'algorithme des nuées dynamiques (k-means) pour la construction de notre dictionnaire.

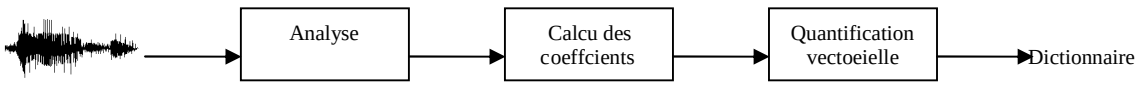

Figure 4.1 Etapes de construction du dictionnaire

3 Définition et caractérisation des mots clefs

Afin de pouvoir évaluer notre approche, nous avons défini un certain nombre de mots dans le corpus utilisé comme étant des mots clefs et nous avons constitué une base de mots clefs. C'est la base BMCAp.

3.1 Caractérisation acoustique

Chaque mot clef a été caractérisé par un ensemble de caractéristiques. Ces caractéristiques représentent un sous ensemble des caractéristiques qui constituent le dictionnaire.

La caractérisation des mots clefs est réalisée en utilisant les techniques de l'analyse de données qui nous permettent de déceler les corrélations entre un ensemble de caractéristiques acoustiques sur un signal est l'observation d'un mot donné. Il en résulte, un ensemble de règles de la forme : $C_i \wedge C_j \wedge \quad \rightarrow MC_l$.

Pendant la phase de détection, un mot clef est détecté si l'ensemble de ses caractéristiques sont trouvées dans un certain ordre.

Nous avons ainsi caractérisé tous les mots clefs qu'on avait définis.

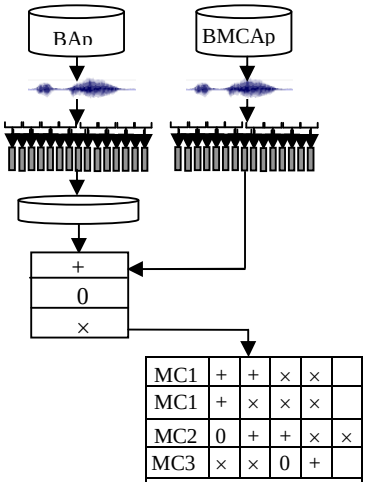

Figure 4. 2 : Les étapes de caractérisation d'un mot clef

3.2 Construction des modèles de référence :

Une fois avoir caractérisé tous les mots clefs, et avant d'aborder la phase de détection, nous devons construire un modèle de référence pour chaque mot clef de la base des mots clefs.

En effet, la validation de la détection consistera en un alignement du signal d'entrée avec le modèle de référence du mot clef.

Nous avons utilisé les modèles de markov cachés (HMM). Chaque mot clef de la base des mots clefs, a été représenté par un HMM.

Un modèle de markov caché (HMM) est un automate probabiliste caractérisé par :

- Un ensemble d'états
- Des transitions entre les états
- Des probabilités de transition entre les états

Les modèles HMM sont très utilisés en reconnaissance automatique de la parole, car ils sont bien adaptés à la modélisation de processus à évolution temporelle ce qui est le cas du signal de la parole.

3.3 Topologie du modèle

L'unité acoustique que nous avons choisie est le mot. Chaque mot clef de la base a été modélisé par un modèle HMM gauche droite. Chaque mot est d'abord décomposé en phonèmes et chaque phonème représente un état du HMM.

Nous avons construit des modèles comme suit :

- Un modèle HMM est caractérisé par des états, des transitions entre états et de probabilités de transitions.
- Un modèle HMM correspond à un automate d'états finis, où chaque état correspond à un phonème du mot clef.
- Les transitions entre états sont de gauche à droite (il n'y a pas de transitions du sens inverse car dans la parole il y a un aspect important à respecter qui est le temps).

La décomposition des mots clefs en phonèmes a été réalisée manuellement. La grammaire utilisée pour la transcription en phonèmes est présentée par le tableau TIMIT prédéfini dans [RES][Becchetti 99]. A titre d'exemple nous présentons la transcription phonétique le modèle HMM du chiffre cinq comme suit :

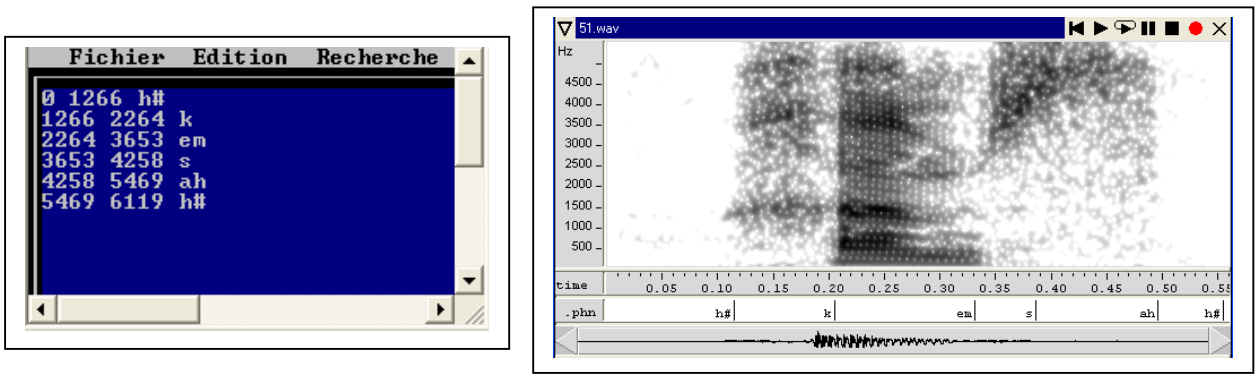

Figure 4.3 : Transcription en phonèmes du chiffre cinq (5)

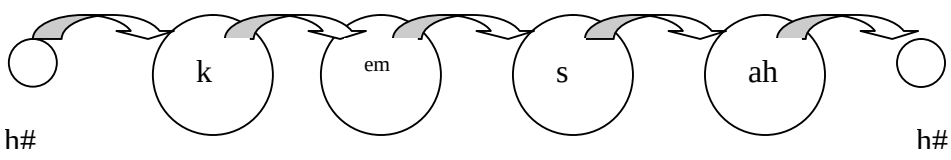

Figure 4.4 : Modèle du chiffre cinq (5)

4 La détection

La phase de détection consiste à trouver le mot clef cherché dans un flux d'entrée. Pendant cette phase, le flux d'entrée est segmenté en trames de 10s. Chaque trame est ensuite traitée et quantifiée. On recherche sur chaque trame, les caractéristiques du mot clef désigné. La détection est réalisée suivant l'algorithme :

1- Le signal d'entrée est découpé en fenêtre d 10s,
2- Sur chaque fenêtre on effectue une analyse du signal et on extrait ses caractéristiques acoustiques,
3- On applique une quantification vectorielle aux vecteurs des caractéristiques de la fenêtre analysée,
4- On recherche des caractéristiques du mot clef désigné, on enregistre les positions de la première et de la dernière caractéristique dans des variables ce qui permet de délimiter le début et la fin du mot clef recherché,
5- On effectue un alignement avec le modèle HMM de référence du mot clef

Nous avons utilisé les deux approches de reconnaissance : l'approche analytique puisqu'il y a la segmentation en phonèmes du mot clef prononcé puis l'approche globale car nous reconnaissons le mot dans sa totalité.

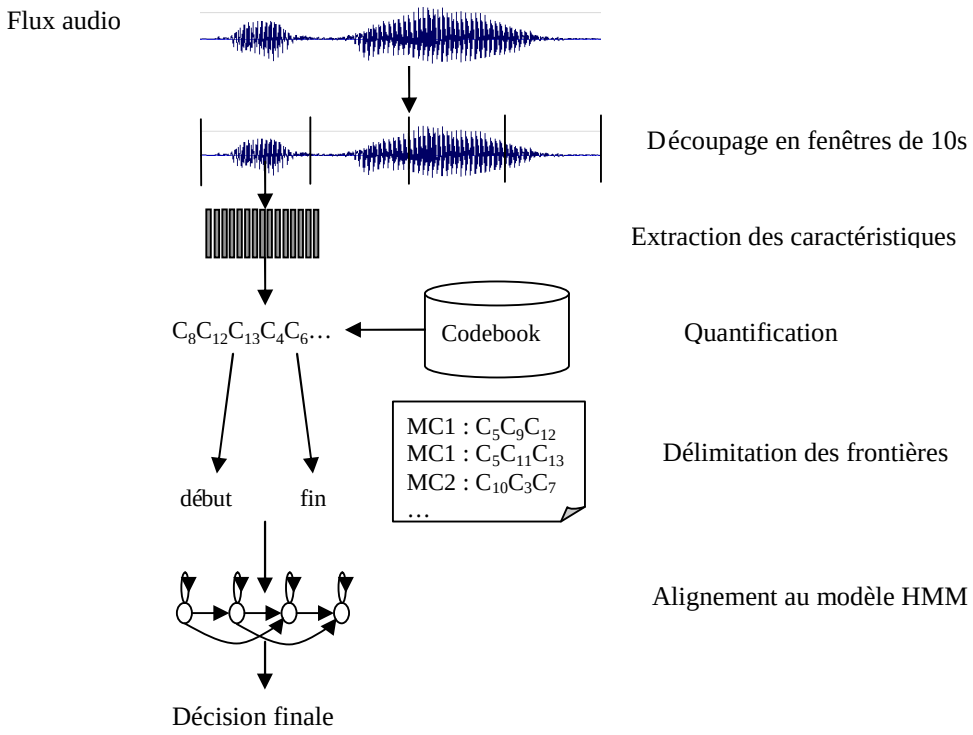

Figure 4.5 : Etapes du processus de détection

5 Conclusion

Dans ce chapitre nous avons présenté la démarche suivie pour mettre en œuvre une approche de détection de mots clef basée sur les ilots de confiance.

Nous avons d'abord expliqué la première phase qui nous a permis de traiter les fichiers audio du corpus de test et d'obtenir des vecteurs de caractéristiques. Ces vecteurs ont ensuite été quantifiés pour construire le dictionnaire.

Dans la deuxième phase, nous avons défini un ensemble de mots clefs que nous avons caractérisés puis modélisés par des modèles de markov cachés.

Dans une troisième étape, nous avons entamé la phase de détection d'un mot clef dans un flux de paroles en entrée et nous avons mis en œuvre un algorithme qui permet de rechercher un mot clef en retrouvant ses caractéristiques.

Pour l'évaluation de notre approche, nous avons réalisé un système de détection basé sur l'algorithme proposé. Le déroulement et les résultats obtenus lors de cette phase sont présentés dans le chapitre suivant.

Chapitre 5 : Evaluation de l'approche proposée

Présentation du chapitre

Dans ce chapitre nous allons présenter un plan d'évaluation de notre approche, ainsi que les principaux choix d'implémentation que nous avons effectué, pour le mettre en œuvre.

Le plan d'évaluation que nous proposons inclut deux aspects. D'abord, la recherche d'un mot clef dans des fichiers de durée variable, pour évaluer l'approche qualitativement. Ensuite, rechercher un ensemble de mots clefs dans un flux audio entrant de durée fixe, ce dernier aspect, nous permet d'avoir une estimation quantitative de la performance de l'approche.

1 Introduction

Nous nous situons dans notre application dans un cadre général de recherche d'information audio, et dans un contexte particulier de routage d'appels téléphoniques, ce qui peut évidement être étendu à d'autres applications de catégorisation de documents audio. Il s'agit de définir dans un flux de documents audio ceux qui sont relatifs à un sujet défini à l'avance. La catégorisation n'est pas notre souci dans ce mémoire, mais surtout le contexte dans lequel les mots clefs sont définis à l'avance. Dans une application comme la nôtre le nombre des appels téléphoniques est très grand ; il est impératif de disposer d'un outil de détection de mots clefs qui soit « rapide ». Nous avons proposé une méthode qui tente de localiser un mot clef dans un flux audio en se basant sur la détection de certaine de ces caractéristiques. L'algorithme proposé donne en sortie les frontières du mot clef dans le flux s'il est détecté; ce qui permet d'accélérer la recherche. Si l'application nécessite un grand degré de confiance la portion du signal sélectionnée est soumise à comparaison avec le modèle HMM du mot clef, avec dans ce cas une approche reconnaissance de mot isolé. Dans ce qui suit, nous détaillons les détails d'implémentation du système.

2 Extraction des caractéristiques

L'onde temporelle dans sa forme originelle ne peut subir un traitement algorithmique, pour cela, il faut la représenter sous une forme numérique. De plus, comme dans la plupart des applications de traitement du signal, il est nécessaire de procéder à une extraction de caractéristiques du signal afin que les traitements ultérieurs puissent être effectués.

2. 1 Echantillonnage

Les données se présentent sous forme des fichiers WAV, enregistrés à une fréquence d'échantillonnage variant de $f_e = 11025\ Hz$ *à* $2000Hz$ sur 16 bits.

En effet, la fréquence d'échantillonnage f_e doit être au moins égale au double de la fréquence maximale du signal à numériser (Théorème de Shanon); mais elle peut varier en fonction du domaine d'application ou des besoins ou contraintes matériels, à titre illustratif nous avons,

	Bande passante	Fréquence d'échantillonnage et précision
Téléphone	300-3500 Hz	8 KHz 8 bits
Voix	60-10000 Hz	16 KHz 16 bits
Hifi	10-18000 Hz	44 KHz 16 bits

Table 5.1 : Variation de la fréquence selon l'application

Si on se pose la question de pourquoi ne pas avoir utilisé une plus grande fréquence d'échantillonnage ou une plus grande précision. Sachant qu'un phone échantillonné à une fréquence de 8000Hz avec une précision de 8 bits préserve la plus grande partie de l'information apportée par le signal [Rabiner 93], il apparaît ne pas être nécessaire d'aller au delà de cette fréquence ou de cette précision. D'autre part, ce choix fût adopté pour pouvoir travailler sur la même base (en particulier en ce qui est des chiffres) que celle qui a servi à établir des résultats avec d'autres travaux et pouvoir ainsi effectuer une étude comparative.

Considérons le flux audio qui se présente sous la forme de la **figure 5.1**.

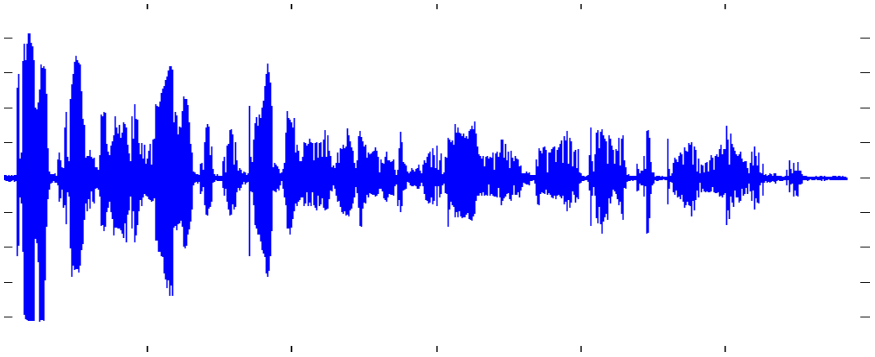

Figure 5.1 : Signal de la phrase narrative1

En réalité cette onde, échantillonnée à 20000Hz est représentée par un vecteur comprenant 116822 échantillons.

Par Exemple les valeurs de Y entre les échantillons 1530 et 1540 sont :

-0.0042
0.0009
0.0023
-0.0002
-0.0032
-0.0078
-0.0029
-0.0060
-0.0047
-0.0004
-0.0006

2. 2 Pré-accentuation

Comme il est commun dans les systèmes de reconnaissance de la parole, un filtre de pré-accentuation est appliqué au signal échantillonné. En effet, les caractéristiques des cordes vocales définissent les phonèmes prononcés. Ces caractéristiques sont visibles dans le domaine fréquentiel par la localisation des formants, i.e. les pics de résonance des cordes. Toutefois, les formants de hautes fréquences ont une amplitude moindre que celles des formants de basse fréquence. Une pré-accentuation des hautes fréquences est alors requise pour avoir des amplitudes similaires pour toutes les fréquences. Ce traitement est généralement obtenu en faisant passer le signal par un filtre FIR du premier ordre (pour plus

de détail voir [Caliope 89]), nous prenons un paramètre de pré-accentuation égal à 0.95, ce qui donne une amplification pour les hautes fréquences du signal de plus de 20 dB.

2. 3 Fenêtrage

Les méthodes classiques d'évaluation spectrales que nous avons présentée au chapitre 2, supposent que l'on opère sur un signal stationnaire (i.e. un signal dont les caractéristiques statistiques sont invariantes dans le temps). Pour la parole, ceci n'est vrai que pour des intervalles de temps courts durant lesquels une analyse peut être effectuée. Il faut alors bloquer le signal x'(n) en une succession de trames $x_t(n)$ avec t=1,2,…,T. Ces trames sont appelées fenêtres.

Une fois la fréquence d'échantillonnage fixée, augmenter la résolution spectrale revient à utiliser des séquences du signal plus longues, ce qui est en contradiction avec la nécessité d'avoir des segments stables du signal. Des études ont monté qu'une longueur entre 20-40 ms constitue un bon compromis.

Dans notre application nous choisissons des fenêtres de longueur N=512 échantillons ce qui correspond à un intervalle de temps de 512/11.025 ≈ 46ms. Avec un recouvrement de 100 échantillons entre les trames.

Ainsi, le signal de la figure 5. 1. segmenté produit 284 fenêtres temporelles de 512 échantillons chacune, dont la première commence par :

```
Columns 1 through 8

     0     0     0     0     0     0     0     0

 Columns 9 through 16

     0     0     0     0     0     0     0     0

 Columns 17 through 24

     0     0     0     0  -0.0004  -0.0004  -0.0002  -0.0003

 Columns 25 through 32

  0.0000   0.0004  -0.0001   0.0002   0.0003   0.0001   0.0002   0.0001

 Columns 33 through 40

  0.0002   0.0003   0.0002   0.0003   0.0005   0.0003   0.0002   0.0002

 Columns 41 through 48
```

```
  0.0000   0.0001   0.0004   0.0006   0.0009   0.0004  -0.0001   0.0002

 Columns 49 through 56

  0.0001  -0.0003  -0.0002   0.0002   0.0004   0.0005   0.0006  -0.0001

 Columns 57 through 64

  0.0004   0.0006  -0.0006   0.0002   0.0009   0.0007   0.0008   0.0005

 Columns 65 through 72

  0.0008   0.0007   0.0010   0.0020   0.0020   0.0021   0.0011   0.0011
...
```

2. 4 Fenêtrage de Hamming

Avant de calculer la transformée de Fourier du signal, chaque trame du signal est individuellement pondérée par une fenêtre. Nous utilisons une fenêtre de Hamming.

```
Boolean PreemphasisAndHammingWindow::Apply(VetDouble & smp_vet,t_real  smp_rate)
        {
        static VetDouble vhamm;

        if (vhamm.Dim()!=smp_vet.Dim())
                {
                vhamm.Destroy_And_ReDim(smp_vet.Dim());
                /*************************************************************/
                /*    CREATE HAMMING WINDOW VECTOR                           */
                /*************************************************************/
                const t_real a = 25.0/46.0;
                const t_real b = 1.0-a;
                t_real theta,arg;
                t_index i;
                theta = acos(-1.0);
                theta = 2.0*theta/(t_real)(smp_vet.Dim()-1L);
                for (i=1;i<=vhamm.Dim();i++)
                        {
                        arg = (t_real)(i-1L);
                        vhamm[i-1] = a - b*cos(theta*arg);
                        }
                }
        /*************************************************************/
        /*    COMPUTE CEPSTRUM FOR ALL THE FRAMES                    */
        /*************************************************************/
        t_real mem=0;
        t_real oldmem=0;
        t_index j;
```

```
// PREENPHATISES AND HAMMING WINDOWING
for(j=0;j<smp_vet.Dim();j++)
        {
        mem=smp_vet[j];
        smp_vet[j] = vhamm[j] *      (smp_vet[j] - oldmem* preemphasis);
        oldmem=mem;

        }
return TRUE;
}
```

2. 5 Analyse MFCC

De nos jours l'ensemble de caractéristiques le plus utilisé est celui des coefficients MFCC . Les coefficients MFCC (Mel Fequency Cepstral Coefficients) sont des coefficients cepstraux obtenus à partir des énergies d'un banc de filtres en échelle de fréquence Mel. Cette méthode présente l'avantage d'être résistante au bruit et d'avoir une plausibilité biologique puisqu'elle utilise une échelle psycho-acoustique des fréquences similaire à celle de l'oreille interne.

Une dizaine de coefficients cepstraux sont généralement considérés comme suffisants pour les expériences en reconnaissance de la parole. Nous utilisons les 12 premiers coefficients cepstraux obtenus à partir d'un banc de 26 filtres sur une échelle fréquentielle de Mel, le logarithme de l'énergie de la trame, ajoutée aux 12 coefficients cepstraux pour former un vecteur de 13 coefficients.

```
void CommonMfccAndFft::Perform_FftModule_And_Energy(VetDouble & vet_re,
                        t_real & energy)
{
static VetDouble vet_im;
t_index vet_dim;

Assert(feature_vet_dim>0);
Fft_Call(vet_re, vet_im);
vet_dim = vet_re.Dim()/2;

Magnitude_And_Energy_Of_Real_Vectors(vet_re, vet_im, energy);
vet_re.Save_And_ReDim(vet_dim);
if (compute_log_of_energy)
        energy=log(energy );
return;
}

Boolean MfccAndEnergy::Apply(VetDouble & data,t_real smp_rate)
        {
        static VetDouble temp;
```

```
t_real energy=0;

Perform_FftModule_And_Energy(data, energy);
temp=data;
data.Destroy_And_ReDim(feature_vet_dim);
Mfcc_Call(data,temp, smp_rate);
if(compute_energy)
        data.Append(energy);
return TRUE;
}
```

3 La quantification vectorielle

Une fois l'analyse du signal effectué, nous procédons à une quantification sur les vecteurs acoustiques obtenus car c'est à la base de cette quantification que se fera la détection des mots clefs. La quantification vectorielle nous sert dans notre méthode à faire ressortir les classes acoustiques, que nous considérons comme des caractéristiques du signal sur le plan symbolique (ou même perceptuel). En effet, les différentes classes issues de la quantification serviront à représenter un signal ce qui permet de réduire l'espace de représentation, et permettre ainsi une meilleure interprétation des évènements acoustiques.

3. 2 Etablissement des classes par la méthode de LLOYD généralisée (K-means method)

On dispose d'un ensemble L de vecteurs *x* que l'on désire positionner en *M* classes. On désignera par :

- $x_j^{(i)}$, les vecteurs appartenant à la classe i
- y_i , le centroïde de la classe i
- L_j , le nombre de vecteurs de la classe i
- $d (x_j^{(i)}, y_i)$, la distance ou mesure de distorsion entre $x_j^{(i)}$ et y_i

$$D_i = \sum_J d (x_j^{(i)}, y_i)$$

- D , la distorsion pour l'ensemble des vecteurs

$$D = \sum_{i=1}^{M} D_i$$

Un nombre *M* de classes étant imposé *à priori*, le problème consiste à trouver la partition et les centroïdes de façon à minimiser la distorsion totale *D*. Une procédure itérative peut être basée sur les deux observations suivantes:

- Pour un ensemble donné de centroïdes, la partition qui minimise *D* est celle pour laquelle chaque vecteur x_j est affecté à la classe dont le centroïde est le plus rapproché.
- Pour une partition donnée, il existe pour chaque classe *i* un vecteur *j* qui minimise la distorsion totale D_i de la classe *i*.

Dans nos différents tests, nous avons utilisé des dictionnaires de huit (8) et de seize (16) classes.

```
void Initiali::Symb_Model_Calculation(const String& out_fname, const String& in_fname,
                                      DbaseVoc& dbase, Boolean full_cov, Boolean
                                       load_one_mixture, Boolean unif_sect,
                                       const   ModelType model_type)
    {
    t_index num_mix;
    t_index num_frames;
    t_index phons_card, act_phon, i;
    ifstream one_gauss;

    if(load_one_mixture)
            {
            one_gauss.open(in_fname,ios::in|ios::nocreate);
            if(one_gauss.fail())
                            merr<<"Cannot open "<<in_fname;
            Read_Data_File_Header (one_gauss, stat_dim, full_cov);
            }

    Write_Header_Of_File_Model (out_fname, dbase.Snd_Type(),
      dbase.Label_Type(), dbase.Db_File_List_Name(), dbase.Window_Lenght(),
                    dbase.Window_Overlap(), stat_dim, full_cov);


    phons_card=dbase.Get_Num_Of_Symbols();

    for(i=0; i<init_symb_list.Dim();i++)
            {
            act_phon = init_symb_list[i];
            mstat<<"act_phon: "<<act_phon;  // monitoring elaboration
            states_info.Initialize(num_sections_per_symbol[i], num_mix_per_symbol[i],
                                        stat_dim, full_cov);
            Calculate_One_Mixture_Codebook(act_phon, num_frames, dbase, unif_sect);
            num_mix=1;
            if(load_one_mixture)
                    {
                    states_info.phoneme=act_phon;
                    states_info.stat_dim=stat_dim;
                    states_info.Read(one_gauss, full_cov);
```

```
                num_mix=states_info.num_gauss;
                }

        dbase.Restart();
        states_info.Compute_Whole_Codebook_Clusters_Distortions();
        while ( num_mix < num_mix_per_symbol[i] AND (num_frames!=0) )
                {
                mstat<<"Splitting of codebook number: "<<num_mix;
                Split_Higher_Distortion_Cluster_Of_Each_Section(act_phon,
num_frames, num_mix, dbase);
                states_info.Compute_Whole_Codebook_Clusters_Distortions();
                dbase.Restart();
                }
        states_info.Compute_Cluster_Weights();
        states_info.Store_Codebook(out_fname,  act_phon,  load_one_mixture,
model_type);
        }
    return;
    }
```

3.3 Construction du dictionnaire

Nos dictionnaires ont été construits sur la base de fichiers audio divers issus de milieu d'enregistrements divers et dans un contexte multi-locuteurs, et ce en raison des spécificités de l'application finale.

Voici, l'exemple de deux dictionnaires construits à l'aide de l'algorithme des k-means. Avec une génération selon la loi uniforme des classes initiales.

```
codebook =

  Columns 1 through 8

  -64.1790  -63.9529  -63.5771  -63.0530  -62.3825  -61.5681  -60.6128  -59.5204
  -43.6285  -43.4403  -43.1276  -42.6915  -42.1338  -41.4566  -40.6627  -39.7551
  -14.5536  -14.4097  -14.1706  -13.8374  -13.4115  -12.8949  -12.2899  -11.5992
    9.4774    9.5384    9.6397    9.7806    9.9605   10.1782   10.4325   10.7219
   31.8487   31.8251   31.7857   31.7304   31.6593   31.5721   31.4689   31.3494
   50.0164   49.9552   49.8532   49.7106   49.5276   49.3045   49.0414   48.7387
   75.1493   74.9975   74.7451   74.3926   73.9411   73.3919   72.7463   72.0064
  104.2173  104.0157  103.6803  103.2121  102.6121  101.8822  101.0242  100.0405

  Columns 9 through 14
```

-58.2947	-56.9404	-55.4625	-53.8665	-52.1582	0.0000
-38.7375	-37.6138	-36.3884	-35.0661	-33.6521	0.0001
-10.8259	-9.9734	-9.0456	-8.0466	-6.9808	0.0004
11.0448	11.3993	11.7834	12.1949	12.6313	0.0008
31.2135	31.0611	30.8920	30.7061	30.5032	0.0016
48.3969	48.0163	47.5976	47.1411	46.6476	0.0026
71.1740	70.2514	69.2413	68.1464	66.9696	0.0052
98.9337	97.7067	96.3628	94.9056	93.3390	0.0133

Il est évident que la taille du codebook utilisé influe énormément sur la qualité de la modélisation, car il ne faut pas choisir un nombre de caractéristiques restreint (taille du codebook petite), ce qui risque de perdre de l'information et il ne faut pas choisir un grand nombre de caractéristiques au risque de disperser les propriétés. Nous avons effectués quelques tests, et nous avons retenu un codebook de taille M=16.

4 Caractérisation d'un mot clef

Comme nous l'avions mentionné, considérons le mot clef [musafir] que nous recherchons dans différents documents audio. On notera que nous caractérisons le mot [musafir], sans prêter grande attention à la déclinaison finale du mot.

Pour caractériser un mot il faut observer plusieurs réalisations du mot et rechercher les corrélations existantes entre sa réalisation et les différentes réalisations acoustiques représentées dans le dictionnaire

Voici une caractérisation possible du mot clef [musafir] :

CaractMC1 =

11 13 8 14

11 13 8 13

Cette caractérisation résume les caractéristiques acoustiques qui doivent être détectées en séquence pour prétendre à une détection de ce mot. Cette caractérisation est faite sur la base d'un dictionnaire à seize entrées.

5 Recherche d'un mot clef

5.1 Corpus utilisé

Pour évaluer notre approche nous utilisons l'un des corpus qui sont disponibles sur internet qui est celui de l'association internationale de phonétique [Web01].

Ce corpus comprend entre autre un ensemble fichiers audio qui contiennent des textes narratifs. Nous utiliserons ces fichiers comme flux audio dans lequel nous recherchons nos mots clefs. En particulier, nous développons d'abord le cas où nous recherchons le mot clef [musafir] qui est présent dans certains de ces fichiers.

5.3 La détection

Dans le processus de détection, nous considérons un flux d'entrée. Ce flux est traité par fenêtres de dix seconde. Chaque fenêtre est quantifiée. Sur chaque fenêtre, on effectue une recherche des caractéristiques du mot clef désigné. Dès que la première caractéristique est détectée, une variable qui représente le début du mot est initialisée par la position de cette première caractéristique. La recherche se poursuit jusqu'à la dernière caractéristique du mot.

Le module qui recherche un mot clef, est construit sur la base d'une fonction 'RechCar' dont le but est de détecter une caractéristique donnée.

La fonction « RechCar » a pour entrée :

- Le segment du signal sur lequel s'effectue la recherche,
- Le numéro de la caractéristique à rechercher
- La caractéristique à rechercher
- La position à partir de laquelle débute la recherche
- Le nombre de caractéristique qu'on est autorisé à 'sauter' avant de trouver cette caractéristique

Et parmi les éléments de sortie,

- La position à partir de laquelle va se poursuivre la recherche
- Le numéro de la prochaine caractéristique à rechercher
- Le nombre de caractéristique qu'on est autorisé à 'sauter' pour retrouver la prochaine caractéristique.

Il s'ensuit que si un mot clef est détecté la position du début et de fin du mot, en termes de ces caractéristiques est fournie en sortie du module.

6 Evaluation de performance

La détection de mot clefs génère une série d'alarmes à partir du signal de parole, qui indiquent la présence probable de mots clefs. Ces alarmes peuvent générer deux types d'erreurs. Les erreurs de type I concernent les occurrences de mots clefs qui ne sont pas détectés dans un flux de parole. Les erreurs de type II sont celles où le détecteur de mots clefs génère une fausse alarme en reconnaissant un mot clef dans le signal alors qu'il n'en existe pas. Selon la vocation finale du système, on cherchera à minimiser l'un des deux types.

Le taux de détection de mots clefs est défini comme étant le nombre d'alarmes correctes produites par le détecteur de mots clefs divisé par le nombre d'occurrence de mots clefs dans le signal original.

Le taux de fausse alarme est défini comme le nombre de fausses alarmes par heure de parole normalisé par le nombre de mots clefs.

6.1 Corpus de test

Nous considérons pour évaluer les performances de l'approche un ensemble de fichiers audio audio de la base API. Cet ensemble comprend sept (7) fichiers dont la concaténation produit un récit complet.

6. 2 Taux de détection

En plus des sept (7) fichiers, nous construisons des fichiers composés de la concaténation de deux (2), trois (3), quatre (4) ou cinq (5) fichiers, pour évaluer le comportement de l'algorithme en présence de plus d'une occurrence d'un mot clef. Nous recherchons dans les différents fichiers le mot [musafir]. Voici les résultats obtenus :

Nom du fichier	Nombre d'occurrence du mot clef	Nombre d'occurrence détecté
Narrative1.wav	0	1
Narrative2.wav	1	1
Narrative3.wav	1	1
Narrative4.wav	0	1
Narrative5.wav	1	1
Narrative6.wav	1	0
Narrative7.wav	0	0
Narrative8.wav	2	2
Narrative9.wav	3	4
Narrative10.wav	4	4

Table5.2 : taux de détection obtenus

Ainsi, nous pouvons calculer le taux de détection

6. 3 Taux d'erreur

Pour calculer le taux d'erreur nous construisons un fichier audio d'une durée d'une heure (1heure), et nous y recherchons le mot clef [musafir]. Sur un enregistrement d'une heure, qui contient plusieurs occurrence du mot clef recherché.

7 La reconnaissance

Comme nous l'avions mentionné, l'algorithme retourne les frontières du mot clef, s'il est détecté. Ce qui permet d'accepter ou rejeter cette alarme, par une reconnaissance du mot clef en utilisant un modèle HMM ; ceci dans les applications à caractère vital. Dans notre cas, le routage des appels, on peut se limiter à la première détection.

Les choix qui ont été faits pour le système de reconnaissance sont les suivants :

- Unités acoustiques : phonèmes.
- Topologie des modèles : modèle de Bakis (modèle gauche droite).
- Probabilité d'émission modélisée par une combinaison de gaussiennes.

Tous les modèles ont la même topologie, et les probabilités d'émission de tous les états sont représentées par un nombre identique de gaussiennes (10). Pour les paramètres de modélisation : Le nombre des états dépend du nombre des phonèmes du mot prononcé.

8 Conclusion

Nous avons présenté, dans ce chapitre, les différentes étapes de l'évaluation de l'approche proposée dans le chapitre précédent.

Afin de faire l'évaluation de l'approche, nous avons utilisé comme corpus de test, les fichiers audio disponibles dans la base API. Nous avons utilisés les sept fichiers de la base et nous avons construits d'autres fichiers par concaténation.

Nous avons testé l'apport de notre approche, par la recherche dans les différents fichiers du mot clef [musafir]. Nous aussi l'avons évalué dans le cas où le fichier comporte plusieurs occurrences du même mot. Nous avons obtenu un taux de reconnaissance acceptable.

Conclusion et perspectives

La détection de mots clefs est un domaine qui a fait l'objet de plusieurs travaux essentiellement depuis l'apparition des applications interactives. Plusieurs méthodes ont été proposées et évaluées selon deux approches principales : les modèles poubelles et les mesures de confiance.

Dans ce mémoire, nous avons proposé une nouvelle approche pour la détection de mots clefs qui se voulait être une alternative aux modèles poubelles et à la détection du début et de la fin du mot clef, dans la mesure où nous ne considérons que les mots clefs sans avoir à modéliser les mots hors vocabulaire. L'approche que nous avons présentée est inspirée de la démarche perceptuelle, où dans un flux de paroles nous détectons un mot clef que nous recherchons à travers ses caractéristiques.

Nous avons testé notre nouvelle approche par la recherche de mots clefs dans un ensemble de fichier audio de la base API.

Dans une première étape, nous avons recherché un mot clef dans un fichier. Puis dans une deuxième tape, nous avons construit de nouveaux fichiers par concaténation, dans le but de tester le comportement de l'algorithme en présence de plusieurs occurrences du même mot clef dans un seul fichier. Les résultats obtenus dans les deux étapes de l'évaluation étaient satisfaisants.

Notre travail s'est déroulé sur un corpus relativement petit. Il reste à tester l'efficacité de l'approche sur un corpus plus grand et avec un nombre plus important de mots clefs.

Nous envisageons l'exploitation de notre travail dans des applications interactives telles que la consultation orale des bases de données comme par exemple la consultation des comptes CCP.

Une autre perspective à notre travail, est l'indexation des documents audio ou la recherche sur le web en utilisant la parole.

Une dernière perspective est l'utilisation de notre approche dans le domaine de la reconnaissance automatique de la parole. En effet, les techniques de détection de mots clefs, peuvent être appliquées pour la création de résumé automatique des fichiers audio.

Références bibliographiques

[Abdillahi et al. 06] N. Abdillahi, P. Nocéra, et J.-F. Bonastre. Towards Automatic Transcription of Somali Language. Dans les actes de Language Resource and Evaluation Conference (LREC).

[Baker 75b] J.K Baker. Stochastic modeling as a means of automatic speech recognition. PHD thesis, Carnegie-Mellon University,1975.

[Baker 05] Jon Baker, Martin Cooke & Daniel P.W.Ellis. Decoding speech in the presence of others sources. Speech communication, Vol.45, n°1 pages 20-25, january2005.

[Bar-Hillel 58] Y. Bar-Hillel. The mechanization of litterature searching. Mechanization of Thought Processes 10, 4–8.

[Baum 72] L.E.Baum. An inequality and associated maximization technique in statistical estimation for probabilistic functions of a markov process. Inqualities, pages 1-8,1972.

[Bayya 00] A. Bayya. Rejection in speech recognition system with limited training. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 305_308, 2000.

[Becchetti 99] Claudio Becchetti, Lucio P. Ricotti. Speech Recognition: Theory and C++ Implementation .John Wiley & Sons Ltd; **Édition :** Har/Cdr

[Bellman 57] R.Bellman. Dynamic programming. Princeton University Press,1957.

[Bernardis et Bourlard 98] G. Bernardis et H. Bourlard. Confidence measures in hybrid HMM/ANN speech recognition. In Proceedings of the First Workshop on Text, Speech, Dialogue, pages 159_164, 1998.

[Bezie et Lockwood 93] O. Bezie et P. Lockwood. Beamsearch and traceback in the frame synchronous two level algorithm. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 511_514, 1993.

[Boite et al 93] J. M. Boite, H Bourlard, B. D'hoore, et M. Haesen. A new approach towards keyword spotting. In Proceedings of the European Conference On Speech Communication and Technology, pages 1273_1276, 1993.

[Bourlard et al 94] H. Bourlard, B. D'hoore, et J. M. Boite. Optimizing recognition and rejection performance in wordspotting system. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 373_376, 1994.

[Brown et al. 01] E. W. Brown, S. Srinivasan, A. Coden, D. Ponceleon, J. W. Cooper, et A. Amir. Toward Speech as a Knowledge Resource. IBM Systems Journal 40, 985–1001.

[Byrne et al. 04] W. Byrne, D. Doermann, M. Franz, S. Gutsman, J. Hajic, M. Picheny, J. Psutka, B. Ramabhadran, D. Soergel, T. Ward, et W. Zhu. Automatic recognition of

spontaneous speech for access to multilingual oral history archives. IEEE Transactions on Speech and Audio Processing 12(4), 420–435.

[Calliope 89] Calliope : « la parole et son traitement automatique». Edition Masson 1989.

[Charlet et al 01] D. Charlet, G. Mercier, et D. Jouvet. On combining confidence measures for improved rejection of incorrect data. In Proceedings of the European Conference On Speech Communication and Technology, 2001.

[Church et Gale 95] K. Church et W.Gale. Poisson mixtures. Natural Language Engineering 1(2), 163–190.

[Clary et Hansen 92] G.J. Clary et J.H.L. Hansen. A novel speech recognizer for keyword spotting. In Proceedings of the International Conference on Spoken Language Processing, pages 13_16, 1992.

[Cole et al 95] R. Cole, L. Hirschman, L. Atlas, M. Beckman, A. Bierman, M. Bush, M. Clements, J. Cohen, O. Garcia, B. Hanson, H. Hermansky, S. Levinson, K. Mckeown, N. Morgan, D. G. Novick, M. Ostendorf, S. Oviatt, C. Weinstein, S. Zahorian, et V. Zue. The challenge of spoken language system : research directions for the nineties. IEEE Transactions on Speech and Audio Processing, 1 :1_21, 1995.

[Davis 80] S.B.Davis, P.Mernnelstein. Comparison of parametric representation for monosyllabic word recognition in continuously spoken sentences, IEEE Trans.Acous, Speech, Signal Processing, vol.28,n°4,pp357-366,1980.

[Demange 07] S.Demange. Contributions à la reconnaissance automatique de la parole avec données manquantes. Thèse de Doctorat. Université Henri Poincaré Nancy 1. Novembre 2007.

[Dempster 77] A.Dempster, N.Laind & D.Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society, pages 1-38,1977.

[Dominich 01] S.Dominich. Mathematical Foundations of Information Retrieval. Kluwer Academic Publishers, Dordrecht, Boston, London, 2001.

[Lleida et al 93] E. Lleida, J. B. Marino, J. Salavedra, A. Bonafonte, E. Monte, et A. Martinez. Out-of-vocabulary word modelling and rejection for keyword spotting. In Proceedings of the European Conference On Speech Communication and Technology, pages 1265_1268, 1993.

[Favre 03] B. Favre. Indexation Multimédia : Caractérisation du Déséquilibre entre les Modalités Texte et Parole. Mémoire de Master, Université d'Avignon.

[Gales 93b] M.Gales& S.J.Young. Segmented HMMs for speech recognition. In EUROSPEECH, pages 837-840, Berlin 1993.

[Garofolo et al. 99] J. Garofolo, C. Auzanne, et E. Voorhees. The TREC spoken document retrieval track : A success story. Dans les actes de Text REtrieval Conference (TREC), Volume 8, 16–19.

[Gelin et Wellekens 96] P. Gelin et C. Wellekens. Keyword spotting for video soundtrack indexing. In Proceedings of the International Conference on Spoken Language Processing, pages 299_302, 1996.

[Géry 02] M.GÉRY. Indexation et interrogation de chemins de lecture en contexte pour la Recherche d'Information Structurée sur le Web .Thèse de Doctorat. Université Joseph Fourier - Grenoble I. Octobre 2002

[Godin et Lockwood 89] C. Godin et P. Lockwood. DTW schemes for continuous speech recognition : a uni_ed view. Computer Speech and Language, 3 :169_198, 1989.

[Godfrey et al. 92] J. Godfrey, E. Holliman, et J. McDaniel. SWITCHBOARD : Telephone speech corpus for research and development. Dans les actes de International Conference on Acoustics, Speech and Signal Processing (ICASSP), 517–520.

[Gorin et al., 1997] A. L. Gorin, G. Riccardi, et J. H. Wright. How May I Help You ? Speech Communication, 23 :113_127, 1997.

[Gupta et Soong 98] S. Gupta et F. K. Soong. Improved utterance rejection using length dependent thresholds. In Proceedings of the International Conference on Spoken Language Processing, pages 795_798, 1998.

[Guy 1993] C.Guy, H.Chafiaa & R.Christian. Signaux déterministes et systèmes linéaires continus et stationnaires à temps discret. Département Mathématique et Systèmes de Communication, Edition Janvier 1993.

[Hernandez-Abrego et Marino 00] G. Hernandez-Abrego et B. Marino. Contextual confidence measures for continuous speech recognition. In Proceedings of International Conference on Acoustics, Speech, and Signal Processing, pages 1803_1806, 2000.

[Higgins et Wohlford 85] A. L. Higgins et R. E. Wohlford. Keyword recognition using template concatenation. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 1233_1236, 1985.

[James et Young 94] D. A. James et S. J. Young. A fast lattice-based approach to vocabulary independent words potting. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 377_380, 1994.

[Jones et al 95] G. J. F. Jones, J. T. Foote, K. Sparck Jones, et S. J. Young. Video mail retrieval : the effect of word spotting accuracy on precision. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 309_312, 1995.

[Kamppari et Hazen 00] S. O. Kamppari et T. J. Hazen. Word and phone level acoustic confidence scoring. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 1799_1802, 2000.

[Lavrenko 02] V. Lavrenko. A Generative Theory of Relevance. Thèse de Doctorat, University of Massachusetts. (Le Meur et al., 2004) C. Le Meur, S. Galliano, et E. Geoffrois, 2004. Guide d'Annotation en Entités Nommées ESTER.

[Levinson 86] S.Levinson. Continiously variable duration hidden Markov models for automatic speech recognition. Computer Speech and Languages, vol. 1, pages 29-45,1986.

[Mathan 91] L. Mathan. Contributions à la reconnaissance de la parole pour des serveurs vocaux interactifs. PhD thesis, Ecole Nationale Supérieure des Télécommunications, 1991.

[Mauuary 94] L. Mauuary. Amélioration des performances des serveurs vocaux interactifs. Thèse de Doctorat, Université de Rennes I, 1994.

[Mazor et Feng 93] B. Mazor et M. W. Feng. Improved a posteriori processing for keyword spotting. In Proceedings of the European Conference On Speech Communication and Technology, pages 2231_2234, 1993.

[Moreau et al 00] N. Moreau, D. Charlet, et D. Jouvet. Confidence measure and incremental adaptation for the rejection of incorrect data. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, 2000.

[Moreau et Jouvet 00] N. Moreau et D. Jouvet. Détermination d'une mesure de confiance pour le rejet des entrées incorrectes. In XXIIIèmes Journées d'Etudes sur la Parole, pages 173_176, 2000.

[Morgan et al 91] D. Morgan, C. Scofield, et J. Adcock. Multiple neural network topologies applied to keyword spotting. In Proceedings of International Conference on Acoustics, Speech, and Signal Processing, pages 313_316, 1991.

[Myers et al 80] C. S. Myers, L. R. Rabiner, et A. E. Rosenberg. An investigation of the use of dynamic time warping for word spotting and connected speech recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 173_177, 1980.

[Myers et Rabiner 81]. C. S. Myers et L. R. Rabiner. A dynamic time-warping algorithm for connected word recognition. IEEE Transations on Acoustics, Speech, and Signal Processing, 29 :351_363, 1981

[Pols 97] L. C. W. Pols. Flexible human speech recognition. In Proceedings of the Workshop on Automatic Speech Recognition and Understanding, pages 273_283, 1997

[Ponte et Croft 98] J. M. Ponte etW. B. Croft. A Language Modeling Approach to Information Retrieval. Dans les actes de ACM Special Interest Group on Information Retrieval (SIGIR), 275–281.

[Rabiner 93] L.Rabiner et H.hwang. Fundamentals of speech recognition. Prentice Hall,1993.

[Rabiner et Schmidt 80] L. R. Rabiner et C. E. Schmidt. Application of dynamic time-warping to connected-digit recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing, 28 :337_388, 1980

[Riccardi et al 96] G. Riccardi, R. Pieraccini, et E. Bocchieri. Stochasic automata for language modeling. Computer Speech and Language, 10 :265_293, 1996.

[Rivlin et al 96] Z. Rivlin, M. Cohen, V. Abrash, et T. Chung. A phone-dependent confidence measure for utterance rejection. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 515_518, 1996.

[Robertson et Spärck-Jones 88] S. Robertson et K. Spärck-Jones. Relevance Weighting of Search Terms. Taylor Graham Series In Foundations Of Information Science 27, 143–160.

[Robertson et Walker 94] S. Robertson et S. Walker. Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. Dans les actes de ACM Special Interest Group on Information Retrieval (SIGIR), 232–241. Springer-Verlag New York, Inc. New York, NY, USA.

[Rohlicek et al 89] J. R. Rohlicek, W. Russell, S. Roukos, et H. Gish. Continuous hidden markov modeling for speaker independent word spotting. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 627_630, 1989

[Rose 95] R. C. Rose. Keyword detection in conversational speech utterances using hidden markov model based continuous speech recognition. Computer, Speech and Language, 9 :309_333, 1995.

[Rose et al 98] R. C. Rose, H. Yao, G. Ricardi, et J. Wright. Integration of utterance verification with statistical language modeling and spoken language understanding. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 237_240, 1998.

[Russel 85] M.Russel & P.Moore. Explicit modeling of state occupacy in hidden Markov models for automatic speech recognition. In Proc.ICASSP, pages 2376-2379, Tampa,1985.

[Russel 93] M.Russel. A segmented HMM for speech pattern matching. In ICASSP, pages 499-502, Mineapolis,1993.

[Sanderson et Shou 02] M. Sanderson et X. M. Shou . Speech and Hand Transcribed Retrieval. Dans les actes de ACM Special Interest Group on Information Retrieval (SIGIR). Springer.

[Silaghi et Bourlard 00] M.C. Silaghi et H. Bourlard. A new keyword spotting approach based on iterative dynamic programming. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 1966_1969, 2000.

[Spärck-Jones et al. 00] K. Spärck-Jones, S. Walker, et S. Robertson. A probabilistic model of information retrieval : development and comparative experiments. Information Processing and Management : an International Journal 36(6), 779–808.

[Sukkar et Wilpon 93] R. A. Sukkar et J. G. Wilpon. A two pass classi_er for utterance rejection in keyword spotting. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 451_454, 1993.

[Rocchio 71] J.J.Rocchio . Relevance feedback in information retrieval. The SMART retrieval system : experiments in automatic document processing, éd. par Gerald Salton, pp. 313–323. – Prentice Hall, 1971

[Salton et al. 75] G. Salton, A. Wong, et C. S. Yang. A Vector Space Model for Automatic Indexing. Communications of the ACM 11(18), 613–620.

[Salton et al. 83] G. Salton, E. Fox, et H. Wu. Extended Boolean Information Retrieval. Communications of the ACM 26(11), 1022–1036.

[Salton et al.90] G.Salton et C.Buckley . Improving retrieval performance by relevance feedback. Journal of the American Society for Information Science, vol. 41, Juin 1990, pp. 288–297.

[Tadj 95] C. Tadj. Méthodes connexionnistes de quantification vectorielle à apprentissage compétitif. Application à la détection de mots clé. Thèse de Doctorat, Ecole Nationale Supérieure des Télécommunications, 1995

[Thong et al. 00] J.-M. V. Thong, D. Goddeau, A. Litvinova, B. Logan, P. Moreno, et M. Swain. SpeechBot a Speech Recognition Based Audio Indexing System. Dans les actes de Recherche d'Information Assistée par Ordinateur (RIAO).

[Vergyri 00] D. Vergyri. Use of word level side information to improve speech recognition. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 1823_1826, 2000.

[Web01] International Phonetic Alphabet, http://www.arts.gla.ac.uk/IPA/

[Wechsler et al. 98] M.Wechsler, E. Munteanu, et P. Schauble. New Techniques for Open Vocabulary Spoken Document Retrieval. Dans les actes de ACM Special Interest Group on Information Retrieval (SIGIR), 20–27. ACM Press.

[Weintraub et al 97] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, et A. Stolcke. Neural-network based measures of con_dence for word recognition. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 887_890, 1997.

[Williams et Renals 97], G. Williams et S. Renals. Confidence measures for hybrid HMM/ANN speech recognition. In Proceedings of the European Conference On Speech Communication and Technology, pages 1955_1958, 1997.

[Wilpon et al 90] J. G. Wilpon, L. R. Rabiner, C. Lee, et E. R. Glodman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. IEEE Transactions on Acoustics, Speech and Signal Processing, 38 :1870_1878, 1990.

[Wong et al. 85] S. K. M. Wong, W. Ziarko, et P. C. N. Wong. Generalized Vector Spaces Model in Information Retrieval. Dans les actes de ACM Special Interest Group on Information Retrieval (SIGIR), 18–25.

[Yapanel 97] U. Yapanel. Garbage modeling techniques for a turkish keyword spotting system. PhD thesis, Istanbul Teknik university, 1997.

[Yousfi 01] A.Yousfi . Introduction de l'énergie et de la vitesse d'élocution dans un modèle de reconnaissance automatique de la parole. Université Mohamed Premier Oujda. Juin 2001.

[Yu et Seide 04] P. Yu et F. Seide. A Hybrid Word/Phoneme-Based Approach for Improved Vocabulary-Independent Search in Spontaneous Speech. Dans les actes de International Conference on Spoken Language Processing (ICSLP).

[Zeppenfeld et Waibel 92] T. Zeppenfeld et A. Waibel. A hybrid neural network, dynamic programming word spotter. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing, pages 77_80, 1992.

[Zhang et Rudnicky 01] R. Zhang et A. I. Rudnicky. Word level con_dence annotation using combinations of features. In Proceedings of the European Conference On Speech communication and Technology, 2001.