République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Cheikh Larbi Tebessi - Tébessa
Faculté des Sciences et des Technologies
Département Informatique

N° D'ORDRE : ……
SERIE : ……..…

MEMOIRE

Présenté en vue de l'obtention du titre de
Magistère en Informatique
Option: Systèmes d'informations et Connaissances(SIC)

DETECTION DE MOTS CLEFS DANS UN FLUX DE PAROLE

Présenté par : Mr BENDIB Issam

Dirigé par : MC. BAHI Halima

SOUTENU LE : 10/06/2009

Devant le Jury composé de:

Président:	Mme Dr Hayet Merouani	**Maître de conférences**
Rapporteur:	Mme Dr Bahi Halima	**Maître de conférences**
Examinateurs :		
	Mr Dr Meslati Djamel	**Maître de conférences**
	Mme Dr Labiba Souici	**Maître de conférences**

A mes parents

A mon épouse

A mes enfants : Ala errahmene et Med Taha

A toute la famille

Remerciements

Tout d'abord je voudrais remercier Mme BAHI. H, mon directeur de thèse, pour m'avoir accueilli au sein du laboratoire LABGED pendant ces deux années d'études et pour m'avoir encadré pendant cette thèse.

Je remercie tous les membres de mon jury, c'est-à-dire Dr BAHI Halima pour avoir acceptée d'être rapporteurs de ma thèse, Dr MESLATI Djamel et Dr SOUICI Labiba pour en avoir été examinateurs et Dr MEROUANI Hayet pour sa présidence.

Toute ma gratitude à toutes les personnes ayant relu, corrigé et commenté mon manuscrit et ayant ainsi participé à son amélioration.

Je remercie mes parents et mon épouse pour m'avoir toujours poussé dans mes études. Je remercie aussi ma grand-mère, mes frères Nabil, Atef et Salah et mes sœurs.

Je tiens à saluer aussi Mon ami Ouahid et toutes les personnes du service informatique de la wilaya de Tébessa.

Sommaire

Table des Figures

Introduction générale

L'apparition de la technologie multimédia et de systèmes d'informations intégrant d'autres médias que le texte, tels que l'image, la vidéo, le son, a eu un impact certain sur la recherche d'informations, donnant naissance à une nouvelle génération de systèmes de recherche d'informations multimédias. C'est ainsi que sont apparus divers systèmes de manipulation d'images et de documents multimédias complexes. Ces systèmes s'apparentent tant aux bases de données qu'aux systèmes de recherche d'informations, chacun ayant ses spécificités, ses objectifs et ses techniques de représentation particuliers.

La recherche d'information pour les documents texte est performante et aboutie à des résultats encourageants. Toutefois, l'avènement des informations multimédia envahit les banques de données ce qui met la recherche d'information en difficulté, voire même incapable d'accéder aux contenus de ces informations. Cependant le domaine de détection automatique de la parole occupe un grand intérêt dans ces dernières années.

La reconnaissance automatique de la parole est un domaine de recherche très vaste, et surtout avec l'avancement terrible des technologies multimédia. Entre autre, l'avancement dans les technologies d'acquisition et de stockage des documents audio et leurs exploitation via le réseau mondial, impose une manipulation automatique avec des systèmes de reconnaissances de la parole.

Le signal de parole est caractérisé par plusieurs paramètres intrinsèques qui rendent sont interprétation délicate. Cependant, le signal est très varié ; soit d'un locuteur à un autre, ou même pour un locuteur lui-même. Les différences d'age, de sexe, d'accent entre locuteurs, d'origine sociale rendent délicates l'extraction d'informations pertinentes caractérisant le signal, cette extraction doit être indépendante du locuteur. En plus le signal acoustique dans des milieux ambiants comme les bruits extérieurs, bruits de bouches ou respirations et aussi la qualité d'enregistrement génèrent également des difficultés pour les systèmes de reconnaissance de la parole. En plus, le type de signal de parole à traiter : mots isolés, parole continues ou paroles spontanées augmente le degré de complexité de la tâche de reconnaissance.

Position du problème

L'application des systèmes de reconnaissance conventionnels pour la parole spontanée présente quelques problèmes car il faut considérer un grand vocabulaire et le langage doit être modélisé par une grammaire complexe qui considère tous les évènements possibles dans la parole spontanée, comme les phrases tronquées, les phrases grammaticalement incorrectes, toux, début incorrect, etc. Les systèmes de détection de mots clés ont fournit une solution aux processus de la parole spontanée. En effet le but des systèmes de reconnaissance conventionnels 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. Dans

ces systèmes, les segments sémantiques significatifs sont extraits tandis que le reste est ignoré, le contenu sémantique peut être donc capté sans une reconnaissance détaillée de tous les mots de la phrase. Il suffit donc que les mots clés précédemment définis dans le vocabulaire soient détectés s'ils sont prononcés dans la phrase.

Les systèmes de détection des mots clés sont structurés généralement en deux parties : mots clés et des séquences de paroles qui contient les mots non clés (hors vocabulaire) et même de bruits. La réalisation d'un tel système donc implique de modéliser les mots clés (vocabulaire) affin d'accroître la détection et aussi modéliser les mots non clés pour réduire les fausses acceptations. La performance du système est validée par rapport au compromis entre ces deux tâches.

Contribution

Dans ce contexte, notre contribution est d'accentuer la recherche par l'intégration des systèmes capables de chercher l'information dans les fichiers audio en se basant sur la technique détection des mots clés (KWS) dans un flux continu de parole. Cette technique se limite à la recherche et la détection des segments bien définis dans le signal qui nous offre une amélioration importante dans les calculs car les systèmes classiques de reconnaissance de la parole consomment beaucoup de calculs.

Entre outre, la majorité des recherches effectuées dans le domaine de la détection de mots clés se limitent à un vocabulaire limité. De plus, elles traitent les flux de paroles des mots isolés, et n'oublions pas que la majorité de ces systèmes sont très sensibles aux bruits et des effets extérieurs.

Organisation du mémoire

Le premier chapitre contient une présentation des systèmes de recherche d'information, où on y introduit la recherche d'information audio. On deuxième chapitre, on présente la reconnaissance automatique de la parole et on décrit les différents composants d'un système RAP. On y trouve on particulier, une description de l'analyse du signal et des modèles de Markov caché qui vont servir d'outil de modélisation pour les mots clés et les mots non clés dans une application de détection.

Le troisième chapitre présente l'état de l'art en détection de mots clés en tant que discipline annexe de la reconnaissance automatique de la parole. En particulier, on décrit les principales approches dans la modélisation des mots poubelles. En suite, en quatrième chapitre on présente les éléments conceptuels de l'approche qui présente les deux variantes pour la modélisation des modèles poubelles. Et dans le dernier chapitre, on présente un protocole d'évaluation de l'approche

Chapitre 1

Recherche d'information

1.1. Introduction

Les Systèmes de Recherche d'Information documentaire (SRI) sont nés de la nécessité d'automatiser la gestion des informations documentaires. Les études dans ce domaine sont cruciales, tant sur le plan théorique qu'appliqué compte tenu de la quantité d'informations gérées à l'heure actuelle par les entreprises, les administrations ou le grand public et des enjeux sociaux-économiques liés à la rapidité d'accès aux informations et à la pertinence des informations mises à disposition des utilisateurs.

La plupart des Systèmes de Recherche d'Informations (SRI) développés manipulent des documents de nature textuelle. Ce qui a fourni plusieurs modèles théoriques et pratiques de composantes ou de systèmes, parmi lesquels se situe le modèle logique général et les modèles d'indexation sémantiques. Entre outre, la possibilité d'avoir de l'information au format numérique a sans doute beaucoup d'avantages : possibilité de transfert rapide, reproduction illimitée, facilité de stockage. Comme pratiquement toute l'information disponible aujourd'hui est sous format numérique.

L'apparition de la technologie multimédia et de systèmes d'informations intégrant d'autres médias que le texte, tels que l'image, la vidéo, le son, a eu un impact certain sur la recherche d'informations, donnant naissance à une nouvelle génération de systèmes de recherche d'informations multimédias. C'est ainsi que sont apparus divers systèmes de manipulation d'images et de documents multimédias complexes. Ces systèmes s'apparentent tant aux bases de données qu'aux systèmes de recherche d'informations, chacun ayant ses spécificités, ses objectifs et ses techniques de représentation particuliers.

Toutefois, plusieurs des problèmes conséquents aux données multimédias demeurent posés. En effet, les mesures de similarité entre deux données, les représentations de la sémantique de ces données, et leur interrogation définissent les axes majeurs de la recherche actuelle dans ce domaine.

1.2. Architecture Générale

Un système de recherche d'informations manipule un ensemble de documents en vue de satisfaire les besoins d'informations des utilisateurs. Il présente à cet effet des fonctions de manipulation permettant de représenter et de stocker les documents, des moyens d'expression des besoins et des requêtes et enfin une fonction d'évaluation de la correspondance entre une requête et les documents de la base pour déterminer ceux qui sont jugés pertinents (Figure 1.1)

Le schéma général d'un système de recherche d'informations est donc constitué par les trois composantes : d'indexation, d'interrogation et de correspondance.

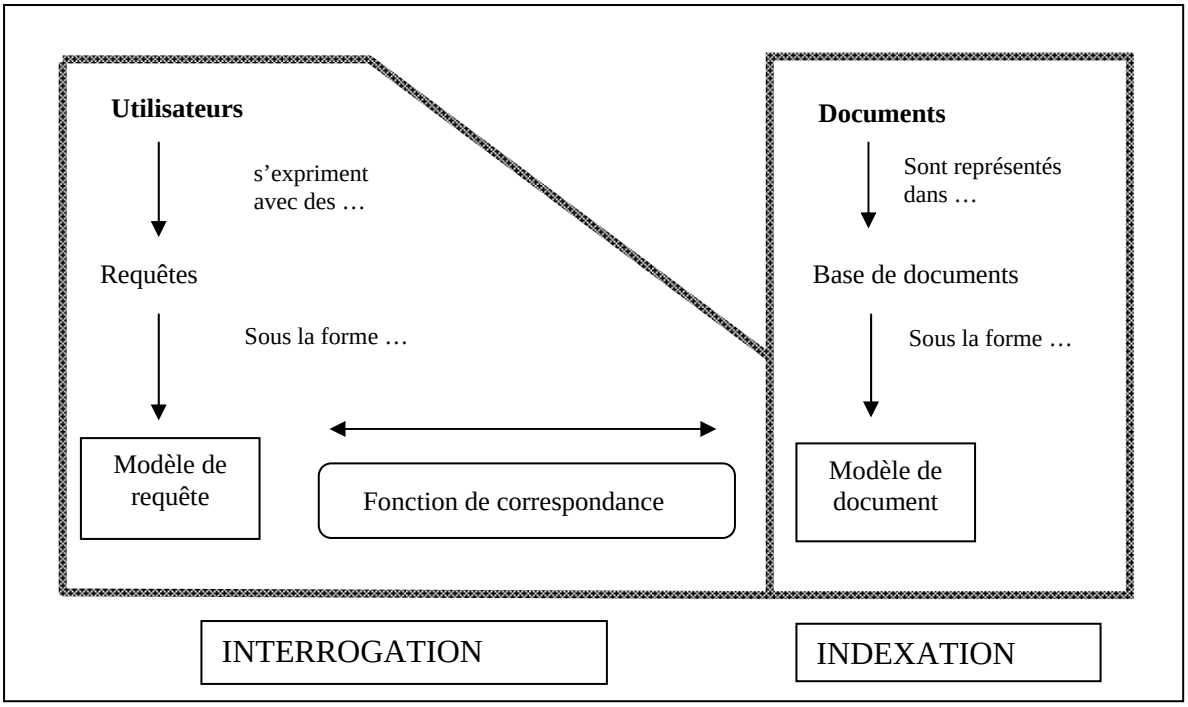


Figure 1.1 : Problématique et principales fonctions de la recherche d'information.

1.2.1. L'indexation :

Un processus d'indexation est mis en œuvre afin d'extraire préalablement une représentation homogène du contenu sémantique, sous forme de termes d'indexation qui sont des éléments d'un langage d'indexation.

Dans les systèmes classiques de la recherche d'informations, l'indexation est organisée en trois étapes : extraction, sélection, pondération, dont l'objectif final est de définir pour chaque document ses termes d'indexation.

En fonction de la nature des documents et du niveau d'appréhension de la sémantique, différentes méthodes d'indexation ont été élaborées et diverses classifications ont été établies. Un modèle d'indexation est nécessaire pour fixer la structure du document d'indexation et le langage d'indexation.

1.2.2. L'interrogation :

L'interrogation a pour objectif de traiter les requêtes d'un utilisateur en offrant un formalisme pour les exprimer de manière à satisfaire les besoins des utilisateurs. Les techniques de représentation des requêtes ont évolué parallèlement à l'évolution de la notion de document, passant des simples ensembles de spécification d'attributs, à la manière des requêtes en base de données, à l'intégration de moyens plus relatifs aux contenus sémantiques des éléments de la base.

Une fois la base d'indexation construite, l'interrogation est la fonction principale d'un système de recherche d'information. Elle offre à l'utilisateur les moyens d'exprimer son besoin selon une «syntaxe» définissant un modèle de requête. L'étape cruciale intervient alors : la mise en

correspondance entre la requête, d'une part, et l'indexation d'un document d'autre part. Le résultat de la mise en correspondance est un ensemble de documents jugés pertinents.

Une requête contient des critères décrivant les caractéristiques souhaitées des documents recherchés. Le modèle de requête n'est pas indépendant du modèle de document retenu, la fonction de mise en correspondance a pour rôle d'établir une base de comparaison.

Un document est dit pertinent si l'utilisateur juge qu'il correspond à son besoin ; on parle de pertinence-utilisateur. Les documents retrouvés par un SRI en réponse à une requête sont jugés pertinents par le système à travers la fonction de correspondance ; on parle de pertinence système. Des décalages sont inévitables entre la fonction de pertinence d'un SRI et celle d'un utilisateur. Au moins deux raisons peuvent être invoquées : l'imperfection des indexations et la difficulté de construire un requête reflétant exactement le besoin réel. Une possibilité d'améliorer les performances d'une recherche est alors de tenir compte des jugements effectués par l'utilisateur, sur les documents proposés à une étape de la recherche, pour reformuler (automatiquement ou interactivement) la requête. Cela se fait en augmentant l'impact, ou même en faisant apparaître, des termes présents dans les documents retrouvés et jugés pertinents pour la requête et inversement, en diminuant les poids des termes présents dans les documents jugés non pertinents. La recherche peut alors être réitérée avec cette nouvelle requête. Le processus interactif résultant est connu sous le nom de bouclage de pertinence.

1.2.3. La fonction de correspondance :

Un processus d'interrogation, permettant à l'utilisateur de formuler une requête et ainsi d'interroger le corpus. La requête et le corpus sont représentés respectivement dans un modèle de requêtes et un modèle de documents (un langage d'indexation). Un modèle de correspondance compare la requête aux documents : les documents répondant à la requête sont par conséquents donnés en réponse. Il faut, à ce niveau, établir une comparaison sémantique (et non une égalité) entre les concepts figurant dans un document et ceux figurant dans la requête.

La comparaison entre la requête et document aboutit rarement à des équivalences strictes, mais plutôt à des équivalences partielles : le document correspond à une partie seulement de la requête. Les documents peuvent ainsi être ordonnés selon une relation d'ordre permettant le classement des documents du plus pertinent au moins pertinent. On appelle pertinence-système la pertinence que le système attribue à chaque document pour une requête donnée.

1.3. Objectifs D'un Système De Recherche D'informations

Satisfaire les besoins des utilisateurs constitue la finalité des systèmes de recherche d'information que Salton [Salton, 1983] présente sous cinq critères fondamentaux :

- L'effort, intellectuel ou physique, nécessaire à l'utilisateur pour formuler les requêtes, conduire sa recherche, visualiser les documents donnés en réponse ;
- Le temps entre l'envoi d'une requête et la représentation de la réponse ;
- La présentation des résultats qui influence l'utilisateur dans sa motivation à consulter les documents retrouvés ;
- Le contenu du corpus, c'est à dire sa capacité à donner intrinsèquement de bonnes réponses à une requête ;
- Et surtout, la capacité du système à identifier uniquement les documents pertinents, et à éliminer les autres.

La plupart des systèmes insistent sur ce dernier point, qui constitue effectivement le point essentiel dans la qualité attendu d'un système. En effet, qu'un système fournisse de bonnes réponses à l'utilisateur est une condition incontournable pour son utilisation !

Au niveau opérationnel ce critère est testé en comparant les réponses données à un ensemble de requêtes par le système à celles souhaités par l'utilisateur. Ces tests mettent en évidence la dualité entre la pertinence système (ce que l'utilisateur souhaite retrouver), en mesurant la distance entre ces deux pertinences.

En pratique, ces tests sont effectués en utilisant les collections tests proposent en effets un corpus et un ensemble de requêtes résolues. Ainsi pour toute requête Q d'une collection test, l'ensemble P des documents pertinents, c'est à dire répondant à la requête Q selon le point de vue de l'utilisateur, et l'ensemble $\neg P$ formé des documents non pertinents pour Q, sont définis en extension et partitionnent le corpus C de la collection test.

La réponse d'un système de recherche d'informations S à la requête Q partionne le corpus C en deux sous-ensembles : l'ensemble R des documents retrouvés par le système de recherche d'informations S, c'est à dire répondant à la requête Q selon ce système, et l'ensemble $\neg R$ des documents non retrouvés. Ainsi on obtient figure 1.2 :

Le système de recherche d'informations est celui tel que R=P, pour toute requête Q de la collection test, mais un tel système n'existe pas. Aussi existe-t'il un certain nombre de mesures permettant de caractériser chaque système ?

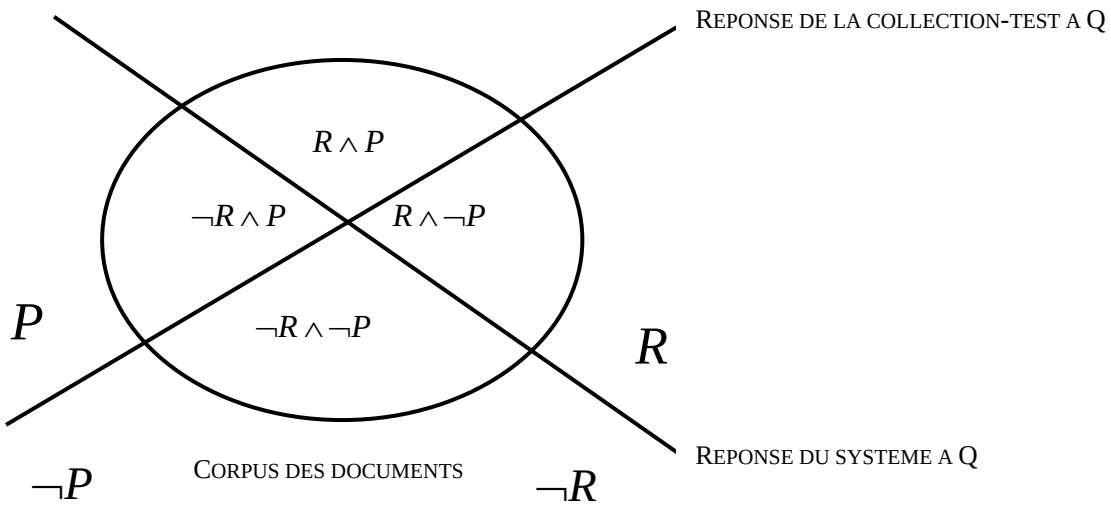


Figure 1.2 : Corpus Test.

1.4. Critères D'évaluation D'un Système De Recherche D'informations

1.4.1. Mesures qualitatives d'un système de recherche d'informations :

Pour mesurer les performances qualitatives des systèmes de recherche d'informations, on procède à la comparaison des ensemble P , $\neg P$, R , $\neg R$ sur l'ensemble des requêtes. Il existe à cet effet de nombreuses mesures, chacune mettant en évidence telle ou telle propriété du système testé. Les mesures de rappel (recall) et de précision :

$$RAPPEL = \frac{R \wedge P}{P} \qquad \textbf{(1.1)}$$

$$PRECISION = \frac{R \wedge P}{R} \qquad \textbf{(1.2)}$$

Le rappel mesure la capacité du système de recherche d'informations à trouver, pour une requête, tous les documents pertinents. Le rappel peut se donc se définir comme la probabilité pour un document d'être retrouvé, sachant qu'il est pertinent.

La précision mesure la capacité du système à trouver, pour une requête uniquement des documents pertinents. La précision est une mesure très intéressante pour mesurer la quantité des réponses d'un point de vue de l'utilisateur du système.

L'élimination (discrimination) et son complémentaire, l'hallucination (fallout) , permettent de mesurer la capacité du système à éliminer les documents non pertinents :

$$ELIMINATION = \frac{\neg R \wedge \neg P}{\neg P} \quad \textbf{(1.3)}$$

$$HALLUCINATION = \frac{R \wedge \neg P}{\neg P} \quad \textbf{(1.4)}$$

L'élimination mesure la capacité du système à éliminer tous les documents non pertinents. L'élimination peut donc se définir comme la probabilité pour un document d'être éliminé, sachant qui n'est pas pertinent.

D'autres mesures comme le bruit (le complémentaire de la précision), le silence (le complémentaire du rappel), la généralité (la proportion pour une requête donnée, de documents retrouvés par rapport à tout le corpus) soit ($\frac{R}{R \vee \neg R}$) existent de par leur complémentarité, ces mesures peuvent s'exprimer entre elles : par exemple de connaître trois des mesures parmi le rappel, la précision, l'hallucination, la généralité, pour calculer la quatrième.

Majoritairement, les systèmes opérationnels affichent leurs tableaux, de rappel et de précision. Pourtant certains travaux tels que préconisent l'utilisation plutôt du rappel et de l'élimination ou bien du rappel et de l'hallucination. Leur argumentation se base sur le fait que, contrairement à l'hallucination ou à l'élimination, la précision est trop sensible aux variations de la généralité de chaque requête. Dans ce contexte, le choix entre rappel/ précision et rappel/ hallucination relève d'abord et avant tout de la mise en évidence de propriétés particulières du système. Le rappel indique la proportion des documents pertinents retrouvés, et la précision est une mesure de l'efficacité avec laquelle les documents les documents pertinents sont retrouvés, En ce sens, le couple rappel/ précision donnent une mesure orientée utilisateur des performances du système. Par ailleurs, l'hallucination mesure l'efficacité de l'élimination des documents non-pertinents. Ainsi le couple rappel/ hallucination donne une mesure orientée système des performances du système.

1.4.2. Mise en cause de ces mesures qualitatives :

Les performances des systèmes de recherche de l'information sont ainsi systématiquement calculées et publiées depuis plusieurs décennies. Pourtant à fait un bilan pessimiste des performances des systèmes : 60% en rappel 40% en précision en moyenne. Dans ce contexte, ce résultat n'est pas un constat d'échec, mais le résultat d'un quiproquo sur l'interprétation des mesure d'expliquons en deux points :

1. La valeur accordée à ces mesures doit être relativisée.

2. Par l'utilisation de ces mesures, les performances des systèmes de recherche d'informations sont évaluées globalement, et ne permettent pas d'isoler les performances individuelles des différents processus. On constate ainsi que les systèmes mesurés sont en général munis de langages à mots clés et que bien souvent la comparaison entre systèmes se ramène donc à une comparaison de leur fonction de correspondance.

1.4.2.1. Valeur relative des mesures qualitatives

Le problème par toutes mesures qualitatives réside dans le décalage entre leur sens et leur mise en œuvre. A cet effet, deux utilisateurs différents formulant une même requête n'ont pas le même point de vue sur les réponses du même système. De plus si on confronte ces deux utilisateurs à deux systèmes différents, chaque utilisateur préfère l'un des deux systèmes, mais obligatoirement le même ! Pour cela, donner des mesures qualitatives d'un système n'a de sens que si nous indiquons le contexte dans lequel le système est utilisé. Sinon, il existe sûrement un utilisateur à qui on donne entière satisfaction, un autre qui est très déçu, et il est préférable d'afficher uniquement les mesures de l'utilisateur satisfait !

De façon plus générale, cet argument remet en cause les comparaisons faites actuellement avec les collections-tests. Ces collections-tests donnent un corpus, des requêtes et leurs réponses. On suppose que le plus on s'approche de ces réponses, meilleur est le système. Malheureusement, ceci ne prouve pas que le système soit bon, ceci prouve que le système fonctionne parfaitement pour la ou les personnes qui ont construit cette collection. La question peut alors être complètement déviée en choisissant la collection-test avec laquelle le système fonctionne le mieux.

1.4.2.1.1. L'hypothèse de l'existence de $P, \neg P$

Toutes les mesures prennent pour hypothèse l'existence d'une dichotomie du corpus en deux ensembles : les documents pertinents et les documents non-pertinents. L'existence de ces deux ensembles est une hypothèse peu réaliste pour des raisons pratiques et humaines :

- Qui peut définir de façon crédible l'ensemble P des documents pertinents d'une requête,... dans un corpus de plusieurs millions de documents ? Le rappel est donc une mesure intéressant, mais incalculable Concrètement. Il en est de même pour l'élimination et l'hallucination qui nécessitent la connaissance de l'ensemble $\neg P$ des documents non pertinents.
- Certes un utilisateur accordera à certains documents le fait d'être pertinent ou non pertinent. Mais tous les documents n'entrent pas dans cette catégorisation : certains seront, selon le point de vue de l'utilisateur, « relativement intéressants »,d'autres encore auront le statut « à voir »,... Par ailleurs, l'utilisateur peut déclarer des documents pertinents pour des raisons diverses : ce document est pertinent pour telle raison, cet autre document est pertinent pour telle autre raison. La sémantique de l'ensemble P et alors complexe.

1.4.2.1.2. Définition de R et $\neg R$ dans un système pondéré :

La pondération des documents retrouvés n'est pas introduite dans les mesures actuelles. Les calculs effectués montrent généralement l'évolution du rappel et de la précision en fonction du nombre de documents pris en compte. Mais afin de calculer le rappel et la précision, les documents

sont catégorisés dans les ensembles R et $\neg R$ sans tenir compte de la pondération accordée à chacun d'entre eux.

Ainsi ces mesures n'introduisent pas de statuts intermédiaires aux documents, et tout le "plus" de certains systèmes (faire découvrir de nouveaux documents, permettre des questions ouvertes, ...) ne peut actuellement être valorisé.

1.4.2.2. Problématique de l'évaluation :

Comparer statistiquement des systèmes de recherche d'informations élude complètement leur apport individuel : A taux de précision/ rappel égal, deux systèmes peuvent se complémenter parfaitement pour satisfaire un utilisateur, en apportant chacun des réponses complémentaires. L'union de ces deux systèmes peut être donner de meilleurs résultats qu'un système plus complexe.

Pour cela, un des problèmes essentiels de l'évaluation vient du fait que sa simulation de la satisfaction de l'utilisateur en deux ensembles (P et $\neg P$) et la visualisation binaire du système (R et $\neg R$) modélisent trop schématiquement la réalité de la recherche de l'information. Une première voie de meilleure modélisation serait sans doute d'introduire une plus grande discrétisation de la satisfaction de l'utilisateur et de la réponse du système, et d'introduire cette discrétisation dans les mesures actuelles.

Une seconde voie consisterait d'une part à introduire une dénotation de la requête et des jugements de la pertinence de l'utilisateur, afin de prendre en compte une sémantique plus fine de ses besoins. D'autre part, le système doit alors restituer à l'utilisateur le même niveau de justification de ses réponses : la réponse ne doit plus être une liste éventuellement pondérée de documents, mais une ou plusieurs listes de documents, chacune justifiée sémantiquement dans ses liens avec la requête. Utilisateur et système peuvent alors entrer dans un dialogue direct et explicite : quiproquos introduits par l'opacité des systèmes classiques peuvent être évités.

Cette seconde voie nécessite cependant une remise en question des systèmes et du cloisonnement de leurs processus : notamment l'indication ne peut plus uniquement produire des listes de termes, et ce de façon indépendante du reste du système. Interrogation et indexation doivent coopérer afin de restituer au mieux à l'utilisateur sa propre vision du corpus.

Par ailleurs cette seconde voie permettrait également d'envisager une typologie des systèmes en identifiant non seulement le nombre de documents satisfaisants, mais aussi en connaissant le sémantique de leur pertinence. Il serait alors intéressant de travailler sur de nouvelles mesures, ou bien dans un premier temps de voir sur un nouveau jour les mesures actuelles. Par exemple, l'hallucination pourrait être transformée en une mesure dite ''découverte'', montrant la capacité du système à montrer d'autres documents à l'utilisateur : la proximité plus ou moins grande de ces documents avec le besoin de l'utilisateur lui permettait alors de découvrir dans le corpus des documents répondant moins bien à sa requête certes, mais lui donnant une satisfaction partielle. Ainsi soit sa question est une question ouverte et l'utilisateur est content de regarder dans d'autres

directions, soit l'utilisateur ne trouve pas ou pas assez de bonnes réponses et il accepte de regarder d'autres documents.

1.4.3. Pour une recherche en indexation :

Quelle que soit la mesure choisie, l'expression des performances d'un système en donne une vision globale. En ce sens, elle ne permet pas de justifier les performances d'un système par le rôle de chacun des éléments de son architecture ou encore d'identifier dans un système un paramètre particulièrement performant.

Le constat actuel est que cette vision macroscopique des systèmes a pour conséquence de privilégier les travaux au niveau de l'interrogation (interface, fonction de correspondance,...) : Afin d'améliorer les performances, les chercheurs raisonnent généralement uniquement au niveau de la fonction de correspondance, dont la plupart raisonne sur des langages à mots clés. Cependant, l'autre processus du système, l'indexation, a beaucoup moins été étudié ; et peu de travaux s'effectuent sur l'indexation, et qu'elle est encore trop souvent considérée comme une boite noire relativement externe au système, et fournissant en sortie des listes de mots clés.

La première des raisons est qu'il est vrai que pendant longtemps l'indexation a réellement été une boite noire externe au système, et fournissent en sortie des listes de mots clés. En effet des documents n'étaient pas électriques, et l'indexation était produite mutuellement sous forme de listes de mots clés, par des documentalistes particulièrement entraînés. Lorsque les documents textuels électroniques sont apparus, les méthodes automatiques d'indexation sont apparues également. Leur seul objectif était de fournir une indexation similaire à celle produite manuellement, et donc des listes de mots clés. Il est alors vrai et naturel que dans ce contexte, le seul paramètre avec lequel le système peut être amélioré est l'interrogation.

Depuis une dizaine d'années, le développement des données électroniques, le stockage des données multimédia, la mise à disposition des systèmes de recherche d'informations à un public, qui peut aller du néophyte au spécialiste, ont permis l'émergence d'une nouvelle vision de la recherche d'informations, et notamment de mettre au premier plan la nécessité d'indexer autrement : il n'est plus possible d'imaginer que l'indexation d'un document puisse être uniquement une liste fixe de mots clés. Flexibilité, adaptativité, richesse d'expression sont les conditions nécessaires à l'expression d'une indexation. Par ailleurs, son intégration dans le système, son utilisation paramétrée en font dorénavant une pièce maîtresse pour la satisfaction de l'utilisateur.

En ce sens, l'indexation d'un document n'est pas universelle, mais dépendante de la perception de chacun. C'est dans cette nouvelle vision de la recherche d'informations est décliné cette voie selon deux axes principaux :

- Indexation complexe afin de restituer fidèlement le contenu des documents ;
- Système dynamique d'indexation et de recherches. Interrogation et indexation coopèrent afin de produire pour chaque utilisateur une vision adaptée du corpus. L'indexation doit dorénavant justifier chaque terme d'indexation par le rôle effectif qu'il joue dans le document.

1.5. L'indexation en recherche documentaire

Actuellement, l'accroissement du volume d'informations nécessaires à la bonne marche des entreprises et la diversification de ces informations sont tels que l'analyse, le stockage et la recherche de celle-ci sont devenus des domaines de recherche et de développement à part entière. L'information doit pouvoir être disponible dans les plus brefs délais et localisable de la manière la plus simple et précise que possible.

Ces informations, relativement bien structurées dans un premier temps et donc enregistrables et exploitables par l'intermédiaire d'un système de gestion de bases de données, se présentent de plus en plus sous la forme de textes, résumés, notices, documentations, que nous regrouperons sous le terme générique de données textuelles . Ces données textuelles ne sont, bien évidemment, pas faciles à utiliser avec un système de gestion de bases de données et ce n'est d'ailleurs pas leur intérêt. Elles tirent leurs qualités de leur quantité et de leurs provenances diverses. Cette grande quantité les rend, hélas, inexploitables telles quelles, elles doivent donc subir un traitement préalable, c'est de cela que découle le but de l'indexation.

1.5.1. Le rôle de l'indexation

L'indexation a pour rôle de représenter de façon homogène le contenu sémantique des documents du corpus. L'homogénéité de l'indexation réfère à la conformité, de cette représentation, à un langage d'indexation définissant (en extension ou en intention) les termes d'indexation utilisables. La notion de termes d'indexation est à prendre dans ce contexte au sens large, et le terme d'indexation présente toute forme produite par l'indexation d'un document, quelle que soit sa complexité.

Le terme « indexation » encapsule donc deux problèmes distincts :

- La définition d'un langage d'indexation, permettant l'extraction des concepts des documents du corpus.
- La mise en place d'un processus d'indexation permettant l'extraction, à partir des documents du corpus, des termes d'indexation, c'est à dire de leur représentation conforme au langage d'indexation.

Même si elle est très vague, cette première définition est cependant consensuelle. Par exemple Borko et Bernier disent " indexing is the process of analysing the informational content of records of knowledge and expressing the informational content in the language of indexing system", alors que Willish donne une définition plus technocrate "an operation intended to represent the results of analysis of a document by means of a controlled or natural indexing language". De même Rowley écrit "the indexing process creates a description of a document or information, usually in some recognizes and accepted style or format." Salton [Salton, 1983] complète cette définition, en ajoutant trois objectifs à l'indexation :

- "To allow the location of documents dealing with topics of interest to user"
- "To relate documents to each other, and thus relate the topic areas, by identifying distinct documents dealing with similar, or related, topic areas."
- "To predict the relevance of individual documents to specific information requirements through the use of index terms with well-defined scope and meaning."

Ces différentes définitions montrent la dualité de l'indexation : représenter le contenu des documents afin de permettre aux utilisateurs de les retrouver. Ces deux objectifs sont difficiles à réunir, et la recherche en indexation le montre bien. En effet dans la plupart des travaux, l'indexation est soit orientée document soit orientée requête. L'indexation orientée document a pour objectif de résumer ou de présenter le contenu de chaque document, c'est à dire son signifiant et son signifié. L'indexation orientée requête doit, pour chaque document, refléter les requêtes pour lesquelles il est pertinent : l'indexation d'un document doit alors représenter les raisons pour lesquelles un utilisateur consulte ce document.

1.5.2. Indexation orientée document :

L'indexation orientée document consiste à définir, à partir du document seulement, son contenu, que l'on qualifie dans ce contexte d' « à-propos ». Lancaster décrit cette indexation comme : " a conceptual analysis, witch, first and foremost, involves deciding what document is about - that is, what it covers" .Indexer un document revient à définir le processus qui permet de passer de la forme ou du signal d'un document à son fond, en d'autres termes de son signifiant à son signifié. Déterminer le signifié d'un document est une démarche délicate et subjective, car beaucoup de paramètres interviennent dans cette identification : la qualité du signifiant, l'indexeur, la base de connaissances,...

Cette notion d'à-propos prend une dimension très complexe sur les données images, audio et vidéo, où la partie subjective des documents est délicate à déterminer. En effet, leur signifiant est constitué physiquement d'un signal numérique et leur signifié est perçu par un sens (l'écoute, la vue). Contrairement aux données textuelles, la distance entre signifiant et signifié est importante. De plus la perception du signifié se modifie non seulement en fonction des connaissances, mais également d'une

personne à une autre. Il suffit pour le prouver de montrer une même image à différentes personnes, leur demander ce qu'elles observent. Le passage du signifiant au signifié devient alors un problème particulièrement complexe. La littérature met souvent en évidence la difficulté des indexeurs (humain ou non) à mener cette tâche, et les tests de consistance d'indexation le montrent par les désaccords entre indexeurs.

1.5.3. Indexation orientée requête

La façon la plus classique de procéder à une indexation orientée requête est d'anticiper les requêtes et donc de confronter chaque document de la base à une liste de requêtes prédéfinies. La liste de requêtes forment alors le langage d'indexation.

Certains utilisent la méthode suivante pour déterminer un langage d'indexation : pour tout document du corpus, un groupe de documentalistes répond à la question « pourquoi un de nos utilisateurs serait-il intéressé par ce document ? » En répondant à cette question par une liste de termes, les documentalistes génèrent ainsi un langage d'indexation, Ensuite, le processus d'indexation procède à l'indexation grâce à un filtrage : l'indexeur vérifie chaque terme associé a priori à un document et se demande : « est-ce que l'un de nos utilisateurs intéressé par ce document utiliserait ce terme pour formuler sa requête ?».

Les problèmes majeurs posés par une indexation orientée requête résident dans son évolution face à de nouvelles requêtes, et surtout dans la difficulté à l'automatiser.

1.5.4. Quelle indexation choisir ?

Les indexations proposées dans les systèmes de recherche d'information doivent être mixtes. En effet, les besoins des systèmes sont doubles :

- Afin de servir le spectre le plus large d'utilisateurs, le système doit disposer du maximum d'informations sur les documents. De ce point de vue, l'indexation doit donc être orientée document.
- Afin de servir au mieux un utilisateur, de ce point de vue, l'indexation doit être orientée requête.

Entre autre, une prééminence des travaux de recherche en indexation orientée document. Ceci s'explique relativement facilement par le fait que d'une part, il est difficile de disposer d'une liste représentative de requêtes, alors que d'autre part le corpus de documents est disponible et donc utilisable.

Cependant, ces approches peuvent générer un dysfonctionnement du système par rapport à ses utilisateurs en oubliant que l'indexation d'un document n'est pas unique. Cependant un certain nombre d'approches actuelles intègrent cette personnalisation du système, ces approches définissent en fait l'indexation orientée document comme une « couche basse » de l'indexation, à partir de laquelle ils proposent une indexation orientée requête. En effet la finalité de ces systèmes est de

satisfaire pleinement le besoin des utilisateurs et donc de savoir parfaitement adapter (et non fixer) leur indexation à ces spécificités.

1.6. EXTENSION A LA PAROLE

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 TREC (Text REtrieval Conference, campagne d'évaluation) comme le présente la figure 1.3. Cette tâche est associée à la recherche d'information dans des documents papier numérisés par Optical Caracter Recognition (OCR) car, dans les deux cas, les erreurs introduites peuvent être assimilées à un bruitage du contenu linguistique initial. La tâche SDR de TREC consiste à indexer 500 heures d'émissions radio en anglais, en utilisant les transcriptions automatiques. 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.

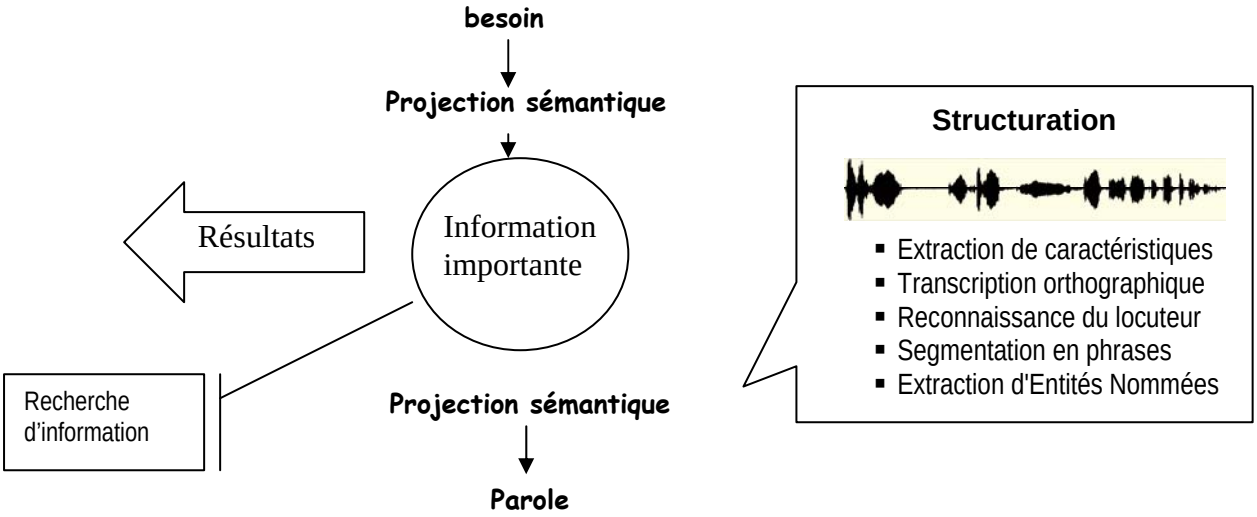


Figure 1.3 : Schéma bloc du système de recherche documentaire audio (SDR)

Les évaluations TREC montrent que le taux d'erreur de 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. 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. 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.

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 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. 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. 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[1] 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. Autre travaux 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 Favre [Favre, 2007].

[1] le mot « treillis » est utilisé dans le sens de graphe d'hypothèses, de l'anglais lattice.

Chapitre 2

Reconnaissance de la Parole

2.1 Introduction

La reconnaissance automatique de la parole est un domaine de recherche très vaste, et surtout avec l'avancement terrible des technologies multimédia. Entre outre, l'avancement dans les technologies d'acquisition et de stockage des documents audio et leurs exploitation via le réseau mondial, impose une manipulation automatique avec des systèmes de reconnaissances de la parole.

Le signal de parole est caractérisé par plusieurs paramètres intrinsèques qui rendent sont interprétation délicate. Cependant, le signal est très varié ; soit d'un locuteur à un autre, ou même pour un locuteur lui-même. Les différences d'age, de sexe, d'accent entre locuteurs, d'origine sociale rendent délicates l'extraction d'informations pertinentes caractérisant le signal, cette extraction doit être indépendante du locuteur. En plus le signal acoustique dans des milieux ambiants comme les bruits extérieurs, bruits de bouches ou respirations et aussi la qualité d'enregistrement génèrent également des difficultés pour de le système de reconnaissance de la parole. En plus, le type de signal de parole à traiter : mots isolés, parole continues ou paroles spontanées augmente le degré de complexité de la tâche de reconnaissance.

2.1.1 Complexité du signal de parole

Le signal de parole est une combinaison de plusieurs événements acoustiques des différents organes de l'appareil phonatoire et l'impact du milieu extérieur.

2.1.1.1 Redondance

Dans le domaine temporel, le signal acoustique est redondant, ce qui impose un traitement préalable avant d'entamer la procédure de reconnaissance. En effet, il existe une disproportion entre le débit du signal enregistré et la quantité d'informations manipulées lors d'une tâche de reconnaissance. Un signal échantillonné à 16 Khz sur 16 bits représente un débit de 256 Kbit/s, sachant que dans la phase de reconnaissance on cherche une dizaine de phonèmes à la seconde. Dans ce contexte, il faut représenter le signal acoustique dans un espace plus compacte. Il existe un grand nombre de paramètres possibles, souvent dérivés d'une analyse spectrale. Le choix de ces paramètres est lié étroitement avec le problème à traiter, car les paramètres qui ne sont pas pertinents pour la reconnaissance deviennent décisifs pour l'identification du locuteur.

2.1.1.2 Continuité et coarticulation

Le flux de parole est un ensemble de mots, qui peuvent à leur tour être décrits comme une suite de symboles élémentaires appelés phonème par les linguistes. De plus, la parole est un processus séquentiel, au cours du quel des unités indépendantes se succèdent. Malheureusement, les spécialistes de phonétique eux-mêmes ont parfois des difficultés à identifier individuellement ces unités discrètes dans le signal. La parole est en réalité un flux continu, et il n'existe pas de pause entre les mots qui pourrait faciliter leur localisation automatique par les systèmes de reconnaissance.

De plus, la production d'un son est fortement influencée par les sons qui le précèdent mais aussi par ceux qui le suivent en raison de l'anticipation du geste articulatoire. L'identification correcte d'un segment de parole isolé de son contexte est parfois impossible. La prise en compte des phénomènes de coarticulation ne suffit pas à prédire la réalisation acoustique d'une phrase en raison de la grande variabilité de la parole.

2.1.1.3 Variabilité

On distingue généralement deux sources de variabilité pour deux flux de parole qui contiennent la même phrase : la variabilité inter-locuteurs et variabilité intra-locuteur : la variabilité inter-locuteurs est due à la différence physiologiques entre locuteurs, qu'il s'agisse de la longueur du conduit vocal ou du volume des cavités résonantes. En plus, des habitudes acquises en fonction du milieu social et géographique, comme les accents régionaux. La variabilité intra-locuteur est plus réduite, mais n'est pas négligeable. L'état physique, par exemple la fatigue ou le rhume, les conditions psychologiques, comme le stress, et même le bruit de fond pendant l'élocution, influent sur la production du flux de parole.

2.1.2 Les défis de la reconnaissance

Plutôt que d'affronter simultanément toutes ces difficultés, il est préférable de simplifier le problème de la reconnaissance automatique de la parole en se limitant à des sous problèmes. Les difficultés de mise au point d'un système de reconnaissance de la parole dépendent des conditions d'utilisation du système, qui sont caractérisées par leur degré de liberté, du plus contraint au plus libre, dans les domaine suivants :

- Le nombre d'utilisateurs du système : celui-ci peut être mono-locuteur, multi-locuteurs ou indépendant du locuteur.
- La taille du vocabulaire : petit vocabulaire (moins de mille mots), grand vocabulaire (moins de cent mille mots) ou très grand vocabulaire (plus de cent mille mots).
- La complexité du langage utilisé : langage contraint par une syntaxe artificielle ou langage naturel.
- Le mode d'élocution : mots isolés ou parole continue.
- La robustesse aux conditions d'enregistrement : système nécessitant de la parole de bonne qualité ou fonctionnant en milieu bruité.

2.2 Du signal de parole à l'observation acoustique

2.2.1 Modules acoustiques

Les premiers modules de traitement dans un système de reconnaissance de la parole sont les suivants :

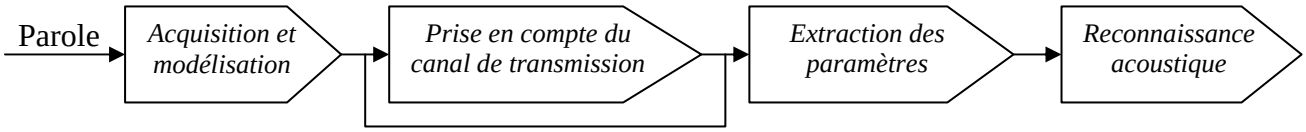


Figure 2.1 : chaîne de traitement acoustique d'un système de reconnaissance de la parole

Comme le montre la figure 2.1, le signal de parole est d'abord numérisé puis modélisé sous une forme généralement fréquentielle. Pourtant, avant d'obtenir ces mesures, le signal a subi des modifications dues à l'environnement dans lequel se trouve le locuteur, à l'influence du système d'acquisition, et à une éventuelle transmission par le biais d'un média informatique, par exemple un réseau. Ces modifications sont souvent regroupées sous le terme générique de « canal de transmission ». Certains systèmes de reconnaissance disposent d'un module de prise en compte de ce canal pour tenter d'éliminer son influence sur le signal de parole. Le module suivant, dans la chaîne de traitement acoustique, est celui qui extrait des paramètres pertinents pour la reconnaissance de la parole. Ces paramètres sont ensuite envoyés au module de reconnaissance acoustique qui identifie les sons présents dans le signal.

Détaillons chacun de ces modules pour comprendre l'enchaînement allant du signal de parole à l'observation acoustique. En ce qui concerne le module de reconnaissance acoustique, nous ne présenterons que la technique de reconnaissance la plus employée à l'heure actuelle : la modélisation par modèles de Markov. C'est celle que nous utilisons, nous aussi, dans notre système de détection de mots clés. Nous n'aborderons pas la technique d'alignement temporel (Dynamic Time Warping en anglais) intégrée, par exemple, dans les téléphones portable. Nous ne présenterons pas non plus les approches fondées sur les réseaux de neurones ou sur des approches hybrides mélangeant modèles de Markov et réseaux de neurones.

2.2.1.1 Acquisition et modélisation du signal

2.2.1.1.1 Numérisation

Pour être utilisable par un ordinateur, un signal doit tout d'abord être numérisé. Cette opération tend à transformer un phénomène temporel analogique, le signal sonore dans notre cas, en une suite d'éléments discrets, les échantillons. Ceux-ci sont obtenus avec une carte spécialisée courante de nos jours dans les ordinateurs depuis l'avènement du multimédia. La numérisation sonore repose sur deux paramètres : la quantification et la fréquence d'échantillonnage.

La quantification définit le nombre de bits sur lesquels on veut réaliser la numérisation. Elle permet de mesurer l'amplitude de l'onde sonore à chaque pas de l'échantillonnage. De plus, cette quantification peut suivre une échelle linéaire ou logarithmique, cette dernière privilégiant la résolution de la quantification pour les niveaux faibles au détriment des niveaux forts.

Le choix de la fréquence d'échantillonnage est aussi déterminant pour la définition de la bande passante représentée dans le signal numérisé. Le théorème de Shannon [Bellanger 95] nous indique que la fréquence maximale *fmax* présente dans un signal échantillonné à une fréquence *fe* est égale à la moitié de *fe*. Un signal échantillonné à 16000 Hertz contient donc une bande de fréquences allant de 0 à 8000 Hertz. D'après ce principe, il est donc inutile de numériser un signal téléphonique à plus de 6800 Hertz, car le résultat ne contiendrait pas plus d'informations fréquentielles. Pourtant, comme la majorité des cartes ne proposent que certaines fréquences d'acquisition, le signal téléphonique est généralement échantillonné à une fréquence de 8000 Hz, ce qui, de plus, facilite la définition de filtres fréquentiels.

2.2.1.1.2 Transformée de Fourier

Joseph Fourier a montré que toute onde physique peut être représentée par une somme de fonctions trigonométriques appelée série de Fourier. Elle comporte un terme constant et des fonctions sinusoïdales d'amplitudes diverses. Ainsi un son sinusoïdal ne comporte qu'une seule raie spectrale correspondant à la fréquence de sa fonction sinus. Un son complexe est composé d'une multitude de ces raies spectrales qui représentent sa composition fréquentielle [Bellanger 95].

Dans le cas d'une séquence d'échantillons, il est alors possible de calculer une Transformée de Fourier Discrète (TFD, Discret Fourier Transform - DTF - en anglais). L'équation 2.1 donne le calcul de la FFT pour une séquence $X(n)$ comportant N échantillons.

$$X(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)e^{-jk2\pi(n/N)}$$

Équation 2.1 : formule de la Transformée de Fourier Discrète

En 1965, [Cooley et al. 65] ont proposé un algorithme de calcul rapide de transformée de Fourier discrète, la Fast Fourier Transform (FFT, Transformée de Fourier Rapide - TFR - en français). La seule limitation de cet algorithme est que la taille de la séquence dont on veut obtenir la FFT doit être une puissance de 2. Le temps de calcul d'une FFT est environ 10 fois inférieur à celui d'une TFD classique. Le lecteur pourra trouver de plus amples informations à propos de ces algorithmes et des implémentations commentées dans [Press et al. 92].

2.2.1.2 Prise en compte du canal de transmission

Comme dans la tâche de reconnaissance de la parole, l'une des deux entités de la chaîne de communication est un ordinateur. De ce fait, il est nécessaire de prendre en compte le canal de transmission entre l'être humain et la machine, car celui-ci introduit des distorsions qui sont de nature à perturber suffisamment le signal de parole pour le rendre difficilement reconnaissable pour la machine. Ce canal de transmission est en général assimilé à un filtre. Il est possible d'inclure dans ce canal des informations comme la réponse impulsionnelle de la pièce où l'enregistrement est effectué, ou encore le bruit de fond.

Si l'on prend comme exemple l'enregistrement via un microphone, la réponse en fréquence de ce dernier introduit une distorsion qui modifie les fréquences identifiables dans le signal. Si l'enregistrement de la voix est réalisé par le biais d'une ligne téléphonique, la réduction fréquentielle est encore plus forte. En effet, dans ce cas, la bande passante se situe entre 300 et 3400 Hertz, ce qui élimine toutes les autres fréquences. De plus, avec l'arrivée des serveurs de reconnaissance distribuée [Klautau et al. 00], le canal peut aussi comporter une transmission via le réseau Internet. Dans ce cas, nous parlerons de transmission de Voix sur IP (VoIP pour Voice over IP en anglais) [Black 00]. Les applications de visioconférence, entre autres, emploient de tels protocoles. Dans ce cas-là, le canal provoque non seulement une distorsion due au codage de la voix mais aussi, du fait que l'implémentation de ces protocoles est basée sur UDP/IP, une perte de paquets et donc de données dans le signal à reconnaître.

Il existe plusieurs façons de s'affranchir du canal par lequel le signal passe pour obtenir des résultats optimaux de reconnaissance. Il faut soit réduire la différence entre les données servant à apprendre les modèles de reconnaissance, soit réaliser un prétraitement pour annuler les effets du canal. Ces deux méthodes posent néanmoins des problèmes. La première méthode nécessite d'avoir une connaissance du canal et de pouvoir construire des bases acoustiques pour l'apprentissage des modèles acoustiques. La seconde, souvent basée sur des filtres adaptatifs, permet de s'adapter en cours de reconnaissance. Dans ce cas, il est nécessaire de connaître le type du canal.

2.2.1.3 Extraction de paramètres

Nous avons vu comment l'ordinateur appréhendait un signal sonore. Pourtant les formes temporelles ou fréquentielles ne sont pas les plus adéquates pour la reconnaissance de la parole continue. Il est nécessaire de calculer plusieurs paramètres dérivés de ce signal. Nous n'aborderons ici que les principaux utilisés dans la littérature, et par nous-même dans notre système d'expérimentation.

2.2.1.3.1 Énergie du signal

Après la phase de numérisation et surtout de quantification, le paramètre intuitif pour caractériser le signal ainsi obtenu est l'énergie. Cette énergie correspond à la puissance du signal. Elle est souvent évaluée sur plusieurs trames de signal successives pour pouvoir mettre en évidence des variations. La formule de calcul de ce paramètre est :

$$E(\,fenêtre\,) = \sum_{n \in\, fenêtre} |n|^2$$

Équation 2.2: Calcul de l'énergie d'un signal échantillonné

Il existe des variantes de ce calcul. L'une des plus utilisées réalise une simple somme des valeurs absolues des amplitudes des échantillons pour alléger la charge de calcul, les variations

restant les mêmes. D'autres, comme celle de [Taboada et al. 94] proposent la modification suivante du calcul intégrant une normalisation par rapport au bruit ambiant.

$$E(\textit{fenêtre}) = \log(\sum_{n \in \textit{fenêtre}} \frac{|n|^2}{R})$$

Équation 2.3 : calcul de l'énergie normalisé par rapport au bruit ambiant

Dans cette équation, R est la valeur moyenne de l'énergie du bruit. Le résultat de ce calcul tend vers 0 lorsque la portion considérée est une zone où il n'y a que le bruit de fond. Tout le problème de cette variante réside dans l'estimation du facteur de normalisation R.

2.2.1.3.2 Mel-scaled Frequency Cepstral Coefficients (MFCC)

Les MFCC sont utilisés en reconnaissance de parole et en identification du locuteur ou de la langue car ces paramètres sont bien adaptés au signal de parole. Ils sont issus de l'hypothèse suivante, à savoir que le signal de parole est le résultat de la convolution entre un filtre (conduit vocal) et une excitation (cordes vocales) :

$$x_n = g_n * b_n \qquad \textbf{(2.1)}$$

avec x_n le signal, g_n l'entrée et b_n le filtre caractérisant le conduit.

Une transformation homomorphique permet de transformer ce produit en une somme qui est ensuite filtrée pour obtenir les MFCC : « Mel Frequency Cepstral Coefficient ». Ces MFCC permettent une déconvolution entre la source des sons produits (caractéristiques du locuteur) et le conduit oral (couplé ou non au conduit nasal) :

$$\tilde{x}_n = \tilde{g}_n + \tilde{b}_n \qquad \textbf{(2.2)}$$

La transformation homomorphique se décompose en trois étapes principales comme le montre la figure 2.2

- un passage dans le domaine spectral par calcul du module de la transformée de Fourier rapide,
- une application du logarithme,
- un retour au domaine temporel par calcul de la transformée de Fourier rapide inverse.

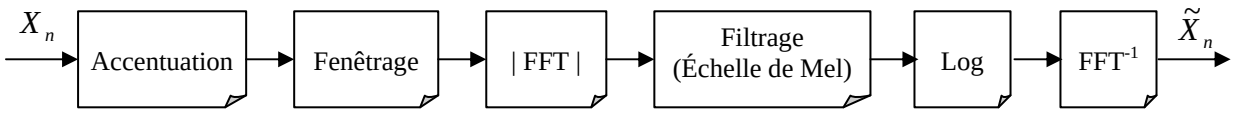


Figure 2.2 : Processus de création des coefficients cepstraux.

où

FFT : Transformée de Fourier Rapide (passage dans le domaine spectral).

FFT^{-1} : Transformée de Fourier Rapide Inverse (retour dans le domaine temporel).

Le calcul de la FFT se fait sur des fenêtres glissantes.

Une accentuation des aigus est présente car les composantes fréquentielles aiguës sont toujours plus faibles que les graves. Un filtrage de type passe-haut est réalisé avec la fonction de transfert :

$$H(z) = 1 - 0.98 * z^{-1} \qquad \textbf{(2.3)}$$

L'utilisation d'une *fenêtre de Hamming* (sur une trame acoustique de 256 ou 512 points en général) avec recouvrement sur la moitié (128 ou 256 points) permet d'éviter la formation d'artefacts liés aux effets de bord durant la transformation du domaine temporel au domaine fréquentiel :

$$W_{Ham\min g}(n) = \begin{Bmatrix} 0.54 - 0.46 * \cos\left(\frac{2\pi n}{N}\right) & pour \\ 0 & ailleurs \end{Bmatrix} \quad 0 \le n \le N-1 \qquad (\textbf{2.4})$$

avec N la taille de la fenêtre.

L'échelle non linéaire Mel est connue pour rendre compte de la perception humaine. Les coefficients sont appelés MFCC car dans le domaine spectral, ce changement d'échelle (utilisation de l'échelle perceptive Mel) est effectué. Ils ont la propriété d'être fortement décorrélés. Dans les systèmes de reconnaissance de la parole, le premier coefficient est souvent utilisé pour définir l'énergie.

Généralement une soustraction cepstrale se fait sur les MFCC pour déconvoluer le signal du bruit du canal (de la source d'enregistrement : micro, canal téléphonique...) et obtenir un signal paramétré débruité [Mok95]. Cette opération résulte du fait que les coefficients cepstraux de la parole ont une moyenne nulle ; pour ôter le bruit causé par le canal, il suffit alors de soustraire à chaque coefficient cepstral du signal bruité leur moyenne, représentative de la moyenne des coefficients cepstraux relatifs au bruit seul.

L'inconvénient majeur de la représentation cepstrale réside dans son manque de lisibilité : il ne s'agit pas d'une représentation directement liée aux informations qu'un expert peut extraire de la lecture d'un sonagramme, ce qui complexifie l'interprétation des paramètres.

Les travaux de Stevens [Stevens et al. 40] ont permis la mise en évidence de la *loi de puissance* ou *loi de Stevens* selon laquelle l'intensité de la perception d'un stimulus n'augmente pas linéairement en fonction de sa puissance mais de façon exponentielle en tenant aussi compte des modalités de l'expérimentation. Les coefficients MFCCs [Davis et al. 80] pour *Mel-scaled Frequency Cepstral Coefficients*, aussi nommés *Mel Frequency Cepstral Coefficients* dans la littérature, sont donc

basés sur une échelle de perception appelée Mel, non linéaire. Celle-ci peut être définie par la relation suivante entre la fréquence en Hertz et sa correspondance en mels :

$$M_{mels} = x \times \log\left(1 + \frac{f_{Hz}}{y}\right)$$

Équation 2.4 : correspondance entre l'échelle Mel et la fréquence en Hertz

Plusieurs valeurs sont utilisées pour *x* et *y*. En 1989, on trouvait dans [Calliope 89] *x = 1000/log(2)* et *y = 1000*. De nos jours, les valeurs les plus couramment utilisées sont *x = 2595* et *y = 700*. D'autres définitions de cette échelle peuvent être trouvées comme par exemple [Umesh et al. 99].

Pourtant l'utilisation de cette unité n'est pas suffisante. Pour avoir une largeur de bande relative qui reste constante, le banc de filtres Mel est construit à partir de filtres triangulaires positionnés uniformément sur l'échelle Mel donc non uniformément sur l'échelle fréquentielle. Cette répartition est illustrée ci-dessous :

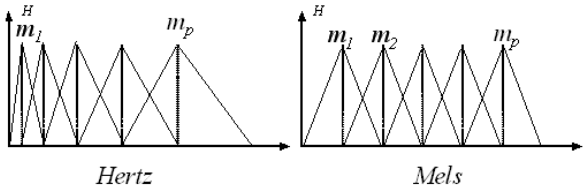


Figure 2.3 : répartition des filtres triangulaires sur les échelles fréquentielle et Mel

Sur cette illustration, m_p correspond au nombre de filtres que l'on souhaite. Lorsque ce banc de filtres est en place, il est alors possible de calculer les coefficients MFCCs. L'algorithme peut être décrit comme suit

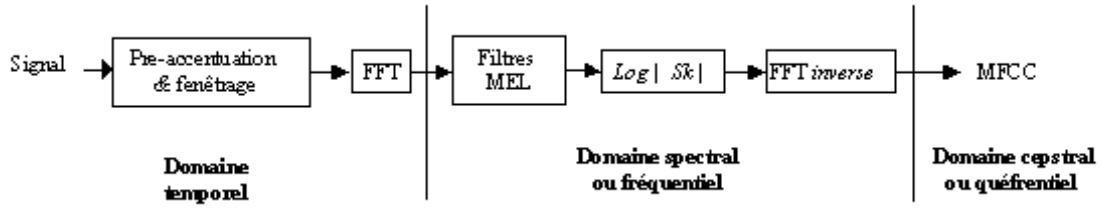


Figure 2.4 : algorithme de calcul des MFCCs

Il est possible de choisir le nombre de paramètres générés en sortie de cet algorithme. Dans la littérature, le nombre de coefficients utilisés varie de 5 à plus d'une quarantaine en fonction de l'utilisation qui en est faite : reconnaissance de la parole, de la langue ou identification du locuteur par exemple. . En ce qui concerne le nombre de filtres, nombreux sont ceux qui choisissent 30 pour un signal avec une bande passante de 0 à 8 KHz.

2.3 Décodage acoustico-phonétique à base de modélisation acoustique

D'après [Haton 1991], un décodage acoustico-phonétique (DAP) est défini généralement comme la transformation de l'onde vocale, en unités phonétiques - une sorte de transcodage qui fait passer d'un code acoustique à un code phonétique - ou plus exactement comme la mise en correspondance du signal et d'unités phonétiques prédéfinies (opération de couplage /identification) dans lequel le niveau de représentation passe du continu au discret.

Ce module est composé d'une première partie consistant à extraire les paramètres choisis pour représenter le signal, et d'une seconde partie qui, à partir de ces jeux de paramètres, apprend des modèles d'unités acoustiques ou décode le signal d'entrée, selon que l'on veuille apprendre ou reconnaître.

2.3.1 Modélisation acoustique

Les approches statistiques et les modèles probabilistes sont très utilisés, de nos jours, dans les systèmes de reconnaissance automatique de la parole. Ces approches, notamment celles basés sur les Modèles de Markov Cachés (HMM), ont atteint des performances remarquables avec des vocabulaires de plus en plus importants et une robustesse au bruit et à la variabilité des locuteurs de plus en plus grande [Rabiner 1993]

Dans les années 70, l'approche consistait en un paradigme de reconnaissance de mots « par l'exemple ». Ces premiers systèmes fonctionnaient à base de patrons de vecteurs acoustiques ou « *template-based systems* » en anglais [Huang 2001]. Le principe consistait à faire répéter plusieurs exemples des mots à reconnaître et à les analyser sous forme de vecteurs acoustiques dans un patron. Ensuite, pour reconnaître un mot inconnu, il « suffisait » de comparer le jeu de vecteurs acoustiques extraits du signal avec les suites d'exemples appris (ou enregistrés) précédemment. Ce principe de base n'est cependant pas implémentable directement parce qu'un même mot peut être prononcé de nombreuses de façons différentes, en changeant le rythme de l'élocution. La superposition du signal inconnu aux signaux de base doit dès lors se faire en acceptant une certaine « élasticité » temporelle, formalisée mathématiquement par l'algorithme *Dynamic Time Warping* (DTW) [Silverman 1990].

Cette approche pionnière s'est rapidement confrontée aux grands problèmes de la reconnaissance automatique de la parole. Comment faire face à la variabilité due aux locuteurs et au contexte d'enregistrement, comment élaborer une construction sémantique et non simplement lexicale et donc comment gérer de très grands dictionnaires ?

De tous ces obstacles sont apparus des unités acoustiques plus petites (en termes de temps) et les modèles probabilistes. Les unités acoustiques ne sont plus des mots mais des phonèmes. Le principe consiste alors à déduire des modèles de phonèmes plutôt que des exemples de mots. Ainsi, ces modèles sont beaucoup plus souples, dans le sens où ils couvrent beaucoup plus de variations et permettent la gestion de gros vocabulaires sans modifier le nombre d'unités acoustiques représentées. Les modèles peuvent être applicables pour n'importe quelle voix. Il est même possible de découper

encore ces petites unités acoustiques au sein du modèle lui-même. Enfin, ces unités acoustiques peuvent être non plus des phonèmes, mais des combinaisons de phonèmes, c'est-à-dire un phonème en fonction de son contexte. Par exemple, on modélisera le phonème [ɛ] suivi du phonème [dʒ], ou le phonème [ɛ] suivi du phonème [tʃ], etc. au lieu du phonème [ɛ]. Nous parlons alors de polyphones : diphones, triphones ou même quintphones selon le nombre de contextes pris en compte.

Pour la modélisation statistique acoustique, les modèles de Markov cachés (HMM) sont aujourd'hui utilisés dans un très grand nombre des systèmes de reconnaissance automatique de la parole. Chaque unité de parole est modélisée par un HMM. Dans le cas de petits lexiques, ces unités de parole peuvent être les mots. Dans le cas de grands lexiques, on préférera souvent utiliser des modèles de phonèmes (ou polyphones), ce qui limitera le nombre de paramètres à estimer. Dans ce dernier cas, lors de la reconnaissance, les mots seront construits (dynamiquement) en termes de séquences de phonèmes et les phrases en termes de séquences de mots.

Les HMMs supposent que le phénomène modélisé est un processus aléatoire et inobservable qui se manifeste par des émissions elles-mêmes aléatoires. Cette approche markovienne offre une flexibilité séduisante de modélisation pour un phénomène aussi complexe que la parole.

2.3.2 Modèles de Markov Cachés

Les modèles de Markov cachés sont apparus dans la problématique de la reconnaissance automatique de la parole dans les années 70 [Baker 1975, Jelinek 1976]. L'idée sous-jacente est que la parole peut être caractérisée par un processus aléatoire dont les paramètres peuvent être estimés d'une manière appropriée. Les modèles HMM ont prouvé leur efficacité dans de nombreux domaines de la reconnaissance automatique de la parole. Cependant, les modèles ont été améliorés au fil des recherches pour repousser leurs limites intrinsèques, particulièrement en intégrant des notions de corrélation entre trames et de modélisation de trajectoires.

Un modèle de Markov discret est un automate stochastique à nombre d'états fini N [Rabiner 1993]. Un processus aléatoire se déplace d'état en état à chaque instant et on note q_t l'état atteint par le processus à l'instant t.

Dans le cas des modèles de Markov discrets cachés, l'état réel q_t n'est pas directement observable mais le processus émet un symbole discret après chaque changement d'état. Les observations ne sont plus univoquement liées à une seule classe bien déterminée mais sont donc des fonctions statistiques de ces classes qui ne sont plus observées directement. Ou encore, les états du modèle ne sont plus observés directement à partir des observations qui sont supposées être produites par ces états mais à travers une fonction statistique différente pour chaque classe. On a donc un processus doublement stochastique : modèle stochastique relatif au modèle de Markov sous-jacent et celui décrivant la relation entre les classes (états) et les observations. La figure 2.4 présente un modèle de Markov caché ergodique, où toutes les transitions entre tous les états sont autorisées.

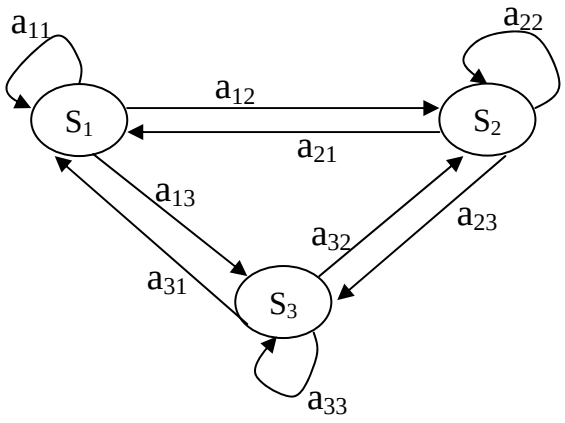


Figure 2.5 : Exemple de modèle de Markov caché ergodique

Concrètement, un modèle de Markov caché HMM est représenté par $\lambda = (N, A, B, \pi)$ qui est caractérisé par les éléments suivants :

- N : est le nombre de nœuds ou d'états du modèle ;
- $S = \{s_1, s_2, ..., s_N\}$: un ensemble des états du modèle avec N le nombre d'états. On note q_t l'état à l'instant t ;
- $O = \{o_1, o_2, ..., o_M\}$: un alphabet des observations avec M nombre fini de symboles d'observation par état. Les symboles d'observation correspondent à chaque sortie physique du système réel qu'on modélise. On note x_t l'observation à l'instant t ;
- $A = \{a_{ij}\}$: une matrice des probabilités de transition entre états, dont a_{ij} est la probabilité de transition de l'état i à l'état j. On a :

$$a_{ij} = P(q_t = s_j | q_{t-1} = s_i), \quad 1 \le i, j \le N \qquad \textbf{(2.5)}$$

$$\sum_{j=1}^{N} a_{ij} = 1, \quad 1 \le i \le N \qquad \textbf{(2.6)}$$

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

$$b_i(k) = P(x_t = o_k \mid q_t = s_i), \quad 1 \le i, j \le N \qquad \textbf{(2.7)}$$

- $\pi = \{\pi_i\}$: une matrice de distribution de l'état initial. On a :

$$\pi_i = P(q_0 = s_i) \quad 1 \quad 1 \le i \le N \qquad \textbf{(2.8)}$$

$$\sum_{i=1}^{N} \pi_i = 1 \qquad \textbf{(2.9)}$$

Pour la modélisation acoustique, le modèle souvent utilisé est donc le modèle HMM gauche-droit (ou de Bakis), illustré par la figure 2.6, dans lequel on ne peut pas revenir à un état précédent, et où seules les transitions non nulles sont représentées.

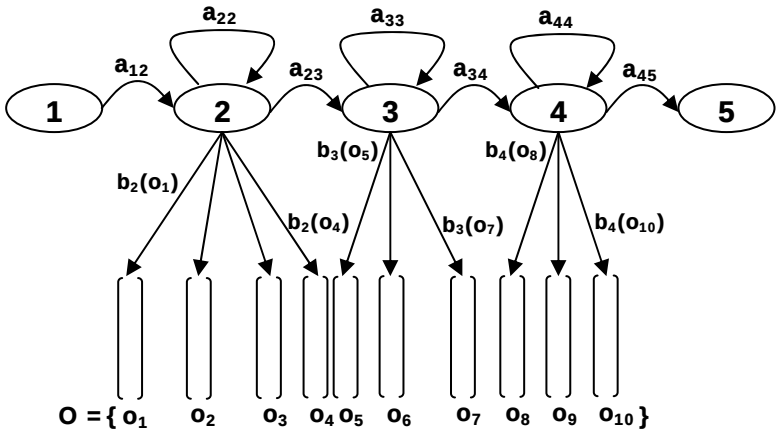


Figure 2.6 : Exemple de HMM à 3 états gauche-droit

2.3.3 Application à la reconnaissance

Soit ***o*** une sous séquence de ***O***. La reconnaissance de la séquence de mots ***M*** passe par le calcul de la probabilité *a posteriori* qu'un modèle acoustique génère la sous séquence ***o*** :

$$P(\lambda|o) \qquad (2.10)$$

La probabilité *a posteriori* que la séquence ***M*** ait été prononcée dans la séquence totale ***O*** s'obtient par un processus de concaténation qui relie les états de sortie de chaque modèle aux états d'entrée de tous les autres par des probabilités [Rabiner and Juang, 1993].

La loi de Bayes permet de réécrire l'expression :

$$P(\lambda/o) = \frac{P(o/\lambda)P(\lambda)}{P(o)} \qquad (2.11)$$

et l'opération de reconnaissance se réduit à maximiser $P(o/\lambda)P(\lambda)$, le produit de la *vraisemblance* de la séquence d'observations ***o*** étant donnée le modèle λ par la probabilité *à priori* du modèle.

$P(\lambda)$, la probabilité du modèle acoustique λ est obtenue par le modèle de langage. Le modèle de langage, parfois appelé grammaire règle l'enchaînement des modèles acoustiques lors de la reconnaissance.

Maximiser $P(o/\lambda)$ revient à chercher la séquence optimale des états de λ, c'est-à-dire la séquence d'états qui explique au mieux la séquence des observations $o = \{o_1, o_2, ..., o_T\}$.

L'utilisation des HMMs dans un système de reconnaissance suppose de pouvoir résoudre les trois problèmes suivants [Rabiner and Juang, 1993] :

- Evaluation : Etant donnée une séquence d'observations $O = \{o_1, o_2, ..., o_T\}$ et le modèle $\lambda = (N, A, B, \pi)$, comment calculer $P(o/\lambda)$?
- Décodage : Etant donnée une séquence d'observations $o = \{o_1, o_2, ..., o_T\}$ et le modèle $\lambda = (N, A, B, \pi)$, comment déterminer la séquence d'états $Q = \{q(1), q(2), ..., q(T)\}$ qui explique le mieux ***o*** ?
- Apprentissage : Comment déterminer les paramètres du modèle $\lambda = (N, A, B, \pi)$ afin de maximiser $P(o/\lambda)$?

2.3.3.1 Evaluation

Soient le modèle $\lambda = (N, A, B, \pi)$, $O = \{o_1, o_2, ..., o_T\}$ une séquence d'observations et $Q = q_1, q_2, ..., q_T$ une séquence d'états. La probabilité d'observer la séquence O pour une séquence d'états Q est

$$P(O/Q, \lambda) = b_{q1}(o_1) \times b_{q2}(o_2) \times ... \times b_{qT}(o_T) \qquad \textbf{(2.12)}$$

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

$$P(Q/\lambda) = \pi_{q1} a_{q1q2} a_{q2q3} ... a_{qT-1qT} \qquad \textbf{(2.13)}$$

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

$$P(O, Q/\lambda) = P(Q/\lambda) \times P(O/Q, \lambda) \qquad \textbf{(2.14)}$$

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

$$P(O/\lambda) = \sum_{Q} P(O, Q/\lambda) \qquad \textbf{(2.15)}$$

$$P(O, \lambda) = \sum_{q_1 q_2, ..., q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) ... a_{a_{T-1} q_T} b_{q_T}(o_T) \qquad \textbf{(2.16)}$$

Pour un modèle à N états et une séquence d'observations de durée de T le calcul de cette probabilité nécessite $(2T - 1)N^T$ multiplications et $N^T - 1$ additions. Cependant, il est possible d'obtenir cette solution plus efficacement, en faisant intervenir l'algorithme *avant-arrière* (Forward-Backward).

2.3.3.1.1 Principe de l'algorithme *:*

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

De la même manière, soit la probabilité arrière $\beta_t(j) = P(o_{t+1}, o_{t+2}..., o_T / q_t = j, \lambda)$, la probabilité d'observer la séquence $o_{t+1}, o_{t+2}..., o_T$ sachant qu'on est à l'état i au temps t et qu'on a le modèle λ. De la même façon, cette probabilité est calculée d'une manière récursive.

2.3.3.1.2 Algorithme Avant :

- Initialisation :

  Pour i=1 jusqu'à N

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

- Itérations :

  Pour t =1 jusqu'à T-1

  Pour j=1 jusqu'à N

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

- Terminaison :

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

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

2.3.3.1.3 Algorithme Arrière :

- Initialisation :

  Pour i=1 jusqu'à N

  $$\beta_T(i) = 1$$

- Itérations :

  Pour t =T-1 jusqu'à 1

  Pour i=1 jusqu'à N

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

- Terminaison :

  $$P(O/\lambda) = \sum_{i=1}^{N} \alpha_T(i) = \sum_{i=1}^{N} \pi_i b_i(o_1) \beta_1(i) = \sum_{i=1}^{N} \alpha_t(i) \beta_t(i)$$

2.3.3.2 Problème de décodage

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

$$\gamma_t(i) = P(q_t = i/O, \lambda) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)} = \frac{\alpha_t(i)\beta_t(i)}{P(O/\lambda)}$$

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

L'état optimal à un instant t sera donc :

$$q_t = \arg_i \max[\gamma_t(i)]$$

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

Si ces critères sont tout à fait adaptés à certaines applications, le critère le plus utilisé est celui qui cherche la meilleure séquence d'états globale (le meilleur chemin), c'est-à-dire qui maximise $P(Q/O, \lambda)$. Une technique formelle existe pour calculer ce chemin optimal, il s'agit de *l'algorithme de viterbi* [Yassine BenAyed, 2003].

2.3.3.2.1 Principe de l'algorithme *:*

Pour trouver la meilleur séquence d'états $Q = q_1, q_2, ..., q_T$, connaissant une séquence d'observations $O = o_1, o_2, ..., o_T$, on a besoin de définir la quantité $\delta_t(i)$.

$$\delta_t(i) = \max_{q1,q2,...,qt-1} P(q_1, q_2, ..., q_t = i, o_1, o_2, ..., o_t/\lambda)$$

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

$$\delta_{t+1} = \max_{1 \le i \le N} [\delta_t(i) a_{ij}] b_j(o_{t+1})$$

2.3.3.2.2 Algorithme :

- Initialisation :

  Pour i=1 jusqu'à N

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

$$\psi_1(i) = 0$$

- Itérations :

  Pour t =2 jusqu'à T

  Pour j=1 jusqu'à N

$$\delta_t(j) = \max_{1\le i\le N}\left[\delta_{t-1}(i)a_{ij}\right]b_j(o_t)$$

$$\psi_t(j) = \arg_{1\le i\le N}\max\left[\delta_{t-1}(i)a_{ij}\right]$$

- Terminaison :

$$P^* = \max_{1\le i\le N}\left[\delta_T(i)\right]$$

$$\psi_T^* = \arg_{1\le i\le N}\max\left[\delta_T(i)\right]$$

- Recherche :

  Pour t=T-1 jusqu'à 1

$$q_t^* = \psi_{t+1}(q_{t+1}^*)$$

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

2.3.3.3 Problème d'apprentissage

Les paramètres des modèles sont les probabilités de transition entre états et les probabilités d'émission associées aux états. La topologie du modèle (le nombre d'états des modèles et les transitions autorisées entre ces états) est supposée fixée a priori. Ainsi, connaissant une suite d'observations émises par un modèle, il est possible de modifier les paramètres du modèle de manière à rendre plus probable l'émission des observations par le modèle. Il s'agit d'une estimation par le critère du maximum de vraisemblance (*Maximum Likelihood Estimation*, MLE), obtenue par l'algorithme de Baum-Welch [Baum, 1970].

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

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

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

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(o_{t+1}) \beta_{t+1}(j)}{P(O/\lambda)}$$

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

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

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

$$\overline{\pi}_i = \gamma_1(i), \qquad 1 \le i \le N$$

$$\overline{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}, \quad 1 \le i \le N, \quad 1 \le j \le N$$

$$\overline{b}_j(k) = \frac{\sum_{t=1, o_t=k}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \lambda_t(j)}, \quad 1 \le j \le N$$

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

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

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

En se basant sur cette procédure, si nous utilisons itérativement le modèle $\overline{\lambda}$ à la place de λ et si nous répétons l'étape de la ré-estimation des paramètres. Nous pouvons alors améliorer la probabilité que O soit observée sachant le modèle jusqu'à atteindre un certain point limite.

Le résultat final de la procédure de ré-estimation est le maximum de vraisemblance du HMM. Il existe d'autres critères d'apprentissage, comme les critères MAP (Maximum A Posteriori) ou MMI (Maximum Mutual Information), mais leur mise en œuvre est généralement plus difficile [BENAYED, 2003].

2.4 Définition des modèles

Les principes fondamentaux de la modélisation markovienne, introduits dans la section précédente, laissent une grande marge de manoeuvre pour leur mise en œuvre concrète au sein d'un système de reconnaissance.

En premier lieu, il convient de définir le modèle utilisé dans le système. Pour cela, trois éléments doivent être précisés : l'unité de modélisation, la topologie des modèles et le type de fdp associées aux états. Les choix concernant ces éléments seront le plus souvent dictés au concepteur du système par sa connaissance experte du domaine. Quelques procédures automatiques ont toutefois été proposées pour s'affranchir de l'approche heuristique sur certains points.

2.4.1 Unité de modélisation

L'unité de modélisation définit le symbole qui sera représentée par un HMM. Trois critères permettent de guider le choix entre les alternatives. Les allophones contextuels constituent un des meilleurs compromis entre ces critères dès lors que des techniques adaptées permettent de résoudre la difficulté de leur apprentissage.

2.4.1.1 Critères de choix

De façon générale, l'unité de modélisation est imposée par la tâche qui fixe l'unité symbolique à reconnaître. Toutefois, lorsque les symboles peuvent être exprimés à l'aide de symboles plus élémentaires, il peut se révéler préférable de définir ces sous symboles comme unités de modélisation. Lorsque le choix de la décomposition n'est pas unique, trois critères doivent être pris en compte pour déterminer la meilleure unité de modélisation :

- La consistance : des occurrences différentes d'une même unité ont des réalisations physiques proches ;
- La capacité d'apprentissage : on dispose pour chaque unité d'un nombre suffisant d'exemples pour garantir une estimation robuste des paramètres du modèle ;
- La capacité d'expression : la relation entre les unités de modélisation et les unités symboliques peut être facilement obtenue ; par exemple au moyen d'un lexique phonétisé.

Le dernier point est rarement abordé du fait que les relations entre les unités symboliques et les unités de modélisation sont issues de la connaissance experte du domaine, exception faite de

l'approche par unités acoustiques automatiques sans signification phonologique appelé fénones. La recherche du compromis entre consistance et capacité d'apprentissage a conduit, en reconnaissance de la parole, à la coexistence de différentes approches.

2.4.1.2 Choix de l'unité

Les mots constituent l'unité symbolique à reconnaître a priori dans le cadre de la RAP. Ils sont donc des candidats de choix pour l'unité de modélisation. Lorsque la reconnaissance de la parole se limitait à la reconnaissance de mots isolés ou de petits vocabulaires, la modélisation par mots se révélait très efficace [Rabiner, al, 1989]). Ces unités sont consistantes car elles modélisent de façon implicite les co-articulations intra-mots.

Pour la RAP continue à grand vocabulaire, il devient complexe d'apprendre un modèle pour chaque mot. De plus, chaque nouveau mot introduit dans le système nécessitera un apprentissage particulier. Le recours à une modélisation sub-lexicale doit donc être envisagée pour satisfaire le critère lié à la capacité d'apprentissage. Ces unités peuvent être des phonèmes, des phonèmes contextuels, des concaténations de phonèmes (e.g. les diphones [Schwartz, Klovstad et al., 1980]), des syllabes [Hunt, Lenning et al., 1980] ou encore des unités acoustiques automatiques sans signification phonologique comme les fénones [Bahl, Brown et al., 1993].

L'unité sub-lexicale la plus immédiate est le phonème : il représente le son le plus bref qui permet de distinguer tous les mots d'une langue donnée. Cette définition est d'ordre phonologique et ne tient pas compte de l'acoustique. Les réalisations physiques d'un phonème peuvent varier considérablement en fonction de facteurs tels que le contexte (co-articulation), la vitesse d'élocution, le dialecte, le style et le locuteur. Les occurrences acoustiques particulières d'un phonème sont appelées allophones.

Les langues comportent un petit nombre de phonèmes : quelques dizaines pour l'arabe comme pour le français. Cette approche est donc satisfaisante du point de vue de sa capacité d'apprentissage. Elle l'est beaucoup moins du point de vue de sa consistance. Parmi les facteurs de variation cités plus haut, l'influence de la co-articulation peut être diminuée par la prise en compte du contexte lors de la modélisation. On distingue :

- les *multi-phonèmes* formés d'une concaténation de phonèmes. Le plus courant est le diphone qui représente deux phonèmes successifs ;
- les *syllabes* dont le statut linguistique est assez bien défini mais dont la localisation est difficile ;
- les *allophones contextuels* qui modélisent les phonèmes suivant leur contexte phonétique. Les unités allophoniques se distinguent des multi-phonèmes car leur support temporel est celui d'un seul phonème.

Dans chacun de ces cas, les unités sont consistantes mais difficiles à apprendre en raison de leur nombre élevé. L'approche par allophones contextuels est plus utilisée car elle présente des caractéristiques favorables à l'application de techniques d'apprentissage visant à compenser leur

nombre élevé, telle que la mise en commun entre plusieurs allophones de la partie centrale du phonème.

Toutes les unités de modélisation précédentes ont en commun d'être issues de connaissances expertes du domaine de la parole. Une approche non-paramétrique au problème de la modélisation sub-lexicale a été introduite par les chercheurs d'IBM [Bahl, Brown et al., 1993]. Leur proposition repose sur un découpage de l'espace de représentation acoustique. Chaque zone est associée à une unité élémentaire, le fénone.

2.4.1.3 Apprentissage des modèles d'allophones contextuels

De nombreuses techniques ont été développées pour améliorer la capacité d'apprentissage des modèles d'allophones contextuels. En effet, pour n phonèmes différents, on obtient n^3 allophones contextuels gauche et droit (appelés des triphones bien qu'il ne s'agisse pas de multi-phonèmes). Pour le français comme pour l'arabe, une cinquantaine de phonèmes sont généralement considérés ; ce qui conduit à environ 125.000 triphones. La taille de la base de données acoustiques d'apprentissage peut alors devenir insuffisante pour apprendre correctement chacun des modèles. Il faut alors soit diminuer le nombre de modèles soit diminuer le nombre de paramètres indépendants dans le système jusqu'à ce que chacun dispose de suffisamment de données d'apprentissage. De plus, un certain nombre d'allophones contextuels peut ne pas être rencontré du tout dans la base.

Parmi les techniques proposées, les allophones généralisés consistent à regrouper dans une même classe les allophones présentant des réalisations acoustiques proches. Le regroupement peut-être obtenu de plusieurs manières : par une classification automatique basée sur une mesure de similarité entre modèles [Lee, 1990], par la construction d'un arbre de décision [Bahl, de Souza et al., 1991] ou par l'utilisation de connaissances phonologiques [Ljolje, 1994]. Il a aussi été proposé de modéliser d'abord séparément l'influence des contextes gauches et droits ; puis de les interpoler au sein d'un nouveau modèle appelé quasi-triphone [Ljolje, 1994].

2.4.2 Topologie des modèles

La topologie d'un modèle regroupe le choix du nombre d'états ainsi que la définition d'une matrice de transition initiale. Ce choix est fortement dépendant de l'unité de modélisation choisie.

Les états des HMM peuvent être interprétés comme des zones stationnaires du signal. Les début et fin des réalisations phonétiques présentant des caractéristiques assez différentes de leur centre, plusieurs états sont nécessaires. La topologie gauche-droite à 3 états s'est révélée bien adaptée à la RAP continue. Les techniques d'inférence automatique peuvent être mises en oeuvre pour améliorer l'adéquation entre les données et la topologie des modèles.

2.4.2.1 Modèle gauche-droit

Le modèle gauche-droit traduit la causalité du processus de production de la parole : il n'existe pas de cycle dans le graphe orienté engendré par les transitions entre états. Les modèles gauche-droit particuliers, autorisant le bouclage à l'état courant, le passage à l'état suivant ou le saut

d'un état sont le plus couramment utilisés pour représenter les unités phonétiques. Trois états sont généralement utilisés (Figure 2-6). [Barras, 1996] confirme ce choix en testant sur la base de données TIMIT des systèmes utilisant des topologies phonétiques de 1 à 4 états ; les meilleurs taux de reconnaissance en DAP sont obtenus par les modèles à 3 états.

Des structures plus complexes ont été proposées : la topologie à 7 états de l'équipe d'IBM [Derouault, 1987] qui est utilisée dans le système SPHINX du CMU [Lee, 1988] ou encore le modèle allophonique du CNET [Jouvet, 1995].

2.4.2.2 Inférence de topologie

Quelques tentatives ont été menées pour déterminer automatiquement les topologies des modèles. Parmi celles-ci, un certain nombre sont fondées sur les techniques d'inférence grammaticale appliquées aux chaînes de symboles obtenues après quantification vectorielle des vecteurs de paramètres acoustiques [Casacuberta, Vidal et al., 1990; Lockwood & Blanchet, 1993]. Par ailleurs, une approche heuristique a été proposée consistant à optimiser une structure a priori par un recuit simulé guidé par la réduction du taux d'erreur [de Mori, Galler et al., 1995].

Jusqu'à présent, les topologies obtenues par ces techniques n'ont pas été jugées suffisamment pertinentes pour être retenues ; principalement car le coût de calcul pour les obtenir n'était pas justifié par leur apport en terme de performance.

La topologie du modèle fixe les valeurs initiales des éléments de la matrice de transition A ainsi que leur nombre. Il reste pour définir entièrement un modèle à définir le vecteur B en associant à chaque état une fdp.

2.4.3 Estimateur de probabilité

Deux groupes d'estimateur de probabilités peuvent être distingués : les estimateurs paramétriques et non-paramétriques. En RAP, la distinction entre les estimateurs est couramment faite sur une autre de leur caractéristique liée à la continuité de l'espace d'évaluation des fdp. On distingue les estimateurs continus et l'estimateur discret par quantification vectorielle.

2.4.3.1 Estimateurs continus

Les estimateurs continus sont fondés sur le principe que la fdp à estimer appartient à une famille de fonctions. Ils définissent explicitement la famille de fonctions (par exemple les gaussiennes pour les GMM) et associent une somme pondérée (mixture) de fonctions à chaque classe. La somme pondérée associée à une classe est obtenue par l'apprentissage de ses paramètres sur un ensemble de données de référence appartenant à cette classe. Des variantes existent comme les estimateurs semi-continus qui évaluent un ensemble de fonctions a priori représentant au mieux l'ensemble des données de référence, seuls les coefficients de la somme pondérée étant particuliers aux états.

2.4.3.1.1 Principes

Les estimateurs de probabilité continus sont définis par une somme pondérée de fonctions élémentaires. La plus communément utilisée est la somme pondérée de fdp gaussiennes multidimensionnelles (GMM) :

$$b_i(x_0) = \sum_{g=1}^{G} c_{ig} N(x_0; \mu_{ig}, \sum_{ig} \quad) \textbf{(2.22)}$$

où μ_{ig} et $\sum_{ig}$ sont respectivement le vecteur moyenne et la matrice de covariance de la $g^{ième}$ gaussienne de l'état i et c_{ig} le coefficient de pondération qui lui est affecté. L'hypothèse d'une indépendance entre les d dimensions de l'espace de représentation autorise l'utilisation de matrices de covariance diagonales ; ce qui limite le nombre de paramètres à estimer et simplifie les calculs.

L'approche paramétrique de l'estimation de fdp est validée par la théorie des noyaux [Parzen, 1962; Cacoullos, 1966]. Elle montre qu'il est possible d'estimer n'importe quelle fdp en tout point x_0 d'un espace multidimensionnel par une combinaison de fonctions :

$$\frac{1}{nh_n^d} \sum_{j=1}^{n} K\left(\frac{x_0 - x_j}{h_n} \right) \xrightarrow{n\to\infty} f(x_0) \qquad \textbf{(2.23)}$$

avec $x_1 \ldots x_n$ n observations de la variable aléatoire X de distribution f. La suite $\{h_n\}$ converge vers 0 et K est une fonction noyau (i.e. elle est bornée, d'intégrale unitaire et vérifie $\lim_{|y|\to\infty} |y|^d K(y) = 0$).

La théorie des noyaux fonde les principes de la construction d'une fonction à partir d'échantillons de ses valeurs. La distinction entre les approches paramétriques et nonparamétriques tient dans le choix des échantillons pris en compte : l'approche paramétrique considère des points moyens représentatifs des modes de l'ensemble des échantillons initiaux, l'approche non-paramétrique utilise l'ensemble des échantillons disponibles lors de l'estimation.

La loi normale respecte les contraintes des fonctions noyaux et représente donc un candidat de choix pour (2-23). D'autres fonctions ont été utilisées dans les systèmes de RAP : certaines sont des raffinements de la fonction gaussienne comme les sommes pondérées de gaussiennes auto-régressives [Juang & Rabiner, 1985], d'autres sont basées sur des fonctions très proches de la gaussienne comme les sommes pondérées de densités de Richter [Richter, 1986] ou les sommes pondérées de densités de Laplace [Ney & Noll, 1988]. Quelle que soit la fonction choisie, elle doit satisfaire à deux types de contraintes : les contraintes statistiques qui déterminent la qualité de l'estimation obtenue et les contraintes de modélisation permettant d'intégrer l'estimateur choisi dans le formalisme des HMM. Ainsi, afin de montrer la convergence des formules de réestimation utilisées dans l'algorithme de Baum-Welch dans le cas des estimateurs continus, la fonction élémentaire retenue doit posséder une symétrie elliptique [Liporace, 1982]. Cette hypothèse s'est substituée à celle de log-concavité, plus contraignante. L'ensemble des fonctions citées précédemment respecte cette contrainte.

2.4.3.1.2 Variantes

Les variantes de l'estimateur continu, les plus répandues, reposent sur le principe du partage de paramètres appliqué aux fdp des états. Elles interviennent dans les systèmes semi-continus [Huang & Jack, 1989; Huang, 1992] et à sommes pondérées liées (tied-mixture) [Bellagarda & Nahamoo, 1990]. Dans ces systèmes, toutes les fdp sont regroupées et partagées par toutes les classes. La fdp associée à un état est définie par un vecteur de G coefficients de pondérations, G étant le nombre total de fdp du système. Cette approche est finalement peu différente du cas général : tous les états possèdent une somme pondérée de G gaussiennes mais celles-ci sont partagées par tous les états, seules les pondérations sont associées aux états. La technique du partage des paramètres permet de montrer l'équivalence entre les deux estimateurs. Elle a aussi permis d'appliquer ce procédé à tous les niveaux d'un système markovien (état, modèle). [Duchateau, 1998] propose le terme de modèles réduits (*reduced models*) lorsque les estimateurs de fdp sont partagés par des sous-ensembles d'états.

Les systèmes à estimateur semi-continu offrent a priori une moins grande précision de modélisation que les systèmes à estimateur de probabilité continu. La contrepartie est qu'ils nécessitent moins de calculs dès lors qu'un moins grand nombre de gaussiennes doit être réévalué pour chaque trame (la somme pondérée de gaussiennes est un calcul peu coûteux comparé au calcul de la gaussienne pour une vecteur donné). Cette approche peut donc être envisagée pour des considérations de temps de calcul ou lorsque le nombre de données disponibles pour estimer les fonctions pour chaque état est insuffisant ; notamment lorsque l'on fait croître de façon importante le nombre d'états indépendants du système (dans le cas des modèles contextuels par exemple).

2.4.3.1.3 Mise en œuvre

L'estimateur de fdp gaussien est l'estimateur le plus utilisé de l'état de l'art. Lors de la campagne d'évaluation DARPA Broadcast News 1997, un seul système sur dix engagés n'utilisait pas un estimateur continu et parmi les 9 autres, seul le système de Philips utilisait un estimateur à base de laplaciennes [DARPA, 1998].

L'estimateur continu permet d'ajuster la précision d'estimation à un coût de calcul donné ; notamment par le nombre de fonctions élémentaires utilisées par mixture. De plus, la compression d'information introduite par la modélisation des données au travers des paramètres de la fonction choisie permet un gain important en stockage. Toutefois, cette observation doit être nuancée si on observe les systèmes de l'état de l'art à base de GMM qui peuvent utiliser jusqu'à plusieurs centaines de milliers de gaussiennes (environ 300000 pour le système du LIMSI de l'évaluation DARPA Broadcast News 97 [DARPA, 1998]). Cette inflation du nombre de gaussiennes réduit le gain en compression d'information et augmente le coût de calcul.

Des techniques d'optimisation ont été développées. Elles visent soit à améliorer l'apprentissage des paramètres des estimateurs, soit à diminuer l'effort de calcul. Une procédure automatique basée sur la mesure du χ^2 a été proposée pour la détermination du nombre optimal de

gaussiennes par somme pondérée [Barras, 1996]. Cette procédure permet une réduction importante du nombre de paramètres sans baisse de performance. Divers procédés permettent l'accélération du calcul des mixtures de gaussiennes parmi lesquels la tesselation de l'espace de représentation [Ortmanns, Firzlaff et al., 1997] ou encore la technique de sélection des fonctions actives (Fast Removal of Gaussians, FRoG) [Duchateau, 1998].

Le principal inconvénient de l'estimateur continu est qu'il repose sur l'hypothèse que la fonction élémentaire utilisée est cohérente avec la loi réelle des données. Or, les vecteurs d'analyse n'ont pas, en général, de distribution gaussienne [Barras, 1996; Montacié, Caraty et al., 1996]. L'existence d'une somme particulière de fonctions qui converge vers la loi réelle est montrée théoriquement mais pas la manière de l'obtenir. Ainsi, le choix du nombre de fonctions élémentaires dans la somme pondérée, tout comme l'apprentissage de leurs paramètres, est guidé par des heuristiques et ne permet pas, en général, d'assurer une convergence vers la loi réelle.

2.4.3.2 Estimateur discret par quantification vectorielle

Le principe de l'estimateur de fdp discret par quantification vectorielle est de partitionner l'espace de représentation en associant à chaque zone un représentant. Celui-ci est soit le centre de gravité de la partition (*centroïde*), soit le vecteur le plus proche de tous les autres (prototype). L'habitude veut que l'on parle dans les deux cas de prototypes.

L'ensemble des prototypes constitue le dictionnaire (*codebook*) de l'espace de représentation discretisé. Divers algorithmes ont été proposés pour la construction du dictionnaire des prototypes. Le plus courant est l'algorithme des k-moyennes utilisé dans sa version LBG (*Linde-Buzo-Gray*) [Linde, Buzo et al., 1980]. Chaque vecteur de paramètres est ensuite associé au prototype du dictionnaire dont il est le plus proche au sens d'une certaine distance (classiquement une distance euclidienne pondérée comme la distance de Mahalanobis). La probabilité du vecteur sur un état est alors la probabilité du prototype pour cet état. Les probabilités des prototypes pour chaque état sont déterminées lors de l'apprentissage.

La taille du dictionnaire permet un contrôle de la distorsion introduite par la substitution d'un vecteur par son prototype. La contrepartie à cette perte de précision dans la modélisation des données est un coût de calcul faible : la distance d'un vecteur à tous les prototypes. Ce dernier point justifie son utilisation lorsque de fortes contraintes temporelles (par exemple un fonctionnement en temps réel) sont imposées au système. Le cas limite de l'estimation discrète, au prix d'une augmentation importante de son coût de calcul, consiste à conserver dans le dictionnaire tous les vecteurs de référence. L'estimateur discret devient alors l'estimateur des 1-ppv, cas particulier de l'estimateur non-paramétrique des K-ppv.

Une fois les modèles définis, une procédure automatique permet l'apprentissage des paramètres restés libres : les probabilités de transition et les paramètres des fdp des états.

2.5 Techniques d'apprentissage

L'algorithme de Baum-Welch est communément utilisé pour l'apprentissage initial des modèles. A partir des modèles initiaux, plusieurs techniques permettent d'améliorer la qualité de l'estimation des paramètres. La réestimation connectée est une variante de l'algorithme de Baum-Welch qui permet l'apprentissage simultané de l'ensemble des modèles. La technique du partage de paramètres permet d'augmenter artificiellement les données d'apprentissage associées à chacun des paramètres. Finalement, une adaptation des modèles peut être réalisée afin de les spécialiser sur un locuteur particulier.

2.5.1 Réestimation connectée

Dans l'apprentissage par l'algorithme de Baum-Welch, tel qu'il a été introduit dans, chaque modèle phonétique est appris séparément. Il est donc nécessaire de pouvoir associer les segments de données aux classes phonétiques modélisées par les HMM. Ceci suppose qu'il existe une segmentation a priori de la base de données. Cette opération, réalisée par des experts humains, est longue et complexe. C'est pourquoi peu de bases de données de parole disposent d'une segmentation phonétique experte. La méthode généralement retenue pour l'apprentissage des HMM consiste alors à générer des modèles initiaux à partir des données segmentées dont on dispose. Puis, ces modèles sont utilisés soit pour corriger automatiquement les erreurs de segmentation soit pour segmenter de nouvelles phrases à l'aide de leur transcription phonétique (ou lexicale).

La réestimation connectée permet la mise en oeuvre d'une telle approche. Connaissant la transcription phonétique (à l'aide d'un lexique phonétisé si l'on ne dispose que de la transcription lexicale) d'une séquence acoustique d'apprentissage, un réseau est constitué par la concaténation des modèles phonétiques correspondants. Ce réseau est ensuite traité par l'algorithme de Baum-Welch comme un modèle unique. La procédure d'apprentissage complète est reprise dans la Figure 2-7.

```
• Apprentissages séparés des modèles initiaux ;
• Réestimation connectée :
      Pour chaque phrase :
            Construction du réseau de modèles phonétiques ;
            Estimation du réseau par l'algorithme de Baum-
            Welch : accumulation des statistiques ;
      Mise à jour des paramètres pour tous les modèles ;
```

Figure 2-7 Déroulement de l'apprentissage d'un système markovien avec la réestimation connectée.

La réestimation connectée permet donc la prise en compte de données non segmentées lors de l'apprentissage des modèles. Elle conduit à augmenter considérablement le nombre de données

utilisables en minimisant le coût humain et financier. Malgré tout, lorsque le nombre de paramètres du système augmente mais que le nombre de données ne peux plus être augmenté, la technique de partage de paramètres doit être envisagée.

2.5.2 Partage de paramètres

La technique du partage de paramètres entre HMM permet s'assurer une meilleure généralisation des modèles avec des données en quantité limitée. Tous les paramètres (ou groupe de paramètres) d'un HMM peuvent être partagés, à commencer par le HMM en entier dans le cadre des modèles contextuels généralisés. Les formules de réestimation des procédures d'apprentissage ne sont pas modifiées par la technique du partage de paramètres [Bellagarda & Nahamoo, 1990].

Le partage de paramètres a deux avantages principaux :

- favoriser une estimation robuste des paramètres lorsque les données sont limitées. Ainsi, il est possible de concevoir des procédures automatiques qui regroupent un certain type de paramètres jusqu'à ce que la quantité de donnée associée à chacun d'entre eux soit suffisante.
- réduire la taille de stockage des modèles et accélérer la vitesse de calcul. Lorsque le mécanisme de partage est correctement implémenté, une seule occurrence des paramètres doit exister pour un grand nombre de pointeurs vers celle-ci. Ce qui a pour effet de diminuer la taille des modèles en mémoire. De plus, lors de l'utilisation de ces paramètres, un système de cache permet de ne procéder qu'à un seul calcul.

Le partage des paramètres suppose de regrouper les paramètres en classes partageant des caractéristiques acoustiques communes. De telles classes peuvent être obtenues à partir des différents critères qui seront détaillés dans le cas de la modélisation contextuelle.

Le réestimation connectée et la technique du partage de paramètres permettent de s'assurer d'un nombre suffisant de données pour l'apprentissage des modèles. La première augmente le nombre de données disponibles, la seconde adapte le nombre de paramètres indépendants des modèles au nombre de données disponibles. La précision de la modélisation peut être aussi améliorée par la prise en compte du locuteur.

Chapitre 3

Etat de l'art

3.1 Modélisation acoustique en intégrant les modèles Poubelle

L'application des systèmes de reconnaissance conventionnels pour la parole spontanée présente quelques problèmes car il faut considérer un grand vocabulaire et le langage doit être modélisé par une grammaire complexe qui considère tous les évènements possibles dans la parole spontanée, comme les phrases tronquées, les phrases grammaticalement incorrectes, toux, début incorrect, etc. Les systèmes de détection de mots clés ont fournit une solution aux processus de la parole spontanée. En effet le but des systèmes de reconnaissance conventionnels 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. Dans ces systèmes, les segments sémantiques significatifs sont extraits tandis que le reste est ignoré, le contenu sémantique peut être donc capté sans une reconnaissance détaillée de tous les mots de la phrase. Il suffit donc que les mots clés précédemment définis dans le vocabulaire soient détectés s'ils sont prononcés dans la phrase.

Les systèmes de détection des mots clés sont structurés généralement de deux parties : mots clés et des séquences de paroles qui contient les mots non clés (hors vocabulaire) et même de bruits. La réalisation d'un tel système donc implique de modéliser les mots clés (vocabulaire) affin d'accroître la détection et aussi modéliser les mots non clés pour réduire les fausses acceptations. La performance du système est validée par rapport au compromis entre ces deux tâches.

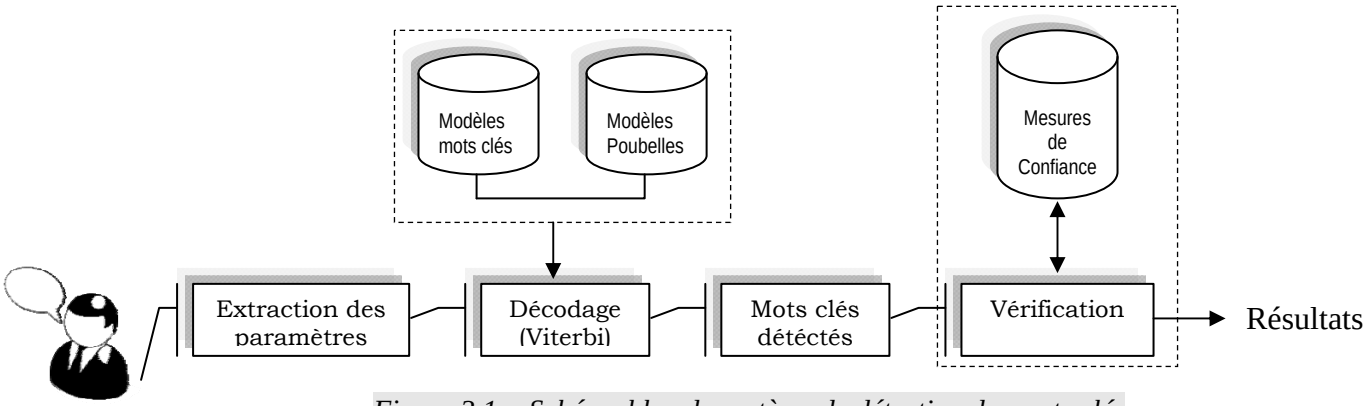


Figure 3.1 : Schéma bloc du système de détection des mots clés

Pour les systèmes de détection des mots clés. Il faut modéliser non seulement les mots clés avec un grand détail acoustique, mais aussi les mots non clés afin d'augmenter le taux de détection et de minimiser le nombre de fausse acceptation. Il existe plusieurs méthodes de détection des mots clés, parmi eux, la méthode à base des models poubelles. Les modèles poubelles sont utilisés pour modéliser les mots non clés, les faux départ et même le bruit pour absorber tous les mots hors vocabulaire (HV).

Dans la littérature, plusieurs travaux ont utilisé la notion des modèles poubelle [Rohlicek et al, 1989], [Rose et Paule, 1990], [Choy et Leung, 1998], [Cuayàhuitl et Serridge, 2002], [Szöke et al, 2005]

Rohlicek [Rohlicek et al, 1989], présente un système à base des modèles de Markov cachés avec une densité d'émission à base de mélange de gaussienne (HMM / GMM). Les mots clés sont modélisés par un HMM linéaire « gauche à droite » à trois états pour chaque phonème. Pour les modèles poubelles, il a utilisé deux configurations : La première basée sur un HMM avec un seul état ayant un mélange gaussien basé sur une pondération uniforme de toutes les distributions dans les états de mot-clé. La deuxième, un réseau de modèles où les composants de ces modèles sont des segments des modèles de mot-clé. Notons que la deuxième donne des résultats meilleurs que la première.

Rose [Rose et Paule, 1990] propose l'un des premiers systèmes de détection des mots clés utilisant les modèles HMM pour la reconnaissance de la parole continue à grand vocabulaire. Le principe de ce système est basé sur la séparation entre les mots clés et mots poubelles. Il a construit un réseau constitué de N mots clés et M mots poubelles, voir figure 3.1. Le point d'opération du système peut être ajusté par la configuration des poids des transitions $W_{k,1},...,W_{k,N}$ pour les mots clés et $W_{f,1},...,W_{f,M}$ pour les mots poubelles. Dans ce contexte, ce point d'opération réfère à un compromis entre le nombre de faux rejets et celui des fausses acceptations.

Le score associé à un mot clés décodé par l'algorithme de Viterbi est la vraisemblance de ce mot normalisée par sa durée. Le score S_W^{KW} pour un mot clés KW prononcé dans l'intervalle de temps de T_1 à T_F avec un état terminal s_F est donnée par la formule

$$S_{KW} = \frac{\log P\left(s_F, y_{T_1},...,y_{T_F}\right)}{T_F - T_1} \qquad \textbf{(3.1)}$$

avec y_t est un vecteur d'observation acoustique

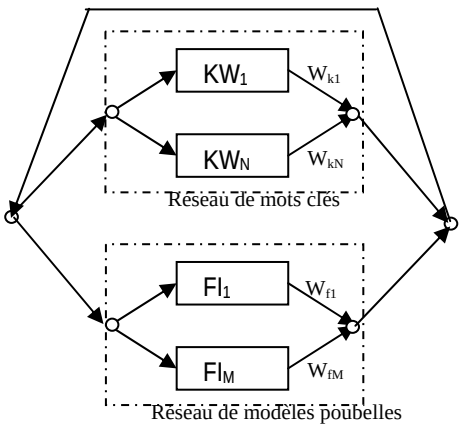


Figure 3.2 : Système de détection de mots clés qui intègre les modèles poubelles.

Pour réduire les fausses acceptation, un réseau parallèle « en arrière plan » de modèle de bruit a été inclus comme le montre la figure 3.3. Le score S_W^{KW} pour un mot clés peut être calculé pour une séquence de modèles de bruits qui se chevauchent avec un mot clé. Le score final en terme de probabilité devient alors :

$$S_{LR} = S_{KW} - S_{BA} \qquad \textbf{(3.2)}$$

avec S_{BA} est le score de recouvrement du bruit.

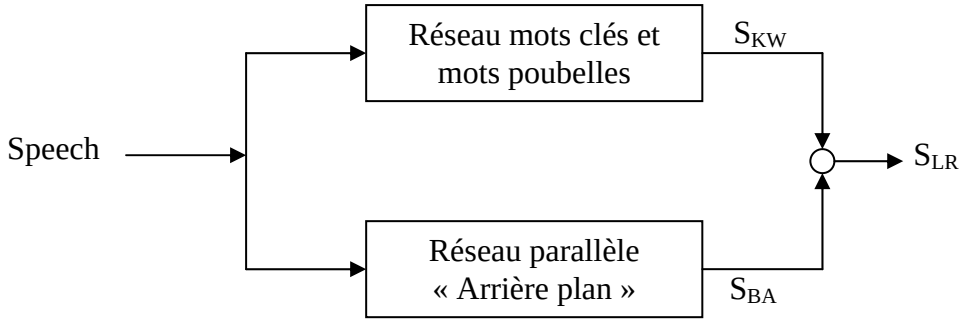


Figure 3.3 : Système de détection de mots clés qui intègre les modèles parallèles

Pour la construction du réseau des modèles poubelles, il a utilisé 80 modèles poubelles qui correspondent aux mots hors vocabulaire de la base d'apprentissage. Mais la majorité de ces mots n'existe pas dans la base de test. Pour remédier ce problème, il introduit des modèles de triphones composant les 80 mots hors vocabulaire. Il a obtenu ainsi, 268 triphones représentés par des HMM linéaires de trois états. Une amélioration des performances a été donc observée, de plus il y a parfois des chevauchements des contextes entre les modèles poubelles et les mots clés, ce qui implique une dégradation de la performance du système. En effet, à chaque chevauchement, le mot clés en question sera représenté par le modèle poubelle superposé. La solution qui a été proposée consiste à utiliser les phonèmes indépendants du contexte au lieu des triphones. Cette solution a donné des performances meilleurs que la précédente.

Choy [Choy et Leung, 1998] utilise des modèles poubelles à base des syllabes modélisées par des HMM à 5 états avec 8 mélanges gaussiennes. Assimilé au modèle mot-clé modèle, les états de HMM sont d'une topologie de gauche à droite, et la réestimation des modèles poubelles est réalisé avec L'algorithme Baum-Welch., sachant que les modèles poubelles sont indépendants du contexte.
Il a constaté que l'utilisation de plus petites unités de sous mots augmente les erreurs d'insertion. Donc les modèles poubelles à base de syllabe ont été utilisés dans les expériences pour la représentation de la parole continue.

Meliani [Meliani et O'Shughnessy, 1998] utilise deux types de modèles poubelles lexicaux à base de syllabes. Dans le premier type, chaque syllabe représente un modèle poubelle à part et dans

le deuxième type, un modèle poubelle contient toutes les syllabes ayant la même fréquence d'apparition dans le corpus d'apprentissage. En total, 12226 syllabes ont été récupérées de la base d'apprentissage, mais il leur faut une étape de filtrage pour ne laisser que les syllabes les plus fréquentes. Une série d'expérience a été menée afin de décider la fréquence minimale des syllabes à conserver. Les meilleurs résultats ont été obtenus pour une fréquence de 0.005%.

Cuayàhuitl [Cuayàhuitl et Serridge, 2002] propose un système de détection de mots clés avec une modélisation des mots hors vocabulaire pour la langue espagnol. Il a utilisé dans la tâche de détection des mots clés différentes unités de modèles poubelle a base de : phonèmes, syllabes, mots et combinaison de ce qui précède. Affin de réaliser un système robuste qui détecte des mots clés dans un flux de paroles sans contraintes.

Pour la modélisation des modèles poubelles, il a utilisé d'abord des modèles à base de phonèmes qui offre deux avantages : d'abord il permettent des expressions avec un vocabulaire flexible, de plus les mots non clés sont modélisés par les même unités ; et en second lieu, il simplifie pour un nouveau système la tâche de prévoir tous les mots non clés. Mais la contrainte principale de la parole continue c'est la coarticulation, et comme le phonème est influencé par ces phonèmes adjacents, les évidences phonologiques suggèrent que les unités à base de syllabes jouent un rôle important dans le traitement de la parole continue en particulier dans des situations acoustiques défavorables. Dans ce contexte, il a utilisé 860 syllabes différentes, bien que ce nombre soit un sous ensemble des syllabes possible de la langue espagnole.

Dans la détection de mots clés, l'utilisation d'un nombre élevé des modèles poubelle gaspille beaucoup de temps de calculs. Pour cela, il propose de les grouper en ensembles des syllabes acoustiquement similaires, avec ce groupement il a réduit le nombre à 344 groupes de syllabes, mais le coût de calculs reste toujours important. Affin de remédier à ce problème, il a introduit un autre niveau de classification sur la fréquence relative par rapport au moyen arithmétique des fréquences des groupes de syllabes : « Commun » ou « Rare ». Avec cette classification on arrive à 49 groupes de syllabes.

Szöke [Szöke et al, 2005] propose un nouveau système de détection de mots clés basée sur l'approche de maximum de vraisemblance. La structure générale du système est présentée dans la figure 3.4. Les parties A et C sont les modèles poubelles (boucle de phonèmes) qui modélisent la partie non mots clés du flux de parole. La partie B est le modèle des mots clés (linéaire). La partie D est le modèle de base (background) qui modélise la même partie du flux qui contient le modèle mot clés.

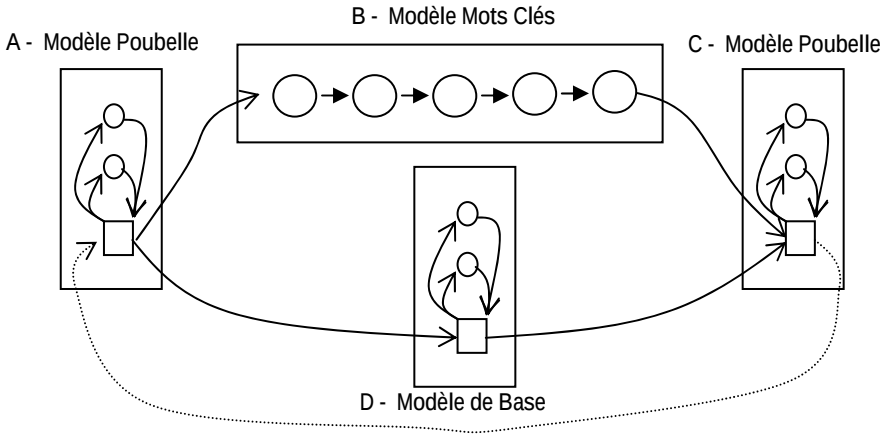


Figure 3.4 : Structure générale du système proposée

Le décodage du signal acoustique est exécuté en deux reprise : premièrement à l'aide du modèle ABC la concaténation des modèles A, B et C ; et ensuite à l'aide du modèle ADC la concaténation des modèles A, D et C. Entre autre, il décrit l'hypothèse que les modèles B et D reconnaîssent exactement la même partie de flux de parole. Donc la différence entre la probabilité final du modèle ABC (LABC) et le model ADC (LADC) est dûe à la différence entre les modèles B et D. Et puis, il calcule la vraisemblance LR= LADC / LABC qu'il utilise comme un seuil de détection.

3.2 Mesures de confiances

Le système de détection des mots clés (KWS) 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 des ces tâches 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 où 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.

Les meilleur taux de reconnaissance étant étroitement lié au nombre de fausses acceptations, nous devons essayer au cours de l'étape de vérification de réduire le nombre de fausses acceptations sans réduire considérablement le taux de reconnaissance. 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 réduire le nombre de fausses acceptations.

Plusieurs travaux dans la littérature exploitent les mesures de confiances dans la tâche de détection des mots clés. Dans ce contexte le processus de vérification devient un problème de classification. 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. Dans ce qui suit, nous présentons quelque approches utilisées.

3.2.1 Programmation dynamique

C'est l'approche la plus ancienne dans la détection des mots clés, elle est basée sur le principe de recalage temporel dynamique (Dynamic Time Warping (DTW)). Cette approche exploite un algorithme basé sur l'estimation de la distance.

Silaghi [Silaghi et bourlard, 2000] propose une 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 du 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. Pour cela, les auteurs proposent une modification sur l'algorithme de décodage de Viterbi.

Etant donnée une séquence d'observations acoustiques $X = \{x_1,...,x_n,...,x_N\}$ dans laquelle on veut chercher un mot clé et M le modèle HMM du mots clé m qui contient L états $Q = \{q_1,...,q_l,...,q_L\}$. On suppose que dans cette séquence on ne peut détecter qu'un seul mot clé m dont la sous séquence correspondante est : $X_b^e = \{x_b,...,x_e\}$ avec$1 \le b \le e \le N$. Dans ce contexte, il s'agit d'une mise en correspondance entre la séquence X de longueur N et le modèle HMM $\overline{M}$ qui contient la séquence d'états $\overline{Q} = \{q_G,...,q_G,q_b,q_{b+1},...,q_e,q_G,...q_G\}$avec $b-1$ états poubelles q_G qui précèdent x_b et $N-e$ états poubelles qui suivent l'état q_e. Ces états émettent respectivement les séquences des vecteurs X_1^{b-1} et X_{e+1}^N associés aux segments des mots non clés. Etant donnée une estimation de la probabilité $P(q_G/x_N)$ (obtenue en utilisant les fonctions des densités de probabilités apprises sur les mots non clés), le chemin optimal $\overline{Q}^*$ est donc donnée par :

$$\overline{Q}^* = \arg\min_{\overline{Q}\in\overline{M}} \left\{ -\log P\left(Q/X_b^e\right) - \sum_{n=1}^{b-1} \log P\left(q_G/x_n\right) - \sum_{n=e+1}^{N} \log P\left(q_G/x_n\right) \right\} \qquad \textbf{(3.3)}$$

Cette équation peut être résolue par une application directe de la programmation dynamique. Le problème principal est de trouver la meilleure estimation de la probabilité $P(q_G/x_N)$ afin de

minimiser l'erreur introduite par les différentes approximations. L'algorithme suivant estime la probabilité $P(q_G/x_N)$ d'une façon itérative.

3.2.1.1 Algorithme :

- Hypothèse :
  $\varepsilon = -\log(P(q_G/x_N))$
- Initialisation :
  $\varepsilon = -\log$ du maximum des probabilités a posteriori locales $P(q_G/x_N)$ pour chaque trame x_n
- Itération t :
  On cherche le chemin optimal $(\overline{Q}_t, b_t, e_t)$ selon l'équation précédente étant donnée la valeur estimée ε de $P(q_G/x_N)$
- Itération t+1 :
  La valeur estimée ε_t est définie comme étant la moyenne des probabilités a posteriori locales le long du chemin optimal Q_t.

  $$\varepsilon_{t+1} = \frac{1}{e_t - b_t + 1} \log P\left(Q_t / X_{b_t}^{e_t}\right)$$

  Retour à Initialisation
- Terminaison :
  On cherche la convergence. Si nous ne sommes pas intéressés par une segmentation optimale, ce processus peut être arrêté dès que ε atteint un seuil minimum qui nous permet de déclarer que le mot en question est détecté.

3.2.2 Algorithmes de Viterbi et de Baum-Welch

Le modèle HMM peut être utilisé pour réaliser la reconnaissance ou la détection des 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.

Rohlicek [Rohlicek et al, 1989] présente une méthode pour estimer la probabilité de détecter un modèle de mot clé à l'instant t. Le score $S_f(w,t)$ du mot w à l'instant t utilise seulement le score $\alpha(s,t)$ de l'algorithme de Baum-welch à l'état s, tel que :

$$S_f(w,t) = \frac{\alpha(e_w,t)}{\sum_s \alpha(s,t)} \qquad \textbf{(3.4)}$$

Où e_w est le dernier état du mot w.

La sommation se fait sur l'ensemble de tous les états s. Le maximum local est recherché dans ces scores pour déterminer l'éventuelle apparition d'un mot clé.

La probabilité avant-arrière « forward-bacward » mesure la probabilité qu'un mot clé se termine à un instant t, étant donnée l'ensemble des observations et le modèle du mot. Cette estimation est calculée, une deuxième fois, par l'algorithme de Baum-Welch, cette fois si en utilisant une recherche en arrière. Pendant l'étape « forward », les scores $\alpha(s,t)$ ont été calculés. De manière similaire, une fois la séquence est achevée, le calcul des $\beta(s,t)$ se fait en sens inverse. Les scores avant et arrière sont alors combinés pour donner le score total $S_{fb}(w,t)$ du mot, qui n'est autre que la probabilité que le mot clé w se termine à l'instant t, étant données toutes les observations :

$$S_{fb}(w,t) = \frac{\alpha(e_w,t)\beta(e_w,t)}{\sum_s \alpha(s,t)\beta(s,t)} \qquad \textbf{(3.5)}$$

Ces scores combinent implicitement les scores acoustiques et lexicaux.

Jouvet [Jouvet et Monné, 1999], combine deux approches pour la reconnaissance des épellations prononcées à travers une ligne téléphonique. La première approche est basée sur un algorithme avant-arrière « forward-bacward » et la deuxième utilise une procédure d'extraction à base d'un HMM discret. Dans cette dernière approche, tout d'abord un décodage du signal de parole est réalisé afin de fournir la séquence de lettres contenant les erreurs de substitution, d'insertion et de suppression. Ensuite, sachant la séquence de lettres reconnue, la procédure d'extraction recherche le nom le plus probable dans le lexique. Diverse procédures d'extraction ont été proposées dans la littérature, la meilleure performance a été obtenue en utilisant un modèle de Markov discret dans lequel les lettres reconnues représentent ses sorties. Chaque lettre est modélisée par un modèle HMM à deux états et trois transitions. De même, les erreurs d'insertion, de substitution et de suppression sont modélisées par des fonctions de densités de probabilités. Ce formalisme permet d'utiliser des procédures d'apprentissage de HMM pour estimer les valeurs optimales des paramètres de chaque modèle. Après le processus d'apprentissage, ces modèles représentent les erreurs de reconnaissance observées au niveau du résultat obtenu.

3.2.3 Seuil sur les scores de reconnaissance

Dans le domaine de détection de mots clés, on trouve des travaux qui se basent sur un seuil et dans lesquels le rejet ou l'acceptation se fait en comparant le score d'un mot à ce seuil. On peut citer 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.

On trouve aussi des approches basées sur l'estimation de la probabilité a posteriori du phonème. Ils ont utilisés le logarithme de ces probabilités a posteriori sur un intervalle d'hypothèse de phonèmes afin de calculer une mesure de confiance au niveau du phonème. Une mesure de confiance au niveau de mot est créée en utilisant les mesures des phonèmes qu'ils constituent.

D'autres travaux proposent d'effectuer le rejet des mots hors vocabulaire (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 [Benayed, 2003].

Moreau [Moreau et al, 2000] 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.

Soit w, le résultat du décodage du signal d'entrée X, le rapport de vraisemblance s'écrit sous la forme suivante :

$$LR(X/w) = \frac{P(X \mid w \quad correct)}{P(X \mid w \quad incorrect)} \qquad \textbf{(3.6)}$$

En fixant un seuil w_0, nous avons les décisions suivantes :

$$LR(X/w) \geq w_0 \Rightarrow w \quad est \quad accepté$$
$$LR(X/w) < w_0 \Rightarrow w \quad est \quad rejeté$$

Pour calculer ce rapport de vraisemblance, il a estimé que ce dernier sera la combinaison des rapports de vraisemblances calculés au niveau des trames acoustiques. Pour cela, il a effectué l'alignement de l'entrée X sur le modèle de Markov associé à l'hypothèse w, afin d'associer à chaque trame x_i du signal $X = (x_1, x_2, ..., x_T)$ un état acoustique q_i de la séquence d'états $Q = (q_1, q_2, ..., q_t)$.

On définit le rapport de vraisemblance au niveau de la trame x_i par l'équation suivante :

$$LR(x_i/q_i) = \frac{P(x_i/Mq_i)}{P(x_i/M\overline{q}_i)} \qquad \textbf{(3.7)}$$

où $P(x_i/Mq_i)$ et $P(x_i/M\overline{q}_i)$ sont respectivement les scores de x_i sachant le modèle des événements corrects associés à l'état q_i et le modèle des événements incorrects associés au même état q_i.

Ainsi le rapport de vraisemblance s'écrit de la façon suivante :

$$LR(X/w) = \frac{\prod_{i=1}^{T} P(x_i / Mq_i)}{\prod_{i=1}^{T} P(x_i / M\overline{q}_i)} \qquad \textbf{(3.8)}$$

La mesure de confiance $CM(w)$ de l'hypothèse w sera le $\log$ du rapport de vraisemblance global normalisé par le nombre de trames acoustiques T :

$$CM(w) = \frac{1}{T}\log[LR(X/w)] \qquad \textbf{(3.9)}$$

Afin de calculer cette mesure de confiance, il est nécessaire de faire l'apprentissage du modèle Mq_i et celui du modèle des événements incorrects (ou anti-modèle) $M\overline{q}_i$ pour chaque état q_i. La principale difficulté réside dans la détermination de l'anti-modèle qui doit modéliser les différents types d'événements incorrects. Pour cela, trois densités ont été apprises pour chaque état q_i : $\overline{q}_{i(sub)}$ pour les erreurs de substitution, $\overline{q}_{i(hv)}$ pour les erreurs de fausses acceptations sur les mots hors vocabulaire et $\overline{q}_{i(br)}$ pour les erreurs de fausses acceptations sur les bruits. Ces densités sont estimées à partir des trames acoustiques associées à l'état q_i au sein d'alignements incorrects (fausses acceptations sur les mots hors vocabulaire ou sur bruit, les substitutions). Plusieurs combinaisons de ces trois densités ont été testées. Parmi eux, la moyenne arithmétique des vraisemblances :

$$P(X/M\overline{q}_i) = \frac{1}{3}\left[\overline{q}_{i(sub)}(x) + \overline{q}_{i(hv)}(x) + \overline{q}_{i(br)}(x)\right] \qquad \textbf{(3.10)}$$

Le modèle Mq_i est représenté par la densité de probabilité apprise de la même manière que les densités précédentes, à partir des trames associées à l'état q_i dans un corpus d'alignements corrects.

Toutes les densités sont estimées dans le même espace de paramètres acoustiques que celui de la modélisation markovienne, c'est-à-dire les coefficients cepstraux et leurs dérivées premières et secondes. Ceci représente un avantage de cette méthode puisqu'elle ne nécessite pas l'extraction d'autres paramètres de post-traitement supplémentaire.

3.2.4 Connaissances acoustiques et linguistiques

En plus de l'utilisation des méthodes basées uniquement sur des connaissances acoustiques, nous trouvons des travaux qui 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 insuffisantes. Rose [Rose et al,

1998] propose une nouvelle méthode qui combine les deux types de connaissances : linguistiques et acoustiques. Dans cette approche, on intègre la notion de mesure de confiance acoustique dans l'automate stochastique utilisé pour décrire le modèle de langage n-gram du système de reconnaissance. 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 présentation 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. Par exemple, si on considère un modèle de langage bi-gram où la probabilité d'un mot w_i sachant son histoire h est approximée à $P(w_i/w_{i-1})$, alors ce modèle sera étendu et la même probabilité sera représentée par $P(w_i/w_{i-1},c_{i-1})$ où c_i est une variable discrète, $c_i \in \{0,1\}$ qui exprime la confiance acoustique de l'histoire du mot w_i. Si $c_i = 0$, la confiance acoustique accordée est faible sinon cette confiance est forte.

Hernandez [Hernandez et al, 2000], 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 des trois mesures de confiance acoustiques à la sortie des résultats de reconnaissance. Ensuite, ces mesures compilées à l'aide d'un système d'inférence flou qui prend en entrée ces trois types de mesures de confiance et fournie 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 $S(w_i)$ comme c'est indiqué par l'équation suivante :

$$C_f(w_i) = C_i(w_i) \times S(w_i) \qquad \textbf{(3.11)}$$

$C_f(w_i)$ est la nouvelle mesure de confiance ré estimée et $C_i(w_i)$ est la mesure de confiance acoustique déjà calculée. $S(w_i)$ est un coefficient de proportionnalité au niveau duquel on fait intervenir l'information contextuelle puisqu'il dépend des probabilité contextuelles $P(w_i/w_{i-1})$ et $P(w_{i+1}/w_i)$. Ce coefficient est calculé à l'aide d'un autre moteur flou qui prend en entrée ces deux probabilités ainsi que les mesures de confiance acoustiques des w_{i-1} et w_{i+1}. Cette méthode nous fournit alors une nouvelle mesure de confiance intégrant deux types de connaissances : linguistiques et acoustiques.

3.2.5 Algorithmes basés sur les transformations linéaires

On trouve aussi des études qui utilisent des algorithmes exploitant les transformations linéaires pour l'extraction du vecteur de caractéristiques et pour la détection de mots clés. Parmi eux, on peut cité :

Kampari [Kampari et al, 2000], présente une technique de calcul de confiance au niveau du mot fondée sur une combinaison de plusieurs caractéristiques, elle-même 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 [Vergvri, 20001 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.

Maison [Maison et Gopinath, 2001] montre que la normalisation du maximum d'entropie convient très bien au rejet des mots inutiles. Il l'utilise comme fonction objective pour choisir les paramètres des mesures de confiance basées sur le graphe des mots et pour optimiser les combinaisons des différentes mesures de confiance. Il a montré que la combinaison linéaire de la technique basée sur le graphe des mots et le score acoustique donne de bonnes performances.

Hazen [Hazen et Bazzi, 2001] combine deux méthodes, la première basée sur un modèle explicite pour la détection des mots Hors Vocabulaire (HV) et la deuxième identifie les mots insérés en se basant sur une mesure de confiance extraite par les systèmes de reconnaissance. Une projection discriminante linéaire simple du vecteur des caractéristiques est employée afin d'extraire une seule mesure de confiance pour chaque mot. L'apprentissage du vecteur de projection est effectué en utilisant l'erreur de classification minimale.

Zhang [Zhang et al., 2001] fait une étude sur l'estimation d'une mesure de confiance pour une application de reconnaissance de la parole continue à grand vocabulaire indépendamment du locuteur. Il a proposé 10 paramètres générés à partir de différents niveaux du processus de reconnaissance. Un algorithme d'analyse de fiabilité des paramètres a été développé afin d'extraire la mesure de confiance finale.

Moreno [Moreno et al., 2001] présente une application de l'algorithme de classification « Boosting » pour le calcul des mesures de confiance. Il dérive un vecteur de caractéristiques à partir du treillis de reconnaissance de la parole et l'introduit dans le classifieur. Ce classifieur combine des centaines de systèmes d'apprentissage simples et dérive des règles de classification pour réduire le taux d'erreur de confiance.

Palmer [Palmer et Ostendorf, 2001] a étudié trois méthodes différentes pour améliorer les scores de confiance : arbre de décision, modèle linéaire généralisé et interpolation linéaire utilisée pour les sorties de la première et de la deuxième méthode.

Charlet [Charlet et al., 2001], présente une technique de combinaison de mesures de confiance basée sur la fonction de régression logistique. Les mesures de confiance utilisées sont calculées au niveau de chaque segment de parole composant une hypothèse de reconnaissance w (un mot ou une séquence de mots). Elles décrivent un ensemble de caractéristiques phonétiques comme voisé\ non voisé, voyelle\ consonne etc.

La technique de régression logistique utilisée permet de fusionner des mesures de confiance pour donner à la fin une réponse sous forme d'une probabilité. Elle est basée sur l'hypothèse que le rapport logarithmique de vraisemblance d'un ensemble de mesures de confiance cm_i avec $0 < i < N+1$ peut être estimé par une combinaison linéaire comme suit :

$$\log \frac{P(cm_1,...,cm_N/coorecte)}{P(cm_1,...,cm_N/incorrecte)} = b_0 + b_1 cm_1 + ... + b_N cm_N \qquad \textbf{(3.12)}$$

Par conséquent :

$$P(correcte/cm_1,...,cm_N) = \frac{1}{1+\exp^{-(a_0+a_A cm_1+...+a_N cm_N)}}$$

$$= \frac{1}{1+\exp^{(A*CM)}} \qquad \textbf{(3.13)}$$

Où $A = \{a_{\grave{a}}, a_1,..., a_N\}$ est un vecteur de coefficients et $CM = \{1, cm_{\grave{a}}, cm_1,..., cm_N\}$ le vecteur des mesures de confiance.

Ainsi la régression logistique permet l'estimation de la probabilité a posteriori que la réponse du classifieur soit correcte en donnant toutes les mesures de confiance.

Les coefficients du vecteur A de la fonction de régression logistique sont estimés afin de maximiser l'ensemble des vraisemblances de développement.

$$L = \sum_{i=1..K} c_i \log(P_i) + (1-c_i)\log(1-P_i) \qquad \textbf{(3.14)}$$

où pour chaque exemple i :

- $c_i = 1$ si le test d'indice i est correct, sinon $c_i = 0$.
- P_i est la probabilité a posteriori, que le test i soit correct, en donnant le vecteur des mesures de confiance CM_i, associé au test i.

En trouve aussi dans la littérature d'autres méthodes et techniques utilisées comme les réseaux de neurones et les méthodes d'adaptation qui ont donné des résultats meilleurs dans la reconnaissance que la tâche de détection de mots clés. Pour notre étude, on juge que l'hybridation des Algorithme de Viterbi et de Baum-Welch avec un seuil sur le score de reconnaissance nous convient bien.

3.3 Taxonomie des systèmes de détection de mots clés

Dans le domaine de la reconnaissance automatique de la parole, il est très difficile de faire une comparaison entre deux systèmes de reconnaissance mais nous pouvons les classifier suivant des critères, comme par exemple la rapidité de reconnaissance, la taille de vocabulaire, la grammaire utilisée, la robustesse au bruit, l'indépendance vis-à-vis des locuteurs etc. Néanmoins, le critère définitif de comparaison entre deux systèmes reste le taux de reconnaissance, mais cette comparaison n'est valide que si les deux systèmes sont évalués au moins dans les mêmes conditions d'application.

3.3.1 Par rapport au taux d'erreurs

La reconnaissance automatique de la parole implique la bonne gestion des erreurs de reconnaissance. Ces erreurs d'origines diverses doivent être estimées d'une façon identique pour qu'on puisse faire la comparaison entre les différents types de systèmes de reconnaissance. Par exemple dans une application de reconnaissance de la parole continue, nous pouvons trouver trois types d'erreurs :

- *Erreur d'insertion* : le système reconnaît un mot alors qu'il n'est pas prononcé.
- *Erreur de substitution* : le système confond deux mots différents c'est-à-dire pour un mot prononcé le système reconnaît un autre mot.
- *Erreur d'omission* : le système ne reconnaît pas un mot prononcé.

L'existence de ces trois types d'erreurs va compliquer le problème d'évaluation surtout qu'une erreur de substitution peut être considérée comme étant la combinaison simultanée d'une erreur d'omission et d'une erreur d'insertion ce qui revient alors à considérer cette erreur comme double. [Benayed, 2003].

Dans la littérature, plusieurs définitions ont été introduites afin de calculer et de comparer les performances des systèmes de reconnaissance. Parmi ces définitions, nous présentons le Taux de Mots Corrects (TMC) (Word Correct Rate) qui est la mesure la plus intuitive pour mesurer les performances d'un système de reconnaissance de la parole. Elle est basée sur le nombre de mots correctement reconnus et le nombre de mots total à reconnaître. La formule de TMC s'écrit sous la forme suivante :

$$TMC = 100 \times \frac{Nombre \quad de \quad mots \quad correctement \quad reconnus}{Nombre \quad total \quad des \quad mots \quad à \quad reconnaitre} \qquad \textbf{(3.15)}$$

Le problème majeur de cette mesure c'est qu'elle ne tient pas compte de toutes les erreurs commises, car elle ne dépend pas du nombre d'insertions. C'est pour cette raison que dans la majorité des évaluations une autre mesure est utilisé, le Taux d'Erreur (TE) calculé au niveau mot, appelé aussi WER (Word Error Rate). Elle est calculée de la façon suivante :

$$TE = 100 \times \frac{Nombre \quad d'insertions \quad + \quad Nombre \quad de \quad substitutions \quad + \quad Nombre \quad d'omissions}{Nombre \quad total \quad des \quad mots \quad à \quad reconnaitre}$$

La mesure du taux d'erreur TE est plus fidèle que celle du TMC, puisqu'elle permet la mise en évidence de toutes les erreurs du système de reconnaissance. Malgré que cette mesure soit plus juste, elle possède aussi un revers. En effet, dans des tâches telle que la détection de mots clés dans un flux de parole continue, il est évident que nous avons beaucoup d'erreurs d'insertion et dans les cas extrêmes nous trouvons plus d'insertions que de mots à reconnaître. Dans ces conditions, le taux d'erreur (TE) sera donc supérieur à 1 et le taux de reconnaissance (TMC) sera inférieur à zéro d'où la nécessité de trouver un autre système d'évaluation en plus de cette simple mesure.

3.3.2 Courbe caractéristique d'opération du récepteur ROC (Receiver Operating Characteristic) :

Dans une tâche de détection de mots clés dans un flux de parole, nous disposons d'un ensemble de mots prononcés et dans lequel nous nous intéressons uniquement à la détection d'un sous- ensemble plus restreint contenant des mots clés fixés a priori par l'utilisateur et qui dépendent de l'application considérée. Pour l'évaluation d'un tel système, les définitions précédentes du paragraphe 3.3.1 ne sont plus valides car la nature des erreurs a changé. Nous distinguons alors deux nouveaux types d'erreurs possibles qui sont :

- *Faux Rejet (FR):* se produit quand le système de reconnaissance ne détecte pas un mot clé alors qu'il est prononcé.
- *Fausse Alarme (FA):* se produit quand le système de reconnaissance détecte un mot clé alors qu'il n'est pas prononcé.

Nous définissons alors deux mesures pour ces deux types d'erreurs qui sont le Taux de Faux Rejet (TFR) et le Taux de Fausse Acceptation (TFA) données respectivement par les deux formules suivantes :

$$TFR = \frac{Nombre \quad de \quad faux \quad rejets}{Nombre \quad total \quad de \quad mots \quad clés \quad à \quad détecter} \qquad \textbf{(3.17)}$$

$$TFA = \frac{Nombre \quad de \quad fausses \quad acceptations}{Nombre \quad total \quad de \quad mots \quad non \quad clès} \qquad \textbf{(3.18)}$$

Pour un système de détection, chaque couple de valeurs des taux TFR et TFA représente un point de fonctionnement. Afin de représenter toutes les possibilités du système, il est préférable de considérer plusieurs points de fonctionnement. Il s'agit de représenter un taux en fonction de l'autre obtenant ainsi une courbe de performance pour le système. La courbe obtenue est une courbe ROC (Receiver Operating Characteristic) appelée aussi courbe caractéristique d'opération du récepteur.

Une courbe ROC est un graphe qui peut être utilisé aussi pour représenter le Taux de Détection (TD) de mots clés (avec TD = TMC) en fonction du taux de fausses acceptations (TFA) ou bien pour représenter le taux de détection de mots non-clés, c'est-à-dire Taux de Rejet Correct (TRC) en fonction du taux de faux rejet (TFR). Nous remarquons que les informations contenues dans ces deux types de courbes sont équivalentes puisque nous avons :

$$TD = 1 - TFR \qquad \textbf{(3.19)}$$

$$TRC = 1 - TFA \qquad \textbf{(3.20)}$$

La figure 3.5 illustre trois courbes ROC possibles de type TD en fonction de TFA. La première correspond à un test idéal, avec un taux de détection TD égal à 1 (TFR 0) pour tous les TFA possibles (le meilleur cas correspond à un taux de fausse acceptation TFA égal à zéro). La deuxième représente un test d'un cas limite avec un taux de détection TD égal au taux de fausse acceptation TFA et elle est représentée par la droite TFA = TD. La troisième courbe est une courbe ROC typique qui se trouve entre ces deux courbes extrêmes.

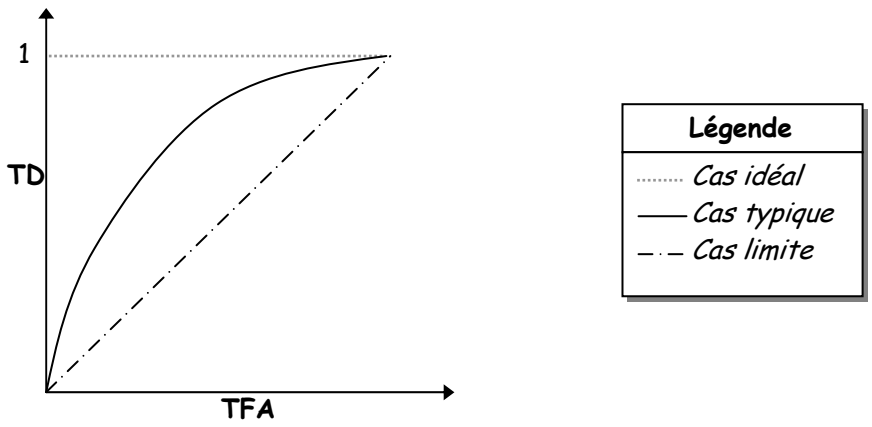


Figure 3.5 : Schéma de courbes ROC pour un test idéal, typique et estimé

Une autre disposition possible des courbes ROC consiste à représenter la probabilité d'erreur de faux rejet TFR en fonction du taux de fausses acceptations TFA. La figure 3.6 fournit une illustration schématique de deux courbes ROC possibles correspondant à deux tests : un cas limite et un cas typique. Le cas limite est représenté par une droite d'équation $TFA + TFR = 1$ et donc $TFR = 1 - TFA$. Nous savons bien que pour un système parfait, nous avons $TFA + TFR = 0$, ce qui correspond au point d'origine $TFA = TFR = 0$, un cas typique est représenté donc par une courbe qui se trouve entre ces deux extrêmes.

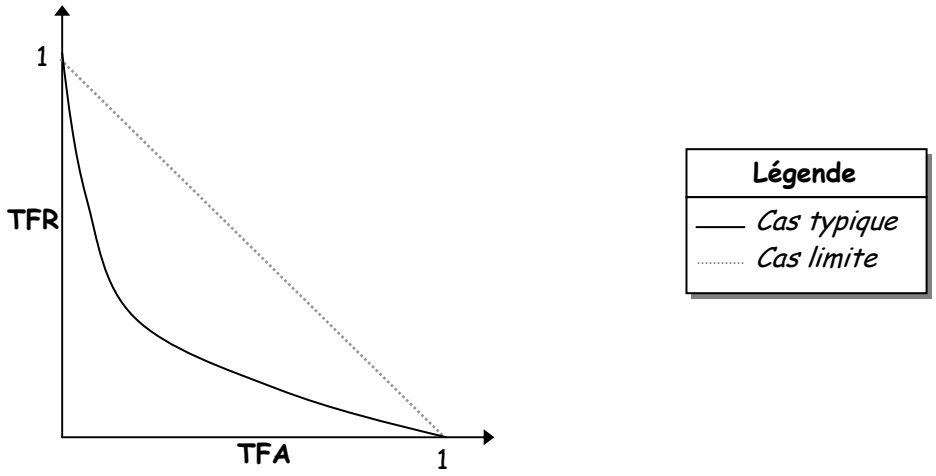


Figure 3.6 : Schéma de courbes ROC du Probabilité TFA en fonction de probabilité TFR

Soit T un seuil de classification, ce seuil est utilisé pour accepter ou rejeter la réponse du système de détection. Autrement dit, si le score de la réponse du système de détection est supérieur à T alors nous acceptons cette réponse, sinon elle sera rejetée. La variation du seuil T le long de l'axe des abscisses donne différentes valeurs de TFR et de TFA. Quand elles sont tracées, elles donnent le graphe théorique illustré par la figure 3.7.

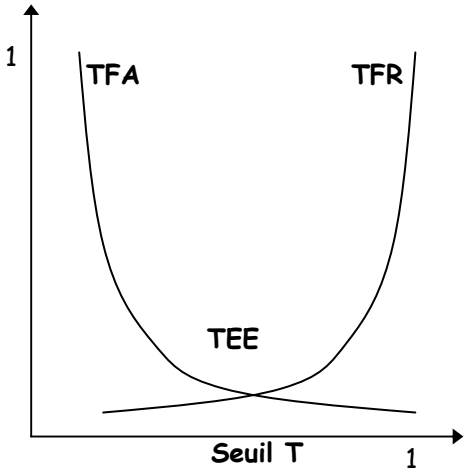


Figure 3.7 : TFA et TFR en fonction du seuil T

Quand on fait augmenter la valeur du seuil T alors le TFA diminue et le TFR augmente. Un grand TFA signifie qu'un mot non-clé a une grande tendance d'être accepté comme étant un mot clé, alors qu'un grand TFR signifie qu'un mot clé a une grande tendance d'être rejeté. Ainsi un grand TFA rend le système moins efficace, puisque par exemple, dans une application d'accès par mot de passe, n'importe quel utilisateur peut accéder au système. De même un grand TFR va compliquer l'utilisation du système.

Il est impossible de minimiser les deux taux TFR et TFA en même temps comme le montre la figure 3.8. Cependant un compromis peut être atteint quand TFR = TFA, ce point s'appelle le Taux d'Erreur Égale (TEE). La valeur du TEE peut être utilisée pour comparer les résultats de deux systèmes de détection. Le meilleur système c'est celui qui a le plus petit TEE. En effet, si la valeur du TEE est faible alors les valeurs TFA et TFR le sont aussi et donc le système commet peu de fautes. Dans la figure 3.8, nous remarquons que le TEE_1 et meilleur que le TEE_2 et donc la courbe ROC_1 montre une meilleure qualité du système que la courbe ROC_2. Plus la courbe est proche de l'origine plus le système est performant.

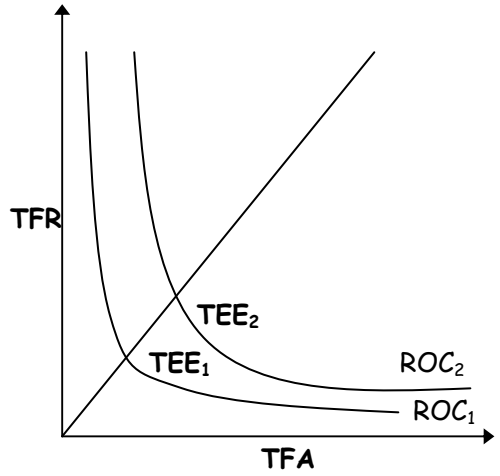


Figure 3.8 : TFR en fonction TFA

Un autre type de courbes ROC bien connu dans le domaine de la détection de mots clés consiste à représenter le taux de détection (TD) en fonction du nombre de Fausses Acceptations par Mot Clé et par Heure (FA/MC/H). Cette disposition de courbes ROC prend mieux en compte la notion du temps puisqu'au niveau de l'axe des abscisses le nombre de fausses acceptations est normalisé par le nombre d'heures. Ce type de courbes est considéré comme étant la plus efficace pour la comparaison de différents systèmes de détection. Dans la figure 3.9, nous illustrons un exemple de deux courbes ROC de ce type où la courbe ROC_1 montre une meilleure performance du système que la courbe ROC_2

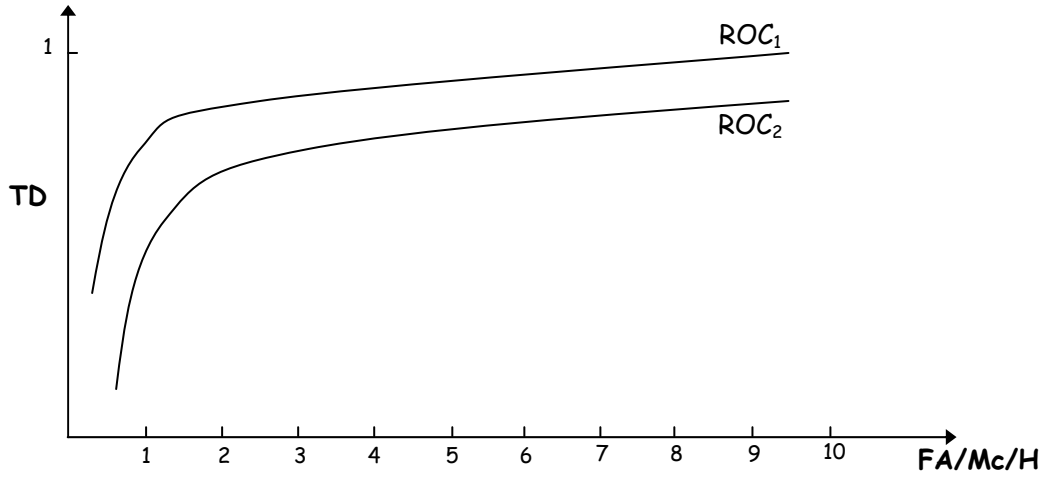


Figure 3.9 : Courbe ROC, Indicateur FOM

Pour obtenir une valeur unique décrivant cette courbe, nous utilisons la moyenne des probabilités de détection pour un taux de fausses acceptations par mot clé et par heure variant entre 0 et 10. Cette valeur est appelée Valeur de mérite ou FOM pour « Figure of Merit » et elle est utilisée pour mesurer les performances d'un système de détection de mots clés pour de faibles taux de fausses acceptations (FA/Mc/H entre 0 et 10). La formule du FOM préconisée par le NIST (National Institute of Standards and Technologv, USA) est donnée par :

$$FOM = \frac{\left(p_1 + p_2 + ... + p_N + ap_{N+1}\right)}{10 \times T} \qquad \textbf{(3.21)}$$

Où

p_i est le pourcentage de détections correctes avant la i[ème] fausse acceptation.

T est la durée du signal de parole.

N est le premier entier $\geq 10T - 1/2$.

$a = 10T - N$ (Facteur d'interpolation à 10 fausses acceptations par heure).

Dans le cas où notre base de test contient exactement une heure de parole, la formule du FOM se simplifie à :

$$FOM = \frac{\left(p_1 + p_2 + ... + p_{10}\right)}{10} \qquad \textbf{(3.22)}$$

Les probabilités de détection p_i sont calculées de deux manières différentes selon la valeur de la durée T de la base de test. En effet, si $T \leq 1$ alors la probabilité p_i est placée sur la courbe

ROC correspondante au niveau de l'abscisse $x = (i - 1/2)$. Sinon, cette valeur est reportée au point d'abscisse $x = (i - 1/2)/T$ fausses acceptations par heure et par mot clé.

3.3.3 Par rappel et précision :

Parmi les objectifs d'un système de détection figure la diminution du nombre de fausses acceptations et de faux rejets. Afin de mesurer la qualité d'un système par rapport à cet objectif on peut utiliser deux critères bien connus qui se nomment « précision » et « rappel ». Ces deux mesures sont fréquemment utilisées dans plusieurs domaines faisant appel à des techniques proches de la statistique et surtout dans ceux qui utilisent des filtres binaires (soit un objet est sélectionné par le filtre, soit il ne l'est pas). En particulier, on les rencontre en permanence en traitement automatique de la langue, aussi bien en analyse ou en compréhension de texte qu'en fouille de documents, etc. Dans notre cas, ces deux mesures vont nous servir pour évaluer les performances des systèmes de détection.

- La précision d'un ensemble de mots clés détectés par le système est la proportion des mots clés corrects dans cet ensemble. Elle est définie par :

$$\text{Précision} = \frac{Nombre\ \ de\ \ mots\ \ clés\ \ correctement\ \ détectés}{Nombre\ \ total\ \ des\ \ mots\ \ clés\ \ détectés} \qquad \textbf{(3.23)}$$

- Le rappel d'un ensemble de mots clés détectés rend compte de la quantité de bonnes réponses par rapport au nombre de mots clés réel. Autrement dit, le rappel est le taux de mots clés corrects détectés par rapport au nombre des mots clés à détecter. Il est défini par :

$$Rappel = \frac{Nombre\ \ de\ \ mots\ \ clés\ \ correctement\ \ détectés}{Nombre\ \ de\ \ mots\ \ clès\ \ à\ \ détecter} \qquad \textbf{(3.24)}$$

On peut également définir les notions de « bruit » et de « silence » qui sont respectivement les notions complémentaires de la « précision » et du « rappel » :

$$bruit = 1 - \text{Précision} \qquad \text{et} \qquad silence = 1 - Rappel$$

Idéalement, on voudrait qu'un système de détection donne de bons taux de précision et de rappel en même temps. Cependant, un système qui donne 100% de rappel et 100% de précision est non atteignable en pratique car cela signifie que ce système détecte tous les mots clés corrects et rien que les mots clés corrects chose parfaite mais très difficilement réalisable. En effet, les deux mesures de rappel et de précision sont intimement reliées et il n'est pas possible d'augmenter l'une sans diminuer l'autre. Par exemple, en tentant d'améliorer la précision, c'est-à-dire d'augmenter la

performance du système à ne détecter que les mots clés corrects, on réduit nécessairement sa fenêtre de possibilités et dans ce cas on réduit aussi sa capacité de rappel car on va provoquer beaucoup de silence. De même on peut facilement avoir un très bon taux de rappel en relâchant les contraintes de détection et on aura dans ce cas un grand nombre de mots clés corrects mais aussi un grand nombre de mots clés non corrects c'est-à-dire beaucoup de bruit et donc un faible taux de précision.

Les mesures de précision et de rappel ne sont pas statiques, c'est-à-dire qu'un système n'a pas qu'une seule valeur de mesure de précision ou de rappel. Le comportement d'un système peut varier en faveur de l'une au détriment de l'autre et vice-versa. Ainsi pour un système donné, on peut tracer l'évolution de la précision en fonction du rappel par une courbe rappel/précision.

L'allure générale d'une courbe « rappel/précision » (pour un cas typique) est donnée dans la figure 3.10. Cette courbe montre qu'il est toujours possible d'obtenir une précision élevée au prix d'un rappel faible ou un rappel élevé au prix d'une précision faible. Dans la pratique, on essaye de trouver un compromis entre ces deux exigences. La courbe optimale, représentée par une droite horizontale à un niveau de précision égale à 1, correspond à un cas idéal où le système arrive à détecter tous les mots clés corrects et rien que ceux-ci.

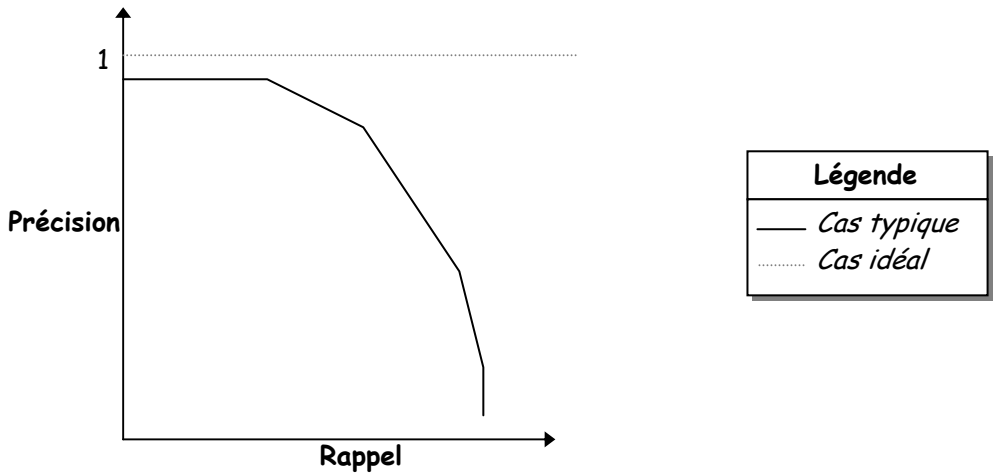


Figure 3.10 : Courbes Rappel/ Précision

Dans la littérature, on parle aussi de courbe rappel précision interpolée où on essaye de calculer la précision pour des valeurs prédéfinies du rappel, de 0% à 100% par pas de 10%. En pratique, les valeurs du rappel peuvent ne pas être atteintes exactement : les valeurs de la précision doivent donc être interpolées. La règle d'interpolation généralement utilisée définit la valeur de la précision pour un niveau de rappel i comme étant la valeur maximale de précision pour tout niveau de rappel supérieur ou égal à i. Cette définition permet d'obtenir une valeur de précision pour un niveau de rappel nul, alors qu'une telle valeur n'existe pas en réalité, et ce en adoptant la valeur de précision la plus importante. Par exemple, s'il existe 200 mots clés, la valeur interpolée de la précision pour un rappel de 10% correspond à la meilleure précision obtenue avec au moins 20 mots clés corrects.

L'avantage de cette interpolation est qu'elle permet de connaître la précision pour des valeurs standardisées. Ainsi, si on étudie les performances du système pour chaque mot séparément, on peut facilement obtenir la courbe moyenne du système en moyennant simplement toutes les précisions obtenues aux différents niveaux du rappel pour tous les mots. Ainsi, les performances d'un système de détection sur un ensemble de mots peuvent être caractérisées par une seule courbe.

3.3.4 Intervalle de confiance :

Dans la pratique, il est fréquent d'introduire la notion d'incertitude des mesures en estimant à chaque fois un intervalle de confiance. Nous cherchons à déterminer l'intervalle de confiance $[a,b]$ centré sur la valeur x qui est une valeur estimée d'un paramètre inconnu θ. Nous supposons que cet intervalle contient la vraie valeur de θ avec une probabilité de $1-\alpha$ fixée a priori.

$$P[a<\theta<b]=1-\alpha \qquad \textbf{(3.25)}$$

Cette probabilité est appelée niveau de confiance de l'estimation, on la désigne par $1-\alpha$. Le α (coefficient de confiance) représente le risque que nous prenons de se tromper en affirmant que θ est bien dans l'intervalle proposé. Une estimation par intervalle de confiance sera d'autant meilleure que l'intervalle sera petit pour un coefficient de confiance grand.

Pour déterminer l'intervalle de confiance, nous avons besoin de connaître outre la taille de l'échantillon, la loi de probabilité du paramètre à estimer. Comme il n'existe pas de résolution générale pour ce dernier problème, différentes solutions ont été proposées. Nous nous intéressons à la solution la plus adoptée qui est l'estimation d'une proportion.

Soit une population dont les individus possèdent un caractère A avec une probabilité p. On cherche à déterminer cette probabilité inconnue en prélevant un échantillon de taille N dans cette population. Nous constatons que n parmi les N individus possèdent le caractère A. La proportion $f_N=\frac{n}{N}$ approxime la vraie valeur de p mais avec quelle confiance?

Soit $f_N=\frac{n}{N}$; f_N une variable aléatoire construite comme étant la somme de N variables aléatoires $O_1,...,O_N$ et de même paramètre p. Il s'agit donc, d'après le théorème central limite, d'une variable aléatoire dont la loi de probabilité tend vers une loi normale de moyenne p et d'écart type $\sqrt{\frac{p(1-p)}{N}}$. Cette approximation est valable uniquement si la taille de l'échantillon est suffisamment grande (en pratique $N>50$).

Nous définissons alors l'intervalle de confiance autour de p de la façon suivante :

$$P\left(\left|f_N - p\right| < t\right) = 1 - \alpha \quad \textbf{(3.26)}$$

$$P\left(f_N - t < p < f_N + t\right) = 1 - \alpha \quad \textbf{(3.27)}$$

Avec α est le risque, f_N est une réalisation d'une variable aléatoire $N\left(p, \sqrt{\frac{p(1-p)}{N}}\right)$.

Par une normalisation et un centrage nous obtenons une nouvelle variable aléatoire U définie par :

$$U = \frac{f_N - p}{\sqrt{\frac{p(1-p)}{N}}} : N(0,1) \quad \textbf{(3.28)}$$

Nous avons

$$P\left(\left|f_N - p\right| < t\right) = 1 - \alpha \quad \textbf{(3.29)}$$

Donc

$$P\left(\left|U\right| < u\right) = 1 - \alpha \quad \textbf{(3.30)}$$

Avec

$$u = \frac{t}{\sqrt{\frac{p(1-p)}{N}}} \quad \textbf{(3.31)}$$

Ainsi l'intervalle de confiance de niveau $1-\alpha$ s'écrit sous la forme suivante

$$P[a < \theta < b] = P\left[f_N - u\sqrt{\frac{p(1-p)}{N}} < p < f_N + u\sqrt{\frac{p(1-p)}{N}}\right] = 1 - \alpha \quad \textbf{(3.32)}$$

La valeur de u sera lue sur une table de loi normale $N(0,1)$

Pour α = 0.05, nous avons u = 1.96. Par conséquent, l'intervalle de confiance de niveau 0.95 d'une proportion p est égale à $\pm 1.96\sqrt{\frac{p(1-p)}{N}}$ où N est la taille de l'échantillon.

Chapitre 4

Conception du Système

4.1 Problématique

La recherche d'information pour les documents texte est performante et aboutie à des résultats encourageants. Toutefois, l'avènement des informations multimédia envahit les banques de données ce qui met la recherche d'information en difficulté, voire même incapable d'accéder aux contenus de ces informations. Cependant le domaine de détection automatique de la parole occupe un grand intérêt dans ces dernières années.

Dans ce contexte, notre contribution est d'accentuer la recherche par l'intégration des systèmes capables de chercher l'information dans les fichiers audio en se basant sur la technique détection des mots clés (KWS) dans un flux continu de parole. Cette technique se limite à la recherche et la détection des segments bien définis dans le signal qui nous offre une amélioration importante dans les calculs car les systèmes classiques de reconnaissance de la parole consomment beaucoup de calculs.

Entre outre, la majorité des recherches effectuées dans le domaine de la détection de mots clés se limitent à un vocabulaire limité. De plus, elles traitent les flux de paroles des mots isolés, et n'oublions pas que la majorité de ces systèmes sont très sensibles aux bruits et des effets extérieurs, ce qui nous conduit à résoudre les problèmes suivants :

Quels sont les paramètres pertinents de signal à utiliser pour maximiser la représentativité du modèle choisi ?

Comment modéliser les mots clés (mot recherchés) afin d'accroître le taux de détection ?

Quelle stratégie à utiliser pour remédier aux problèmes de silence, des mots hors vocabulaire et les hésitations ?

Quelles sont les mesures de confiance à utiliser pour vérifier les performances du système ?

Dans notre première tentative, on va essayer de réaliser un système prototype afin de matérialiser les techniques et les approches existantes dans le domaine. Notre système se charge de détecter un certain nombre de mots clés dans un flux continu de la parole spontanée. Cette idée on peut l'exploiter dans plusieurs domaines d'application et surtout dans des applications qui utilisent la voix comme moyen d'accès aux bases de données, comme l'acquisition du code secret d'un client pour exécuter quelque requête sur les Bases de données via le téléphone comme la présente la figure 4.1. A cet effet, on a travaillé dans un système de détection des chiffres. Et comme il n'existe pas beaucoup des travaux dans la langue arabe, on a choisi de travailler sur la langue arabe malgré l'absence des corpus audio dans cette langue

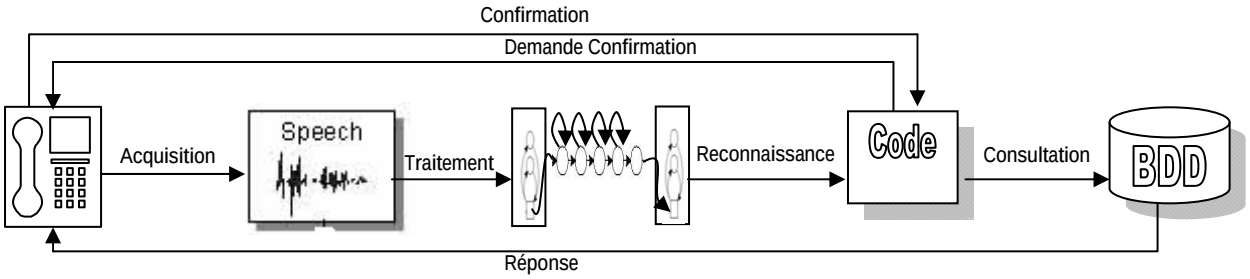


Figure 4.1 : système de vérification du code d'accès au BDD via le téléphone

4.2 Système Proposé

4.2.1 Description du système

Le système proposé est un système de détection de mots clés à base de phonème de la langue arabe. Notre système contient deux modules, Le premier est basé sur un décodage phonétique pour la reconnaissance des phonèmes. Ce processus est nécessaire afin d'inclure le système de mesure de confiance dans notre système de détection de mot clé. Le deuxième, c'est le processus de détection de mot de clés à l'aide de l'algorithme de décodage de Viterbi. Le schéma bloc présenté dans la figure 4.2 illustre cette configuration.

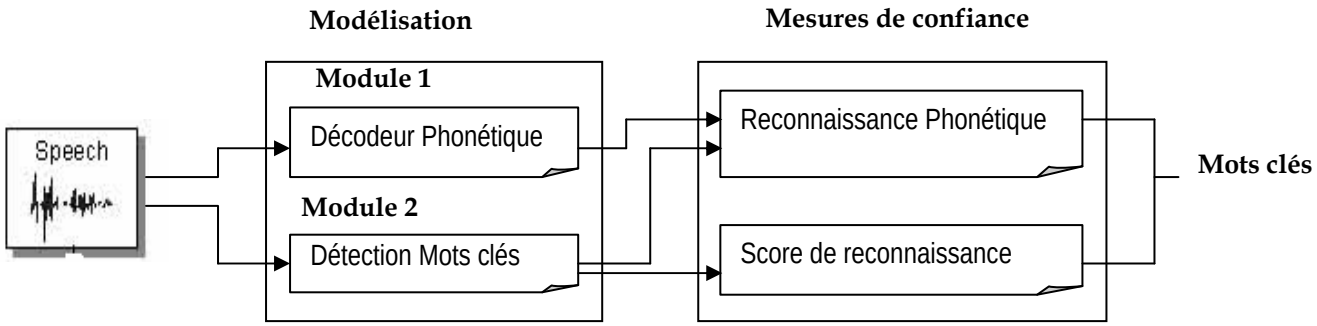


Figure 4.2 : Schéma bloc du système proposé

Dans la phase de modélisation, on a utilisé les modèles poubelle pour absorber tous les mots hors vocabulaire et aussi les hésitations et les faux départs. De plus on a intégré un modèle arrière plan « background model » pour absorber le bruit. La topologie du système est alors un système parallèle avec N modèles mot clé et M modèles poubelle et un modèle background. Aucune

transition n'est permise entre les mots clés et mot poubelles, ce qui permet de trouver plusieurs mots clés ou des instances dans mot clés dans un flux de parole.

Dans nos expériences, on a essayé d'utiliser plusieurs valeurs de M pour chercher le meilleur taux de reconnaissance.

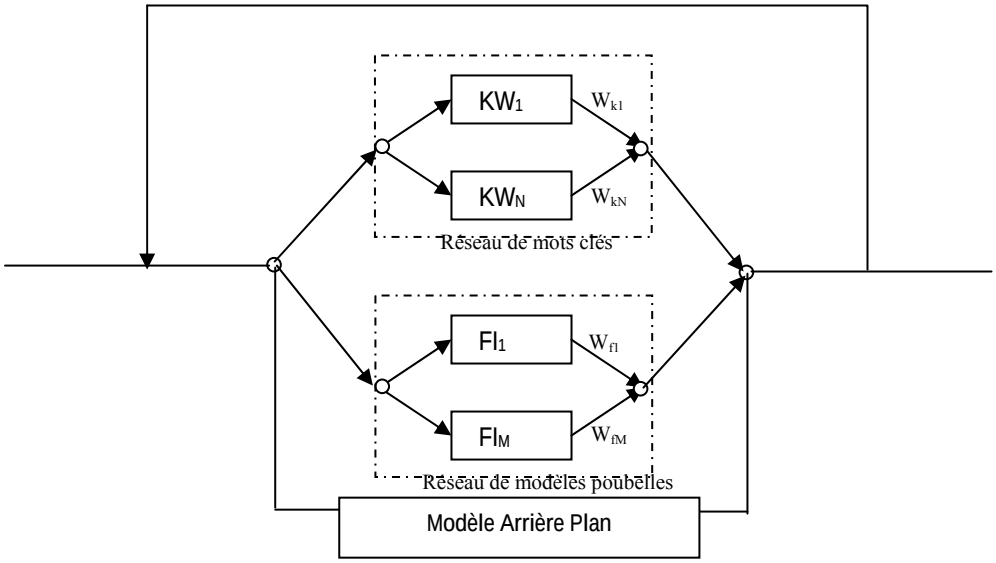


Figure 4.3 : Topologie du réseau proposé

4.2.2 Représentation acoustique

Le corpus utilisé est divisé en deux parties : une partie pour la ré estimation des paramètres des models utilisés et l'autre pour exécuter les test sur les système proposé. Pour chaque entrée, on extrait 12 Coefficient Cepstral (MFCC) et l'énergie avec leurs dérivés premières et secondes ce qui nous donne 39 paramètres représentatifs du signal. Le signal utilisé est échantillonné sur 11 Khz à 16 bits et le format utilisé est « wav ». Ensuite, on applique le fenêtrage de hamming dont la taille de la fenêtre est 25 msec.

Les traitement effectués sur le corpus utilisé est réalisé à l'aide d'un logiciel open source appelé « Wavesurfer ».

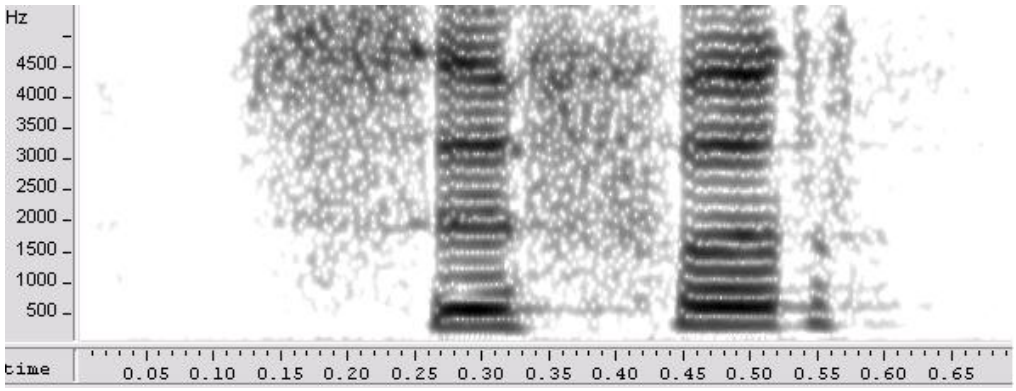


Figure 4.4: Exemple d'un fichier audio du corpus

4.3 Modélisation

Pour la création des modèles poubelles, on a effectué plusieurs configurations et apprentissage de ces modèles en utilisant un corpus audio parlé de la langue arabe. En premier lieu, on a construit les modèles pour tous les phonèmes indépendants du contexte pour ajuster les règles phonétique du parlé arabe. On note toujours, que le parlé arabe change vis- a- avis la région, et pour notre cas c'est le parlé arabe algérien. On à intégré aussi différent type de silence : au début, à la fin, un petit silence et long silence. Cette configuration a été exécutée dans le module 1 de notre système.

En deuxième lieu, on a essayé de travailler dans un ensemble de phonème restreint avec 26 phonème de la langue arabe, et pour les silences on a garder les silences de début et de fin et seulement le silence court dans la parole. Chaque phonème est modéliser par un HMM continu du type Bakis à 3 états émetteurs, et pour les probabilités d'émission on à travaillé avec 15 mélanges de gaussiennes. Et la modélisation des silences du début et fin est réalisé à l'aide d'un HMM continu de 3 états avec 15 mélanges de gaussiennes, tandis que pour le silence du milieu, on a utilisé un HMM continu avec un seul état avec 15 mélanges de gaussiennes.

4.3.1 Mots clés

Les modèles mots clés sont réalisés par la concaténation des phonèmes dans la cadre du parlé arabe dans la région algérienne. De plus, on a intégré des courts silences entre les phonèmes du mot avec la possibilité de passage direct vers le phonème suivant. La figure 4.5 présente un exemple pour la prononciation du mot « 0 » en langue arabe.

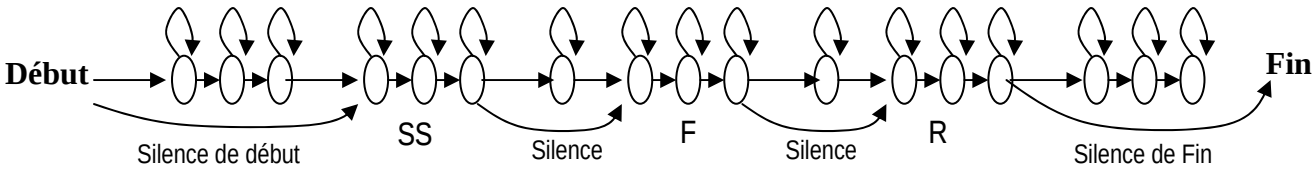


Figure 4.5: Réseau des phonème du Chiffre « صفـر »

4.3.2 Modèle poubelle

4.3.2.1 Première Variante

Plusieurs travaux ont été réalisés en utilisant un modèle poubelle avec apprentissage [Rose et Hofstetter, 1993] [James et Young, 1991], représenté par un HMM à plusieurs états. Yapanel [Yapanel, 1997] a montré qu'un modèle poubelle représenté par un HMM à 6 états est meilleur que celui représenté par un HMM à 9 états et qu'un modèle poubelle modélisé par un HMM à 3 états est meilleur qu'un modèle à 6 états c'est-à-dire que si le modèle HMM est représenté par moins d'états alors il donne de meilleurs résultats.

Les modèles de mélange de gaussiennes (Gaussian Mixture Models : GMM,) sont largement utilisés. Ils peuvent être considérés comme des modèles HMM à un seul état où la fonction de densité est composée d'un mélange de gaussiennes. Une approche basée sur les GMM consiste à produire un modèle sous forme d'une somme pondérée de M gaussiennes. Chaque gaussienne g_i est caractérisée par un poids p_i, un vecteur moyen μ_i de dimension n et une matrice de covariance $\sum_i$ de dimension $(n \times n)$. L'apprentissage des paramètres du modèle $GMM\left(p_i, \mu_i, \sum_i\right)$ pour $i \in \{1,...,M\}$ est réalisé généralement à l'aide de l'algorithme EM (Expectation-Maximisation) qui représente une technique générale pour l'estimation par maximum de vraisemblance. Les modèles à base de mélanges de gaussiennes ont fait leurs preuves dans plusieurs domaines notamment, en identification automatique du locuteur où ils fournissent les meilleurs résultats actuels.

Soient λ un modèle de mélange de gaussiennes et $O = (o_1, o_2, ..., o_T)$ une séquence d'observations relative à un segment de parole. Pendant la phase de reconnaissance du segment de parole, la vraisemblance qu'un vecteur o_T soit produit par le modèle λ est donnée par le mélange de gaussiennes suivant :

$$f(o_t/\lambda) = \sum_{i=1}^{M} p_i f_i(o_t) \qquad \textbf{(4.1)}$$

Où f_i est une gaussienne donnée par la formule suivante :

$$f_i(o_t) = \frac{1}{(2\pi)^{\frac{n}{2}} \left| \sum_i \right|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(o_t - \mu_i)^T \left(\sum \right)^{-1} (o_t - \mu_i) \right] \qquad \textbf{(4.2)}$$

Ainsi, la vraisemblance pour que la totalité du segment de parole O soit produite par le modèle λ s'écrit sous la forme suivante :

$$f(O/\lambda) = \prod_{t=1}^{T} f(o_t/\lambda) \qquad \textbf{(4.3)}$$

Dans la phase de reconnaissance nous disposons des modèles des phonèmes, du modèle de silence (début ou fin) où chacun est modélisé par un HMM à 3 états (gauche-droite) à densités continues et avec 15 gaussiennes par état, et aussi un modèle de court silence modélisé par un HMM à un seul état avec 15 gaussienne et du modèle poubelle modélisé par un mélange de 256 gaussiennes.

La probabilité d'observation du modèle poubelle est faible et ceci est une conséquence naturelle de la manière avec laquelle nous l'avons réalisé. En effet ce modèle ainsi défini, est très général puisqu'il ne représente pas un phonème ou un mot précis. Par conséquent, le système de reconnaissance a une grande tendance à insérer les mots clés, et ceci va faire augmenter le nombre d'insertions.

Pour remédier à cette augmentation du nombre d'insertions, il est nécessaire d'introduire une variable de pénalisation pour le mot en entrée P_{me}. Cette pénalité améliorera le taux de rejet et permettra en la faisant varier, de trouver un compromis entre le nombre de fausses acceptations et le taux de détection.

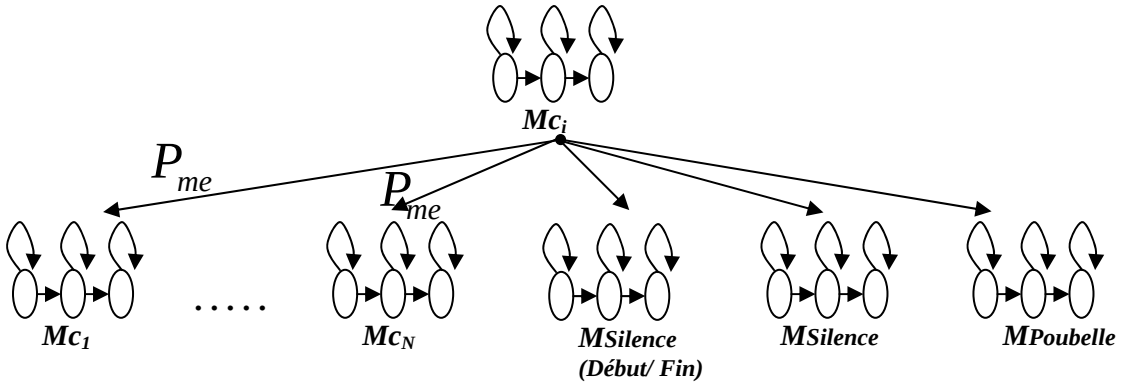


Figure 4.6: Principe de pénalisation des mot clés

Afin d'appliquer cette technique de pénalisation, nous supposons que nous avons reconnu le mot clé Mc_i et que ce mot peut être suivi par l'état poubelle E_{Poub}, ou le modèle silence $M_{Silence(Début/Fin)}$ ou $M_{Silence}$ encore par un mot clé Mc_j avec $j \in \{1,2;...,N\}$ comme montre la figure 4.5. Alors, pour affaiblir le passage vers les mots clés et favoriser le passage vers l'état poubelle, il faut diminuer le logarithme de la probabilité du passage du mot Mc_i au mot Mc_j en utilisant une variable de pénalisation :

$$Log\left[P\left(Mc_j / Mc_i\right)\right] - P_{me} \qquad \textbf{(4.4)}$$

$$P\left(Mc_j / Mc_i\right) = \frac{1}{N_{succ(i)}} \qquad \textbf{(4.5)}$$

où $Mc_j \in \{Mc_1,...,Mc_n\}$ et $N_{succ(i)}$ est le nombre de modèles successeurs au modèle Mc_i, $N_{succ(i)} = n+2$. P_{me} est une variable de pénalisation dont Les différentes valeurs de pénalisation seront choisies en se basant sur une série d'expériences menées sur le base de développement. La variation de cette pénalisation permet de trouver un compromis entre le nombre de fausses acceptations et le taux de détection.

Pendant la phase d'apprentissage, sont appris les modèles des phonèmes, le modèle de silence début ou fin et le modèle silence (silence court entre les mots) et le modèle poubelle GMM. En phase de reconnaissance, nous disposons donc de tous les modèles de mots clés, du modèle silence (début ou fin), du modèle silence, du modèle GMM et de la grammaire. Durant cette phase de reconnaissance, nous utilisons l'algorithme de Viterbi afin d'avoir la meilleure séquence d'états et donc de modèles HMM représentant l'expression à reconnaître.

Nous avons étudié les performances de notre approche en faisant varier la valeur de la pénalisation du mot en entrée pour trouver un compromis entre le nombre de fausses acceptations et le taux de détection ainsi qu'entre le taux de rappel et le taux de précision.

4.3.2.2 Deuxième Variante

Dans un système utilisant une grammaire avec des mots clés et des modèles poubelles, nous remarquons une grande tendance à insérer des mots clés dès que leurs premiers phonèmes sont reconnus, même si les autres phonèmes constituant le mot clé ont de très faibles valeurs de vraisemblance. Nous pouvons expliquer ce phénomène par le fait que le système n'a pas beaucoup de choix : il reconnaît un mot clé et non pas un modèle poubelle puisque il a déjà trouvé des phonèmes de ce mot, et il s'agit du mot le plus probable. Ceci engendre un grand nombre d'insertions et donc une dégradation des résultats. Afin de remédier à ce problème, nous avons proposé de faire la reconnaissance en se basant sur une boucle de phonèmes. Le système fournit dans ce cas la suite des phonèmes les plus probables pour une séquence d'observations donnée. Il ne reste alors qu'à introduire au niveau de la grammaire utilisée les transcriptions phonétiques des différents mots clés. Enfin, nous disposons d'une grammaire composée d'ensemble de mots clés avec leurs transcriptions phonétiques, de l'ensemble des phonèmes indépendants du contexte, du modèle silence du début ou fin et du modèle su silence court (silence entre les mots) comme le présente le figure 4.6.

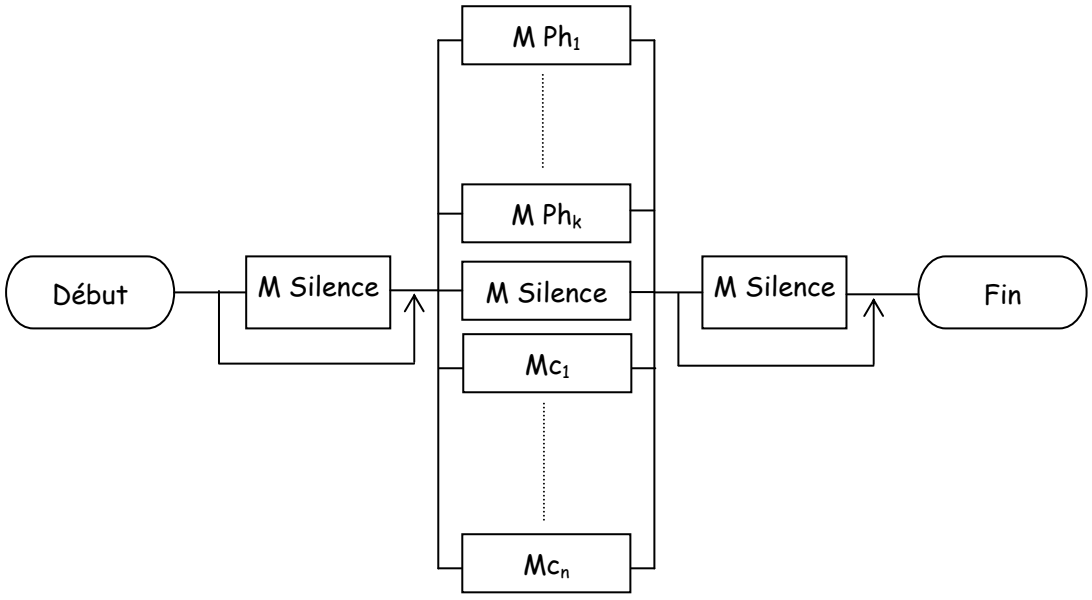


Figure 4.7: grammaire à base de boucle de phonèmes

Ayant la suite des phonèmes reconnus, le système effectue la mise en correspondance entre les transcriptions phonétiques des mots clés fournis par la grammaire et les groupements de phonèmes reconnus. À une mise en correspondance réussie, le système détecte alors le mot clé concerné. Les autres phonèmes (ne correspondant à aucun mot clé) sont laissés en l'état. Ainsi, pour une séquence d'observations correspondant à un message donné, nous obtenons une séquence constituée de phonèmes et de mots clés, comme illustré par la figure 4.7.

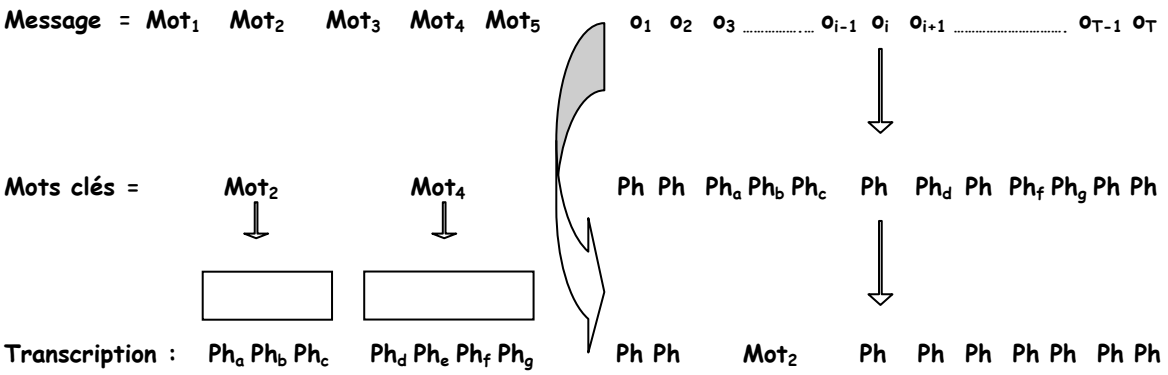


Figure 4.8: Déroulement de la reconnaissance à base de boucle de phonèmes

Avec cette méthode, le système de reconnaissance a tendance de reconnaître plus de phonèmes que de mots clés. En effet, il suffit qu'un seul phonème dans un mot clé soit mal reconnu pour que le mot clé sera déclaré comme une séquence de phonèmes. Ceci est illustré par la figure 4.7 où le phonème Ph_e a été mal reconnu dans le deuxième mot clé dont la transcription phonétique est [Ph_d Ph_e Ph_f Ph_g]. À cause de cette erreur, ce mot clé a été remplacé par une séquence de phonèmes. Avec ce genre d'erreurs, nous avons obtenu en résultat beaucoup plus de phonèmes que de mots clés (un grand nombre d'omissions de mot clés). Pour cette raison, il faut favoriser la reconnaissance des mots clés en augmentant leurs probabilités ou d'une autre façon, en pénalisant les passages aux phonèmes isolés.

Afin de favoriser la reconnaissance des mots clés, nous avons utilisé une approche qui permet l'ajout d'une récompense pour aider le système à détecter les mots clés. Cette récompense a la même valeur pour tous les mots clés (méthode à récompense constante), indépendamment de la longueur du mot clé, c'est-à-dire du nombre de phonèmes qui le constituent.

Supposons que nous avons reconnu le phonème Ph_i qui peut être suivi par un mot clé Mc d'indice k avec $k \in \{1,2,...,n\}$, par un phonème d'indice j tel que $j \in \{1,2,...,d\}$ ou par le modèle silence $M_{silence}$. Alors, le logarithme de la probabilité de passage du phonème Ph_i au mot clé

d'indice k (Mc_k) est incrémenté d'une constante de récompense (positive), comme c'est illustré par l'expression suivante :

$$Log\left[P\left(Mc_k / Ph_i\right)\right] + C \qquad \textbf{(4.6)}$$

$$P\left(Mc_k / Ph_i\right) = \frac{1}{N_{succ(i)}} \qquad \textbf{(4.7)}$$

où $Mc_k \in \{Mc_1,...,Mc_n\}$ un mot clé donné, $Ph_i \in \{Ph_1,...,Ph_d\}$ un phonème reconnu et $N_{succ(i)}$ est le nombre de modèles successeurs au phonème Ph_i, il correspond à la somme du nombre de mots clés n, le nombre de phonèmes d plus le modèle silence, nous avons donc $N_{succ(i)} = n + d + 1$, toutes les transitions entre les modèles sont initialement équiprobables. De cette manière, le système peut passer plus facilement à un mot clé au lieu de passer à un phonème isolé. Ainsi, nous pouvons remédier au problème des phonèmes reconnus avec une faible vraisemblance dans un mot clé donné.

On note aussi qu'il existe d'autre variante de cette méthode, qui peuvent être utilisé dans cette étape comme la méthode de récompense affine et méthode de récompense sigmoïdale.

4.4 Mesure de confiance

Il est souhaitable de calculer une mesure de confiance pour chaque unité reconnue par le système (phonème, mot, etc.), c'est-à-dire d'associer à chacune de ces hypothèses de reconnaissance une mesure qui correspond à sa fiabilité, ce qui permet de rejeter les hypothèses les moins fiables. Dans le cas de la reconnaissance de mots clés, il peut être intéressant d'utiliser un score de confiance pour rejeter les fausses acceptations. Afin de calculer ce score, plusieurs travaux ont utilisé les probabilités a posteriori des mots.

4.4.1 A base des probabilités a posteriori

Pour calculer la mesure de confiance d'un mot clé reconnu, il faut connaître une mesure élémentaire qui est la probabilité a posteriori de chaque phonème. En pratique, pour avoir cette valeur, il faut exécuter la reconnaissance en utilisant une grammaire basée sur les modèles des mots clés, le modèle silence et une boucle de phonèmes indépendants du contexte. Durant cette étape, nous essayons d'avoir le minimum d'omissions puisque dans la prochaine étape, nous aurons besoin de l'ensemble des mots clés reconnus afin de détecter les mots clés qui sont réellement prononcés. Le fait d'avoir des omissions pendant la phase de reconnaissance, nous empêchera de réagir ensuite dans la phase de vérification (ou de décision).

Ensuite, on exécute un alignement de la séquence des phonèmes de la phrase reconnue avec leurs modèles HMM. Ainsi, nous pouvons mémoriser pour chaque état de chaque modèle de phonème, le nombre de trames associés et leurs probabilités d'observations.

Enfin, nous appliquons l'algorithme de Viterbi pour les trois états du modèle HMM du phonème Ph_i étant donné la séquence d'observations O_t correspondante, afin de calculer la probabilité d'observation acoustique locale $P(O_t/Ph_i)$. Ainsi, nous obtenons la probabilité a posteriori du phonème Ph_i : $P(Ph_i/O_t)$ donnée par l'équation suivante :

$$P(Ph_i/O_t)=\frac{P(O_t/Ph_i)P(Ph_i)}{\sum_j P(O_t/Ph_j)P(Ph_j)} \qquad \textbf{(4.8)}$$

où :

$P(O_t/Ph_i)$: est la probabilité d'observation acoustique locale du phonème Ph_i

$P(Ph_i)$: est la probabilité a priori du phonème Ph_i, tous les phonèmes étant équiprobables.

$m=\{Ph_1,...,Ph_N\}$: est la séquence des phonèmes du mot clé prononcé.

$O=\{O_1,...,O_T\}$: est la séquence des observations acoustiques, qui est équivalente à :

$O=\{O_{d[1]},...,O_{f[1]};...;O_{d[i]},...,O_{f[i]};...;O_{d[N]},...,O_{f[N]}\}$, où $d[i]$ et $f[i]$ représentent respectivement, les trames de début et de fin du phonème numéro i, avec $d[1]=1$ et $f[N]=T$.

Après le calcul de la probabilité a posteriori de chaque phonème. Le mot clé est composé d'un ensemble de phonèmes, il faut combiner ces différentes valeurs de probabilités de phonèmes afin de calculer une seule mesure de confiance pour chaque mot clé. Les combinaisons les plus populaires sont bien sûr les moyennes, dont on peut trouver différents types : moyenne arithmétique, géométrique.

Sans conteste, la moyenne arithmétique est la plus courante car elle est la plus intuitive et la plus naturelle. La moyenne géométrique applique le même principe que la moyenne arithmétique, mais en utilisant la notion de multiplication au lieu de l'addition. La moyenne géométrique de deux nombres a et b est la racine carrée de leur produit $m_g=\sqrt{ab}$ c'est une valeur moyenne entre ces deux nombres. En effet, nous pouvons démontrer facilement que si a < b alors a < m_g < b. Cette moyenne n'est pas très courante, mais elle a prouvé son efficacité dans plusieurs problèmes mathématiques. Cependant, nous remarquons, que chaque moyenne essaie d'extraire des informations à partir des données (suite de nombres) d'une façon différente des autres. Pour cela, on essayer de travailler avec ces deux moyennes.

La mesure de confiance est générée pour chaque mot en utilisant les mesures de confiance des phonèmes qui le composent. La mesure de confiance au niveau phonème est calculée comme

étant la probabilité a posteriori causée par l'observation acoustique. Les formules des deux mesures de confiance résultantes sont les suivantes :

$$MC_a = \frac{1}{N}\left[\sum_{i=1}^{N} MC_i\right] \qquad \textbf{(4.9)}$$

$$MC_g = \exp\left[\frac{1}{N}\left[\sum_{i=1}^{N} Log\left(MC_i\right)\right]\right] \qquad \textbf{(4.10)}$$

où :

$MC_i = P\left(Ph_i / O_t\right)$, N est le nombre total de phonèmes dans le mot clé m. MC_a et MC_g sont des mesures de confiance calculées respectivement, comme moyenne arithmétique et géométrique.

Pour chaque mesure de confiance on fixe un seuil γ. Si le score de confiance d'un mot m est inférieur au seuil fixé, ce mot sera rejeté.

$$m = \begin{cases} Accepté \quad Si \quad MC(m) > \gamma \\ \text{Re } jeté \quad Sinon \end{cases}$$

4.4.2 A base de boucle de phonèmes

Dans un système de reconnaissance automatique de la parole utilisant une grammaire composée d'un ensemble de mots clés et de modèles poubelles, le risque d'insertion de mots clés est important. Il suffit qu'un mot commence ou se termine par les mêmes phonèmes qu'un mot clés donné pour qu'il soit remplacé par ce dernier. Ce problème est encore plus sérieux dans le cas où on a des mots qui ressemblent beaucoup à des mots clés. En conclusion, nous pouvons affirmer qu'avec une grammaire composée seulement de mots clés et de modèles poubelles, nous obtenons un grand nombre d'insertions de mots clés.

Effectuer la reconnaissance de la parole en se basant sur une boucle de phonèmes est une solution envisageable. En effet, une telle méthode nous permet de faire la reconnaissance phonème par phonème et de trouver à chaque fois le phonème le plus proche de celui prononcé. Cependant, la question qui reste posée dans ce cas est, de trouver le mot clé prononcé, car nous savons très bien qu'il suffit d'un phonème non correctement reconnu pour que le mot en entier ne soit pas détecté (nous ne pouvons pas trouver une mise en correspondance entre la transcription phonétique du mot et l'ensemble des phonèmes qui lui est associé). Afin de remédier à ce problème, nous proposons de prêter plus d'attention aux phonèmes reconnus avec des vraisemblances importantes et d'en profiter pour décider de l'acceptation ou du rejet du mot en question.

Pour calculer les différentes mesures de confiance, il faut calculer en premier lieu la probabilité d'observation de chaque trame. Afin d'extraire cette information, il est nécessaire de passer par trois étapes.

Premièrement, nous utilisons un système de reconnaissance à base de boucle de phonèmes pour trouver la liste des phonèmes reconnus associés à une séquence d'observations O correspondante à une suite de mots prononcés m . Puis, nous réalisons un alignement de la séquence de phonèmes reconnus sur leurs modèles HMM afin de sauvegarder pour chaque trame sa probabilité d'observation.

Et puis, nous réalisons la reconnaissance en se basant sur une grammaire composée de l'ensemble des mots clés et du modèle de silence. Ceci nous permet d'avoir un grand nombre d'insertions de mots clés dont en aura besoin par la suite. De la même manière, nous réalisons un alignement de la séquence des phonèmes reconnus associés à la même séquence d'observations O sur leurs modèles HMM afin de sauvegarder pour chaque trame sa probabilité d'observation.

Et Finalement, nous associons à chaque trame o_t de O un état q_t , nous obtenons alors, pour le décodage de $O=(o_1,o_2,...,o_T)$, la séquence acoustique des états $Q=(q_1,q_2,...,q_T)$.

Ainsi, on peut accorder à chaque mot clé reconnu deux ensembles de probabilités. Le premier ensemble est composé des probabilités d'observations des états qui constituent ce mot noté $P(o_t/q_i)$. Le deuxième ensemble correspond aux probabilités $P(o_t/q_{bi})$qui sont trouvées en utilisant la méthode à base de boucle de phonèmes.

4.4.2.1 Rapport de Vraisemblance

La décision d'accepter ou de rejeter un mot clé reconnu est prise en se basant sur un test de rapport de vraisemblance. Ce dernier est calculé comme étant un rapport entre le mot reconnu et la séquence de phonèmes qui lui correspond obtenue par la méthode à base de boucle de phonèmes. Il s'obtient en combinant les rapports de vraisemblance calculés pour chacune des trames o_t de l'observation acoustique O et il s'écrit donc, sous la forme suivante :

$$Rv(O/m)=\frac{\prod_{i=1}^{T}P(o_t/q_i)}{\prod_{i=1}^{T}P(o_t/q_{bi})} \qquad \textbf{(4.11)}$$

où : $Rv(O/m)$ est le rapport de vraisemblance, $P(o_t/q_i)$ est la probabilité d'observation correspondante à la trame o_t sachant l'état q_i du mot clé reconnu. $P(o_t/q_{bi})$ est la probabilité d'observation correspondante à la trame o_t sachant l'état cette probabilité est trouvée par la méthode de boucle de phonèmes.

Le score $Rv(O/m)$ sera utilisé ensuite, afin de fournir une décision finale permettant d'accepter ou de rejeter le mot reconnu. Pour ce faire, nous fixons pour chaque score un seuil γ, le mot en question sera donc rejeté si son score est inférieur à ce seuil. Ce qui nous donne :

$$m = \begin{cases} Accepté \quad Si \quad Rv(O/m) > \gamma \\ \text{Re } jeté \quad Sinon \end{cases}$$

4.4.2.2 Distance de Vraisemblance

Dans cette méthode, la mesure de confiance est basée sur une distance de vraisemblance entre le mot reconnu et son image (obtenu par la méthode à base de boucle de phonèmes). Cette distance est calculée en utilisant les probabilités d'observations définies au niveau le plus élémentaire : celui des trames acoustiques. Cette distance s'écrit alors :

$$D(O/m) = \frac{1}{N}\left[\sum_{i=1}^{N}\sum_{j=d[i]}^{f[i]}\left|P(o_t/q_j) - P(o_t/q_{bj})\right|\right] \qquad \textbf{(4.12)}$$

où N est le nombre total des phonèmes du mot clé m et $d[i]$ et $f[i]$ représentent respectivement les trames de début et de fin du phonème Ph_i.

Le score $D(O/m)$ appliqué à un mot hypothèse m est utilisé ensuite pour décider de l'acceptation ou du rejet de ce mot. Il suffit pour cela, de fixer un seuil d'acceptation pour chacune de ces deux mesures. Tout mot ayant un score plus faible que le seuil correspondant sera rejeté. L'utilisation de la distance de vraisemblance comme mesure de confiance permettant d'accepter les mots clés réellement prononcés et de rejeter les mots clés insérés.

Chapitre 5

Expérimentations

5.1 Base de développement

5.1.1 Langage ciblé

La langue arabe est une langue sémitique, elle est parmi les langues les plus anciennes dans le monde. L'arabe classique standard a 34 phonèmes parmi lesquels 6 sont voyelles et 28 sont des consonnes. Les phonèmes arabe se distinguent par la présence de deux classes qui sont appelées pharyngales et emphatiques. Ces deux classes sont caractéristiques des langues sémitiques comme l'hébreu. Les syllabes permises dans la langue arabe sont : CV, CVC et CVCC. Où le V désigne voyelle courte ou longue et le C représente une consonne. La langue arabe comporte cinq types de syllabes classées selon les trais ouvert/fermé et court/long. Une syllabe est dite ouverte (respectivement fermée) si elle se termine par une voyelle (respectivement une consonne). Toutes les syllabes commencent par une consonne suivie d'une voyelle et elles comportent une seule voyelle. La syllabe CV peut se trouver au début, au milieu ou à la fin du mot.

5.1.2 Le Corpus d'analyse

Dans le contexte de la détection des mots clefs dans un flux de parole et dans la mesure où les domaines d'applications visés sont essentiellement des situations de dialogue spontané. La construction des corpus dépend de plusieurs facteurs contrôlables en vue de la modélisation de la durée des différents phonèmes de la langue. Le corpus a été enregistré dans des conditions normales c'est-à-dire sans l'utilisation des cabines insonorisées, par des locuteurs et des locutrices. Le signal est numérisé à la fréquence de 11kHz et à la résolution de 16 bits en format Wave.

Le corpus de mots clés est constitué des six premiers chiffres de l'arabe classique de 0 à 5. Sept locuteurs algériens, 4 males et 3 femelles, sont invités à prononcer les six chiffres vingt fois. Le corpus comprend vingt répétitions par chaque locuteur du même chiffre. Ainsi, le corpus des mots clés est constitué de 840 enregistrements (6 chiffres. 20 répétitions. 7 locuteurs). Pendant l'enregistrement, chaque répétition a été rejouée pour s'assurer que le chiffre entier a été inclus dans le signal enregistré. Ce corpus est utilisé pour construire les modèles de mots clés ; figure 5.1.

On note aussi, que la proposition de ce corpus est due au non disponibilité des corpus audio en langue arabe pour l'exploitation.

Mot clefs	Nombre d'occurrences pour Apprentissage	Nombre d'occurrences pour Apprentissage
صفر	100	30
واحد	100	30
اثنان	100	30
ثلاثة	100	30
أربعة	100	30
خمسة	100	30

Figure 5.1 : Description du corpus

Tandis que pour la phase de détection des mots clés (apprentissage et test). On a construit un corpus composé de deux sous ensembles :

Le premier ensemble regroupe les phrases qui contiennent dedans un mots clés.

Le deuxième ensemble regroupe les phrases qui ne contiennent pas des mots clé.

Dans le tableaux suivant, un extrait des phrases enregistrées, sachant que c'est les mêmes locuteurs qui ont participés a l'enregistrement des ces phrases. Tandis que, le nombre de répétitions pour chaque phrase est de trois (05) fois pour le même locuteur. On a enregistré en tout 30 phrases (15 pour chaque ensemble) avec cinq répétitions pour sept locuteurs qui nous donnent en tous 1050 enregistrements. On note aussi que tous ces enregistrements sont enregistrés dans le même milieu ambiant (ce qui concerne le bruit et les parasites)

Ensemble	Phrase
1	أريد رقم ثلاثة
1	« hesit » واحد
2	الرقم هو سبعة
2	« silence » أريد الرقم ثلا
	.....

Figure 5.2 : Extrait des phrases enregistrés

5.2 Développement

Dans cette expérimentation, on propose une conception à Six (06) mots clés à reconnaître et une conception d'un modèle poubelle pour absorber les autres mots (mots hors vocabulaire, silences, les faux départs,…etc.).

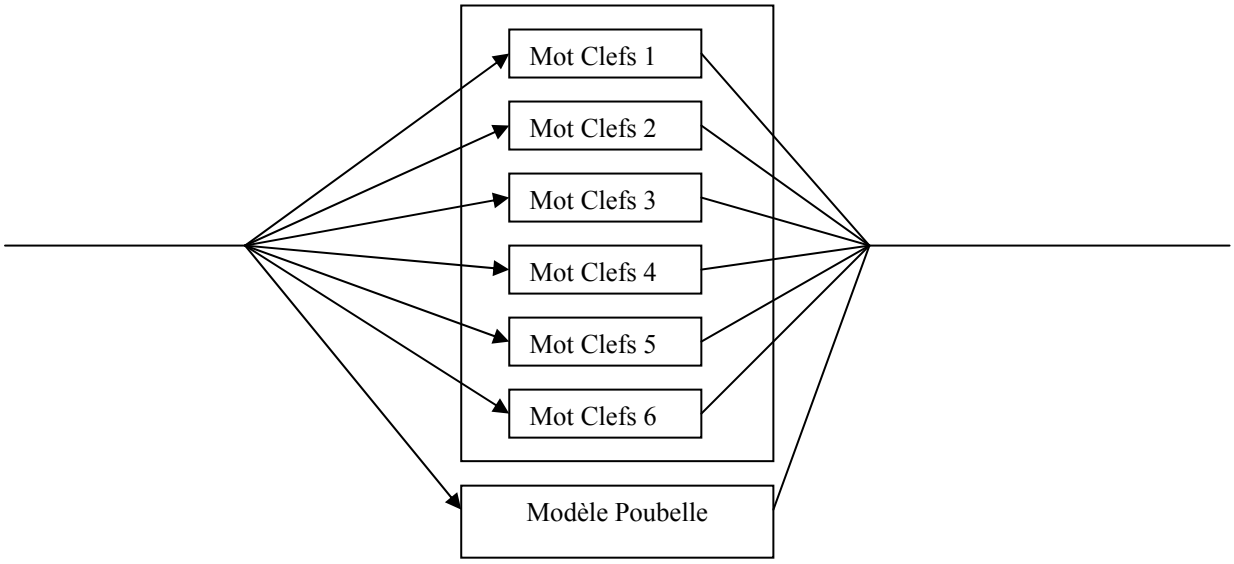


Figure 5.3 : Architecture du développement

Le développement est réalisé dans l'environnement MATLAB dans sa version 7.6.0 (R2008a). L'architecture des différentes routines est présentée dans l'organigramme suivant :

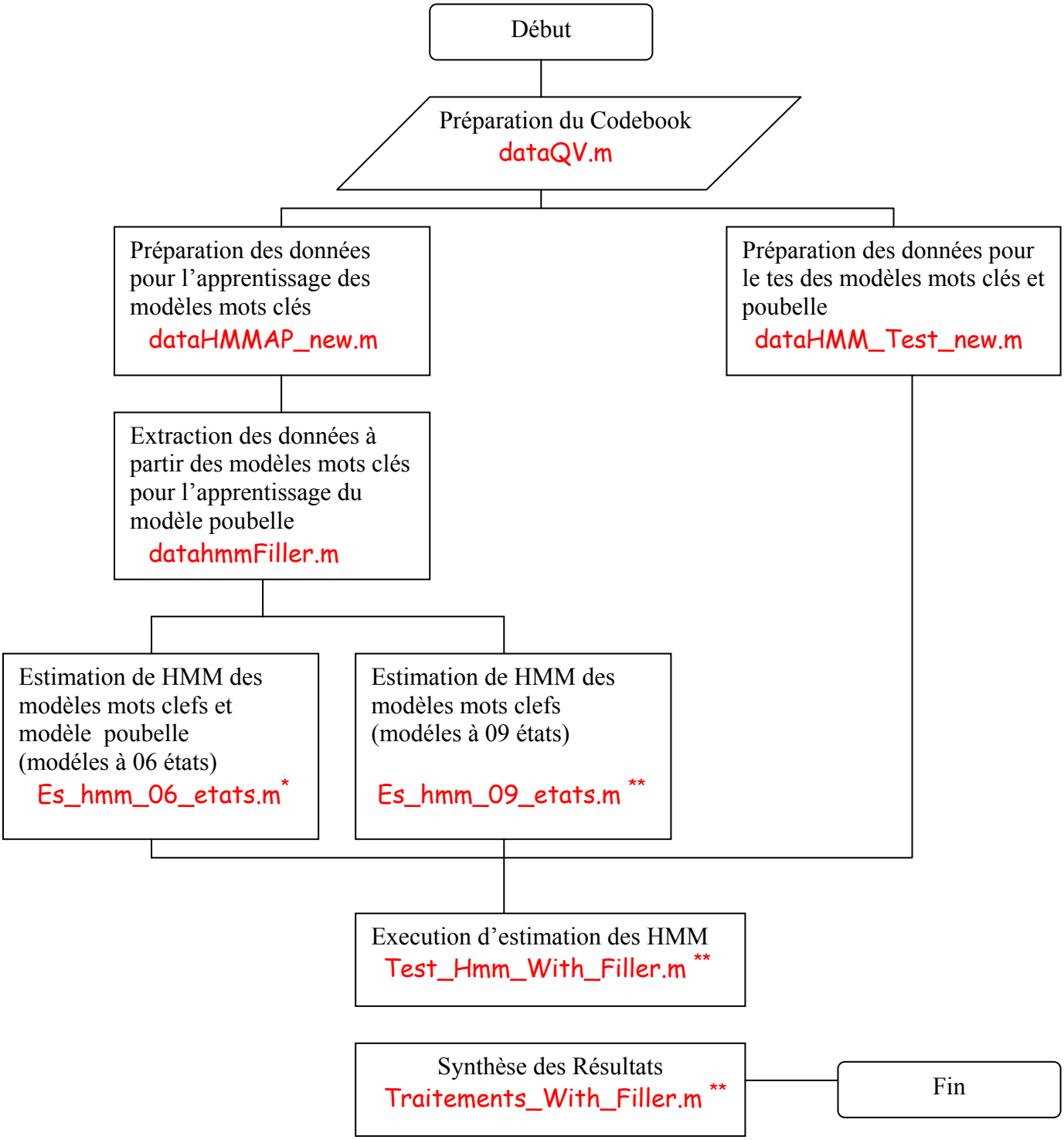


Figure 5.4 : Organigramme de développement

5.3 Résultats

5.3.1 Modèles Mots clefs

- Taux de Reconnaissance d'un mot clefs avec un HMM à 6 états plus les deux états de début et de fin.

Nombre de Paramètres	Mots clés 1	Mots clés 2	Mots clés 3	Mots clés 4	Mots clés 5	Mots clés 6
13 (MFCC +E)	66.3333	53.3333	73.3333	66.6667	53.3333	60.0000
39 ((MFCC +E) +Δ)	60.0000	56.6667	63.3333	60.0000	56.6667	53.3333
39 ((MFCC +E) +Δ+ΔΔ)	76.6667	66.3333	86.6667	80.0000	66.3333	93.3333

- Taux de Reconnaissance d'un mot clefs avec un HMM à 9 états plus les deux états de début et de fin.

Nombre de Paramètres	Mots clés 1	Mots clés 2	Mots clés 3	Mots clés 4	Mots clés 5	Mots clés 6
13 (MFCC +E)	60.0000	70.0000	73.3333	66.6667	66.6667	76.6667
26 ((MFCC +E) +Δ)	56.6667	66.6667	70.0000	60.0000	63.3333	73.3333
39 ((MFCC +E) +Δ+ΔΔ)	73.3333	93.3333	80.0000	60.0000	90.0000	86.6667

Note : Suite à ces résultats j'ai utilisé les deux variants : les mots clefs « صفر » , « اثنان » , «ثلاثة» et « خمسة » sont modélisés à l'aide des HMM à 6 états ; tandis que les mots clefs « واحد » et « أربعة » sont modélisés à l'aide des HMM à 9 états ; sans compté l'état de début et de fin.

- Taux de Reconnaissance d'un mot clefs avec la combinaison des HMM à 9 états et 6 états plus les deux états de début et fin

Nombre de Paramètres	Mots clés 1	Mots clés 2	Mots clés 3	Mots clés 4	Mots clés 5	Mots clés 6
13 (MFCC +E)	66.3333	70.0000	73.3333	66.6667	66.6667	60.0000
26 ((MFCC +E) +Δ)	60.0000	66.6667	63.3333	60.0000	63.3333	53.3333
39 ((MFCC +E) +Δ+ΔΔ)	76.6667	93.3333	86.6667	80.0000	90.0000	93.3333

5.3.2 Modèles Poubelles

- 1ére variante : Modèle Poubelles sans apprentissage

Taux de Reconnaissance du modèle poubelle

Topologie	Mots clés 1	Mots clés 2	Mots clés 3	Mots clés 4	Mots clés 5	Mots clés 6	Modèle Poubelle
HMM 3 états	1.6667	3.3333	1.6667	1.6667	3.3333	1.6667	86.6667
HMM 6 états	0.4167	0.8333	0.4167	0.4167	0.8333	0.4167	96.6667
HMM 9 états	9.1667	4.8533	4.8533	4.8533	9.1667	4.8533	63.3333

NB : On remarque que le modèle poubelle a une grande capacité d'absorber les mots hors vocabulaire

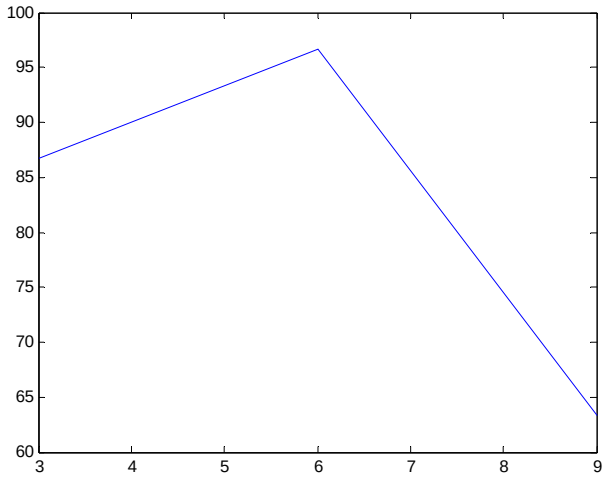


Figure 5.5 : Graphe de l'évolution de taux de détection

- 2éme variante : Modèle Poubelles avec apprentissage

Taux de Reconnaissance du modèle poubelle

Topologie	Mots clés 1	Mots clés 2	Mots clés 3	Mots clés 4	Mots clés 5	Mots clés 6	Modèle Poubelle
HMM 3 états	4.5833	11.4583	2.2917	2.2917	11.4583	4.5833	63.3333
HMM 6 états	1.6667	4.1667	1.6667	0.8333	4.1667	0.8333	86.6667
HMM 9 états	3.3333	8.3333	1.6667	1.6667	3.3333	8.3333	73.3333

NB : On remarque que la capacité du modèle poubelle d'absorber les mots hors vocabulaire est diminuée par rapport au première variante

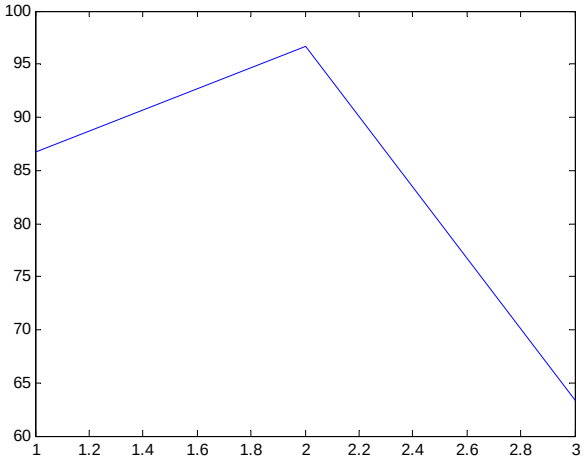


Figure 5.6 : Graphe de l'évolution de taux de détection

Conclusion

La discipline de la détection de mots clés est utilisée dans plusieurs applications interactives utilisant la parole comme moyen de communication. Elle consiste à améliorer les performances de telles applications en les aidant à détecter le sens des énoncés émis et ce en dévoilant seulement les mots porteur de sens pour l'application en question. La liste des mots clés les plus significatifs pour l'application est déterminée auparavant.

Dans la littérature, la détection de mots clés a donné lieu à une grande variété de travaux depuis la fin des années 80. En effet, dès l'apparition des applications interactives, les chercheurs du domaine ont adopté cette technique comme solution pour pouvoir comprendre et satisfaire les requêtes des utilisateurs sans les restreindre à suivre un vocabulaire rigide et limité. Nous distinguons alors deux approches principales utilisées pour détecter les mots clés d'une application donnée. La première, dans l'ordre chronologique, est basée sur l'utilisation des modèles poubelles. La deuxième, qui constitue l'alternative la plus simple, utilise une ou plusieurs mesures de confiance pour décider si un mot reconnu représente bien un mot clé réellement prononcé ou si au contraire il s'agit d'un mot inséré. Dans notre travail, on a essayé d'inspiré de ces travaux pour les intégrés dans le domaine du recherche d'information (IR) afin d'exploité les bases de données audio.

La détection de mots clés peut être simplement réalisée à l'aide d'un système de reconnaissance à grand vocabulaire. Il s'agit de modéliser tous les mots clés et les mots hors-vocabulaire afin de les reconnaître pour ne garder enfin que l'ensemble des mots clés prédéfinis. Tout d'abord nous avons proposé un modèle poubelle représenté par un GMM. Un GMM est un modèle HMM à seul état dont la fonction de densité est composée d'un mélange de gaussiennes et dont les paramètres sont fixés lors de la phase d'apprentissage. Ensuite, nous avons proposé un modèle a base de boucle de phonèmes constituant un mot clé donné. En effet, avec une reconnaissance à base de boucle de phonèmes, on n'est plus limité à un modèle poubelle bien déterminé, les mots hors-vocabulaire sont remplacés par des séquences de phonèmes qui seront ignorés dans la phase de détection.

Après avoir étudié l'utilisation des modèles poubelles pour la détection de mots clés, nous nous sommes penché sur la notion de mesure de confiance qui permet de rejeter les mots les moins fiables donnés en sortie du système de reconnaissance. Nous avons utilisé des mesures de confiance à base de rapport et distance de vraisemblance. Ces mesures sont calculées comme étant les moyennes arithmétique des probabilités d'observations acoustiques locales des phonèmes composant un mot clé reconnu.

Perspectives

Nous envisageons un certain nombre de perspectives pour la poursuite de ce travail.

Tout d'abord, au niveau du corpus arabe utilisé, nous envisageons de l'élargir en ajoutons des locuteurs et des enregistrements, sans oublier qu'on essaye toujours de normaliser ce dernier.

Pour les modèles poubelles, nous pensons à l'introduction de la notion de syllabe dans la construction des modèles.

Une des perspectives les plus attrayantes consiste à intégrer notre système de détection dans une plate-forme réelle pour la consultation orale de bases de données sécurisée, ou pour mener des recherches sur le Web en utilisant la parole. Pour cela, il nous faut envisager un certain nombre de points. La première étape à réaliser consiste à utiliser des bases d'évaluation contenant de la parole spontanée. La deuxième concerne l'augmentation du nombre des mots clés considérés.

Bibliographies

Bibliographies

- [Bahl et al, 1991], Bahl, L., P. de Souza, P. Gopalakrishnan, D. Nahamoo & M. Picheny. "Decision Trees for Phonological Rules in Continuous Speech". ICASSP, Toronto, pp. I.185-188, 1991.
- [Bahl et al, 1993], Bahl, L., P. Brown, P. de Souza, R. Mercer & M. Picheny. "A Method for the Construction of Acoustic Markov Models for Words" IEEE Transactions on Speech and Audio Processing 1(4):443-452, 1993.
- [Baker, 1975] J. K. Baker, Stochastic modeling for automatic speech understanding, Speech Recognition, Academic Press, pp. 521–542, 1975.
- [Barras, 1996], Barras, C. Reconnaissance de la parole continue : adaptation au locuteur et contrôle temporel dans les modèles de Markov cachés. Thèse de Doctorat Université Pierre et Marie Curie, Paris. Web, 1996.
- [Baum, 1970]. Baum, "A Maximization Technique Occuring in the Statistical Analysis of Probabilistic Functions of Markov Chains" Annals of Mathematical Statistics 41:164-171, 1970.
- [BenAyed, 2003]. BenAyed Y, « Détection de flux de parole ». Thèse de Doctorat, Ecole Nationale Supérieure des télécommunications, 2003.
- [Bellagarda et Nahamoo, 1990], Bellagarda, J. & D. Nahamoo. "Tied Mixture Continuous Parameter Modeling for Speech Recognition" IEEE Transactions on Acoustics, Speech and Signal Processing 38(12):2033-2045, 1990.
- [Bellange, 1995] Bellanger M., Traitement numérique du signal, Théorie et pratique, éditions Masson, 1ère édition en 1980. ISBN 2-225-84997-8,1995.
- [Black, 2000] Black U., Voice Over IP, édition Prentice Hall PTR, ISBN 0-13-022463-4, 2000.
- [Cacoullos, 1966], Cacoullos, T. "Estimation of a Multivariate Density" Annals of Inst. of Stat. Math. 18:179-189, 1966.
- [Choy et Leung, 1998] C. Y. Choy et H. C. Leung. "Subword units for a mandarin keyword spotting". In Proceedings of the International Symposium on Chinese Spoken Language Processing 1998.
- [Cuayàhuitl et Serridge, 2002] H. Cuayàhuitl et B. Serridge. "Out-Of-Vocabulary Word Modelling and Rejection for Spanish Keyword Spotting Systems". In Proceedings Second Mexican International Conference on Artificial Intelligence, pages 156-165, 2002.
- [DARPA, 1998] DARPA. DARPA Broadcast News Transcription and Understanding Workshop, Lansdowe, Virginia. Web, 1998
- [Davis et al, 1980] Davis S., Mermelstein P., Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences, IEEE Transactions ASSP, ASSP-28(4), pp. 357-366, 1980.
- [de Mori, 1995], de Mori, R., M. Galler & F. Brugnara. "Search and Learning Strategies for Improving Hidden Markov Models" Computer Speech and Language 9:107-121, 1995.

▪ [Derouault, 1987], Derouault, A.-M. "Context-dependent Phonetic Markov Models for Large Vocabulary Speech Recognition". ICASSP, Dallas, pp. I.360-363, 1987.

▪ [Duchateau, 1989], Duchateau, J. HMM-based Acoustic Modelling in Large Vocabulary Speech Recognition. PhD Thesis Université Catholique, Leuven, 1998.

▪ [Favre, 2007] B. Favre, « Résumé automatique de parole pour un accès efficace aux bases de données audio ». Thèse de Doctorat, Université d'avignon et des pays de vaucluse, 2007.

▪ [Haton et al, 1991] Haton J.-P, Pierrel J.-M, Perennou G, Caelen J et Gauvain J.-L, Reconnaissance automatique de la parole, Dunod, Paris, 1991.

▪ [Hunt et al, 1980], Hunt J., M. Lenning & P. Mermelstein. "Experiments in syllabe-based recognition of continuous speech". ICASSP, Denver, pp. I.880-883, 1980.

▪ [Huang, 2001] X. Huang, A. Acero, H-W. Hon, Spoken Language Processing – A Guide to Theory, Algorithm, and System Development, Practice Hall, 2001.

▪ [Jelinek, 1976] F. Jelinek, Continuous Speech Recognition by Statistical Methods, IEEE Trans. on ASSP, vol 64(4), pp. 532-556, Avril 1976.

▪ [Jouvet, 1995], Jouvet, D. "Modèles de Markov pour la reconnaissance de la parole". Ecole Thématique "Fondements et Perspectives en Traitement Automatique de la Parole", Marseille, pp. I.99-108, 1995.

▪ [Juang et Rabiner, 1985], Juang, B.-H. & L. Rabiner. "Mixture Autoregressive Hidden Markov Models for Speech Signals" IEEE Transactions on Acoustics, Speech and Signal Processing 33(6):1404-1413, 1985.

▪ [Lee, 1990]Lee, K.-F. "Context-Dependent Phonetic Hidden Markov Models for Speaker-Independent Continuous Speech Recognition" IEEE Transactions on Pattern Analysis and Machine Intelligence 38(4):599-609, 1990.

▪ [Liporace, 1982], Liporace, L. "Maximum Likelihood Estimation for Multivariate Observations of Markov Sources" IEEE Transactions on Information Theory 28(5):729-734, 1982.

▪ [Ljolie, 1994], Ljolje, A, "High Accuracy Phone recogntion Using Context Clustering and Quasi-triphones Models" Computer Speech and Language 8:129-151, 1994.

▪ [Meliani et O'Shaughnessv, 1998] R. El Meliani et D. O'Shaughnessy. "Specific language modelling for new-word detection in continuous-speech recognition". In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 321 325, 1998.

▪ [Ney et Noll, 1988], Ney, H. & A. Noll. "Phoneme Modelling Using Continuous Mixtures Densities". ICASSP, New-York, 1988.

▪ [Ortmanns et al, 1997] Ortmanns, S., T. Firzlaff & H. Ney . "Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition". Eurospeech, Rhodes, pp. I.139-142, 1997.

▪ [Parzen, 1962], Parzen, E. "On Estimation of a Probability Density Function and Mode" Annals of Mathematical Statistics 33:1065-1076, 1962.

▪ [Press et al, 1992] Press W.H., Teukolsky S.A., Vetterling W.T., Flannery B.P., Numerical recipes in C: The Art of Scientific Computing (seconde édition), édition Cambridge University Press, ISBN 0-521-43108-5, 1992. Téléchargeable gratuitement sur le site :
http://www.ulib.org/webRoot/Books/Numerical_Recipes/.

▪ [Rabiner et al. 1989] Rabiner L. R., Juang B., Tutorial on hidden Markov models and selected applications in speech recognition, Proceedings of the IEEE, vol. 77, no. 2, pp. 257-285, 1989.

▪ [Rabiner et al, 1993] Rabiner L., Juang B.H., Fundamentals of Speech Recognition, édition Prentice Hall PTR, ISBN 0-130-15157-2, 1993.

▪ [Richter, 1986], Richter, A. "Modeling of Continuous Speech Observations". Advances in Speech Processing Conference, IBM Europe Institute, 1986.

▪ [Rohlicek et al., 1989] J. R. Rohlicek, W. Russell, S. Roukos, et H. Gish, “Continuous HMM for speaker independent word spotting”, In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 627 630, 1989.

▪ [Rose et Paul, 1990] R. C. Rose et D. B. Paul. “A Hidden Markov Model based keyword recognition system”. In Proceedings of the International Conference on Acoustics, Speech and Signal Processing, pages 129 132, 1990.

▪ [Salton, 1983] Salton, G et McGill, M. Introduction to modern Information Retrieval. New York, McGraw-Hill Book Company, 1983

▪ [Schwartz et al, 1980] Schwartz, R., J. Klovstad, J. Makhoul & J. Sorensen . "A Preliminary Design of a Phonetic Vocoder Based on a Diphone Model". ICASSP, Denver, pp. I.32-35, 1980.

▪ [Silverman, 1990]. Silverman H-F, D-P. Morgan, The application of dynamic programming to connected speech recognition, IEEE ASSP magazine, vol.7, pp.6-25, 1990.

▪ [Stevens et al, 1940] Stevens S., Volkman J., The relation of pitch to frequency, American Journal of Psychology, vol. 53, 1940.

▪ [Szöke et al, 2005] I. Szöke, P. Shwarz, L. Burget, M. Karafiàt et J. Cernocky. “Phoneme based acoustics keyword spotting in informal continuous speech”. In proceeding Text, speech and dialogue. International conference 2005.

[Taboada et al. 1994] Taboada J., Feijoo S., Balsa R., Hernandez C., Explicit estimation of speech boundaries, IEEE Proc. Sci. Meas. Technol., vol. 141, pp. 153-159, 1994.

▪ [Umesh et al, 1999] Umesh S., Cohen L., Nelson D., Fitting the mel scale, ICASSP'99, vol. 1, pp. 217-220, Phoenix Arizona (USA), mars 1999.

Annexe 1

Publications

Presentation of password verification system in Arabic

H. Bahi[1], I. Bendib[2]
[1] *LabGEG laboratory, university of Annaba-Algeria*
[2] *University of Tebessa-Algeria*
bahi@lri-annaba.net, bendibissam@yahoo.fr

Abstract

The advances made in automatic speech recognition (ASR) lead to prospecting other possible research areas issued from ASR. In this context, the keyword spotting is a recent research domain issued from the automatic speech recognition. In this paper, we present our investigations concerning this area, and we propose a conceptual model for a password verification system based on keywords spotting, and we are particularly interested with Arabic language.
Key words: Keyword spotting, filler model, hidden Markov models, password verification system.

1. Introduction

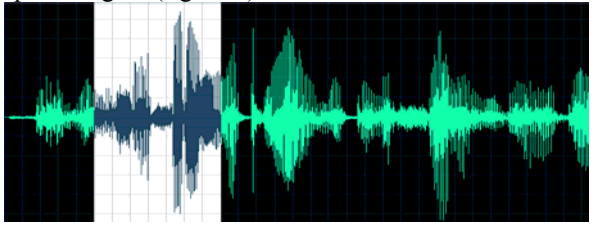
The advances made in automatic speech recognition (ASR) lead to prospecting other possible research areas issued from ASR. In this context, the keyword spotting is a recent research domain issued from the automatic speech recognition.

Keywords spotting consists on detecting a set of keywords in a speech signal. This is very suitable in applications concerning audio information retrieval, where a request involves few words to find in the database. Some approaches were dedicated to this task, and a many potential applications were developed. In this paper, we present our investigations concerning this area, and we propose a conceptual model for a password verification system based on keywords spotting.

2. Keyword spotting

2.1. Keyword spotting

Keyword spotting is the task of identifying the occurrences of certain desired keyword in an arbitrary speech signal (figure 1).



[wa][ʔ i ð][bi] [musafirin] [yatluʕ u] [mutalafiʕ an][bi] [ʕ abaʔ atin] [samika]

Figure. 1. Example of a word to detect in speech strem ([musafirin])

Keyword spotting is issued from speech recognition. Although in speech recognition, an occasional unknown word punctuates a stream of known words. While in word spotting, a relatively small number of keywords float on a sea of unknown words.

Despite this difference in viewpoint, some implementations of the two systems may be very similar.

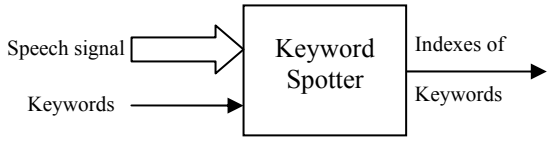


Figure 2. Representation of keyword-spotter

Many potential applications of keyword spotting could be developed ranging from simple routing telephone system to indexing speech corpora.

2.2. Filler models

In word spotting task, the speech signal is assumed to be composed of a combination of keywords and non-keywords speech. One of the most approaches for the keywords spotter implementation is to consider individual models for the keywords and to represent other words by "filler" or "garbage" models. Some systems add extra models for no-speech events, such as coughing or silence (Figure 3).

A particular attention is paid to the keywords modeling, while several strategies were suggested to implement the filler models [1,2,3].

When the vocabulary is known and restricted, it is possible to use a large vocabulary speech recognizer as a word-spotter, since non-keywords could be modeled.

Sometimes, transcribed data is available for a domain, so, word spotting benefits from the more detailed background model.

But, often, number of keywords is very large and the available training data is limited, so, a more general filler models should be used.

When non-keywords are not present in the training data, Rohlicek[4] modeled non-keywords as segments of the keywords.

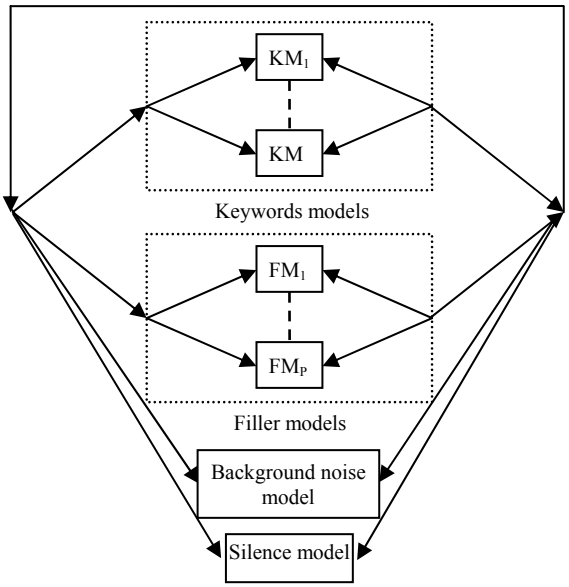


Figure 3. Word spotter models

3. Password verification system

Keywords spotting has a fundamental role in audio/video information retrieval applications, but it could also be used in many innovative applications. In particular, we are interested with the use of keywords spotting in a password verification system.

Several systems need to perform some verification over the user response. Password verification could be done by a classical isolated word recognition, but using a keywords spotter provides more flexibility to the speaker, since it permits introduction of silence, hesitations or street background noise.

3.1. Presentation

In the proposed system the user phone to a voice server, insert his identification number then he uttered his password to access information concerning his postal account. The password consists on a sequence of four Arabic digits. The system identifies the keywords (digits), the obtained sequence is then compared to the sequence corresponding to the identification number. If the two sequences are similar, The client is allowed to access to its account information (figure 4)

.

3.2. Conceptual elements

As for speech recognizers, the first stage of the system is the features extraction, then a comparison is done between the current signal and the reference models. Here, both keywords models and filler models are often HMM-based.

3.2.1. Features extraction

In the following are described the several steps performed in the features extraction stage:

The incoming signal is sampled at 22 KHz, with 16 bits of precision. The sampled signal is processed by a first-order digital filter in order to spectrally flatten the signal.

Sections of 400 consecutives samples are blocked into a single frame, corresponding to 400/11.025 ≈ 36ms. Frames are spaced M samples (M=100). Each frame is multiplied by a N-sample Hamming window. MFCC computation: from each frame, we extract a set of 13 Mel-scale Frequency Ceptral Coefficients (MFCCs) and the energy.

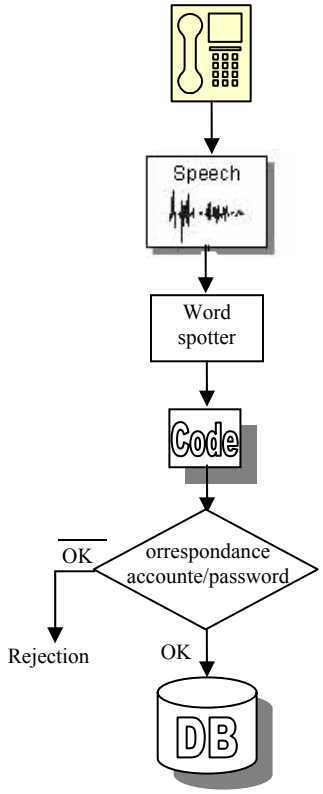


Figure 4. The password verification system

3.2.2. Models of keywords

The most flexible and successful approach to speech recognition has been hidden Markov models (HMMs).

A Hidden Markov Model is a collection of states connected by transitions, as illustrated in Figure 5. It begins in a designated initial state. In each discrete time step, a transition is taken in a new state, and an output symbol is generated in the state.

When an HMM is applied to speech recognition, the states are interpreted as acoustic models, indicating what sounds are likely to be heard during their corresponding segment of speech ; while the transitions provide temporal constraints, indicating how the states may follow each other in sequence.

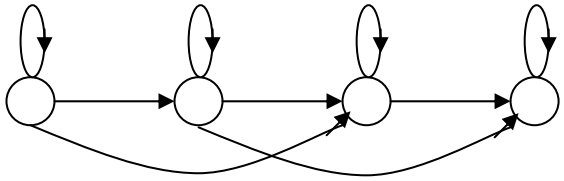


Figure 5. Example of HMM model

An HMM is defined by giving the following elements [5,6]:

{s} is a set of states, each word has as much states as there is phonemes in its structure.

$\{a_{ij}\}$ is a set of transitions probabilities, where a_{ij} is the probability of making the transition from state i to state j.

$\{b_j(O)\}$ is a set of emission probabilities, where b_i is the probability distribution over the acoustic space describing the likelihood of emitting each possible sound O in state j. We use in this application continuous HMMs. In the continuous HMM the density probabilities for the observations are not discrete but continuous, and the observations are not just a single symbol but a vector (here a cepstral vector). The representation most widely used is:

$$b_j(O)=\sum_{k=1}^{M} c_{jk} N(O,\mu_{jk},U_{jk}), i<j<n$$

where N is most of the time assumed to be Gaussian, with covariance matrix U_{jk} for the k^{th} mixture component in state j. μ_{jk} is the mean vector and c_{kj} is the mixture coefficient for the k_{th} component in state j. For our models we assume a mixture of five Gaussian components.

While keyword models represent the ten Arabic digits, the non-keywords are also, whole-word models. They include a model for one syllables words, another for two syllables words, and models for three, four, five and six syllables words. Because syllables are suitable for Arabic words decomposition [7].

The non-speech models represent silence, hesitations, coughing or street background noise.

4. Conclusion

Keyword spotting is an innovative research area which has many applications. This paper has presented method of filler models used to build word spotters. As an application of this approach, we implement a password verification system. In particular, we are working on adaptation of the current techniques with filler models according to Arabic language. We are also working to promote the role of syllables in the keywords modeling.

5. References

[1] [1] Wilpon, J. G. , Rabiner, L. R., Lee, C. , Glodman, E.R., "Automatic recognition of keywords in unconstrained speech using hidden markov models". Trans ASSP vol. 38, no. 11, 1990.

[2] Lleida, E., Marino, J.B., Salaverda, J., Bonafonte, A., "*Martinez, A., Out of vocabulary word spotting modelling and rejection for keyword spotting"*. EUROSPEECH, pp. 1265-1268, 1993.

[3] P. Fitzpatrick, "From word-spotting to OOV models", Automatic speech recognition, 2001.

[4] Rohlicek, J. R., Jeanrenaud, P., Ng, K., Gish, H., Musicus, B., Siu, M., "Phonetic training and language modeling for word spotting*"*. ICASSP, pp. II 459-II 462, 1993.

[5] L. Rabiner, B. Hwang, *Fundamentals of speech recognition*, Prentice Hall, 1993.

[6] C. Becchitti, L. P. Ricotti, *Speech recognition: theory and C++ implementation*, John Wiley, Angleterre, 1999.

[7] H. Bahi, M. Sellami., "An acoustical based approach for arabic syllables recognition", Workshop on software for the arabic language, AICCSA'2001 , Beirut, Liban, 2001.

Annexe 2

Extrait du Programmation

```
function [c]=cmlfcc(s,fs,l, step,k)

p=k;
n=l;
fl=0;
fh=0.5;
w='m';
inc=step;
% creation des frames ponderees par Hamming
z=myframes(s,fs, l, step);

eng=energie(z);

% Premiere etape : f est la transorme rapide de Fourier de z
f=fft(z);

[m,a,b]=melbankm(p,n,fs,fl,fh,w);

y=log(m*(abs(f(a:b,:)).^2));

c=rdct(y,n).';

nc=k+1;
c(:,nc+1:end)=[];
% on supprime la premiere colonne
  c(:,1)=[];
  nf=size(c,1);
  nc=13;
  vf=(4:-1:-4)/60;
  af=(1:-1:-1)/2;
  ww=ones(5,1);
  cx=[c(ww,:); c; c(nf*ww,:)];
  vx=reshape(filter(vf,1,cx(:)),nf+10,nc);
  vx(1:8,:)=[];
  ax=reshape(filter(af,1,vx(:)),nf+2,nc);
  ax(1:2,:)=[];
  vx([1 nf+2],:)=[];
  c=[c vx ax];
c=[c eng'];
%********************FONCTION MELBANKM*****************************
function [x,mn,mx]=melbankm(p,n,fs,fl,fh,w)

if nargin < 6
  w='tz';
  if nargin < 5
    fh=0.5;
    if nargin < 4
      fl=0;
    end
  end
end

f0=700/fs;
fn2=floor(n/2);
lr=log((f0+fh)/(f0+fl))/(p+1);
```

```
% convert to fft bin numbers with 0 for DC term

bl=n*((f0+fl)*exp([0 1 p p+1]*lr)-f0);
b2=ceil(bl(2));
b3=floor(bl(3));
if any(w=='y')
  pf=log((f0+(b2:b3)/n)/(f0+fl))/lr;
  fp=floor(pf);
  r=[ones(1,b2) fp fp+1 p*ones(1,fn2-b3)];
  c=[1:b3+1 b2+1:fn2+1];
  v=2*[0.5 ones(1,b2-1) 1-pf+fp pf-fp ones(1,fn2-b3-1) 0.5];
  mn=1;
  mx=fn2+1;
else
  b1=floor(bl(1))+1;
  b4=min(fn2,ceil(bl(4)))-1;
  pf=log((f0+(b1:b4)/n)/(f0+fl))/lr;
  fp=floor(pf);
  pm=pf-fp;
  k2=b2-b1+1;
  k3=b3-b1+1;
  k4=b4-b1+1;
  r=[fp(k2:k4) 1+fp(1:k3)];
  c=[k2:k4 1:k3];
  v=2*[1-pm(k2:k4) pm(1:k3)];
  mn=b1+1;
  mx=b4+1;
end
if any(w=='n')
  v=1-cos(v*pi/2);
elseif any(w=='m')
  v=1-0.92/1.08*cos(v*pi/2);
end
if nargout > 1
  x=sparse(r,c,v);
else
  x=sparse(r,c+mn-1,v,p,1+fn2);
end
```