République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche Scientifique

Université Hadj Lakhdar Batna

Faculté de Technologie

Département de Génie Industriel

Présenté au

DEPARTEMENT DE GENIE INDUSTRIEL

Pour l'obtention du diplôme de

MAGISTER

Option : Génie industriel et Productique

Par

Amar Zaidi

Ingénieur d'état en informatique

Thème :

Etude et conception d'un serveur vocal : Application aux comptes client d'une banque

Encadreur : Dr.Samir Abdelhamid

Membres de jury

Mohamed Djamel Mouss	M.C. A. Université de Batna	Président
Samir Abdelhamid	M.C. A. Université de Batna	Rapporteur
Nabil Benoudjit	Professeur. Université de Batna	Examinateur
Allaoua Chaoui	Professeur. Université de Constantine	Examinateur

Session 2012/2013

Remerciements

Tout d'abord je voudrais remercier Mr Samir abdelhamid, mon encadreur de thèse, pour m'avoir encadré pendant ce travail de recherche.

Je remercie tous les membres de mon jury, à savoir Mr Samir Abdelhamid pour avoir accepté d'être rapporteurs de ma thèse, Mr Nabil Benoudjit et Mr Allaoua Chaoui pour en avoir été examinateurs de mon mémoire, et M.Djamel Mouss pour sa présidence.

Toute ma gratitude à toutes les personnes ayant relu, corrigé et commenté mon manuscrit et ayant ainsi participé à son amélioration, un grand remerciement à Abdellah Lahoual pour son aide et sa patience pendant la correction de cinquième chapitre.

Enfin, pour leur soutien tout au long de ce mémoire et de sa rédaction, je remercie tout particulièrement Halim Zerraf, Raouf Mehanaoui, Mohamed Boussalem, Hicham Rahab, et Lefdhal Sebaa.

Dédicace

Ce mémoire est dédié à ma maman, qui m'a toujours poussé et motivé dans mes études. Sans elle, je n’aurais certainement pas fait d'études approfondies. Qu'elle en soit remerciée par cette trop modeste dédicace

Résumé

Le but final de ce mémoire de magistère est de développer un outil de la reconnaissance automatique de la parole, prenant comme corpus les chiffres arabes, et dont l'interaction homme-machine se fait à distance, en se basant sur la voix sur IP (VOIP voice over IP).

Le développement inclut une étude du modèle de Markov caché qui est actuellement l'état de l'art du champ de la reconnaissance automatique de la parole (RAP) ; aussi une étude et une comparaison entre les deux protocoles les plus utilisés dans la VOIP (SIP et H323).

Dans la vie réelle la RAP est très utile dans une large gamme de domaines (soit en sécurité ou en médecine ou en robotique etc.…), et dans le but est de faciliter tout types d'activités pénibles.

Ce rapport représente un outil pour la RAP des chiffres arabes avec la possibilité de l'interaction à distance et dont sa base de connaissance est extensible, c'est-à-dire l'utilisateur a la possibilité d'ajouter autre forme des chiffres pour être reconnus.

Mots clés : Vocal, dialogue, reconnaissance, voix sur IP, protocoles.

Abstract

The purpose with this memo was to develop a speech recognition tool , taking the Arabic numbers as corpus; using the Voice Over IP (VOIP) technology to make the man-machine interaction from distance.

The development of this tool include a study of the Hidden Markov Model (HMM), which is currently the state of the art in the field of speech recognition, and another study of the two famous protocols used in VOIP(SIP and H232) , with a comparison of this two protocols.

In the real life the speech recognition is very useful in the wide range of areas (security, or medical or robotic etc...), in order to simplify all types of strenuous activities.

This report provides a tool of speech recognition for Arabic numbers with a possibility of remote interaction and which the knowledge base is extensible, i.e. the user has the possibility to add other number forms to be recognized.

Keywords: Speech, Dialog, Recognition, Voice over IP, Protocols.

Table des matières

Chapitre III : Les modèles de Markov cachés

Chapitre VI : La voix sur IP et les serveurs vocaux Interactifs (les protocoles h323 et SIP)

Chapitre V : Expérimentations et résultats

Liste des figures

Liste des tables

Introduction générale

Introduction générale

Un serveur vocal interactif (en anglais *Interactive Voice Response*) est un système informatique qui permet l'accès à une base de données, en demandant un service, au moyen d'un téléphone fixe, mobile ou d'un softphone. Les serveurs vocaux interactifs entrent plus généralement dans la catégorie des *systèmes de dialogue*. Dans les serveurs vocaux interactifs, les interactions consistent, la plupart du temps, en des cycles pendant lesquels le système diffuse un intitulé préenregistré (bande magnétique ou fichier audio) après quoi la personne est invitée à choisir une option parmi une liste de choix.

À l'origine, pour permettre l'interaction des utilisateurs, les serveurs vocaux utilisaient les codes DTMF, c'est des fréquences engendrées par les touches des téléphones, où les services proposaient sont choisis par des numéros, il suffit de taper le chiffre sur son clavier pour obtenir le service désiré.

La reconnaissance vocale apporte un mode d'interaction à la fois plus naturel et plus pratique, mais surtout autorise la création de serveurs interactifs nettement plus riches. Cet apport a son influence sur les SVIs, en générant de nouveaux SVIs dits (Serveurs Vocaux Interactifs Avancés).

Avec les nouveaux SVIs, le client peut communiquer avec l'ordinateur en utilisant de simples commandes vocales, ce qui fait agrandir l'entreprise servante dans ses yeux sachant que l'installation d'un SVI est simple et ne coûte pas chère.

on peut résumer les paragraphes ci-dessus par le graphe suivant

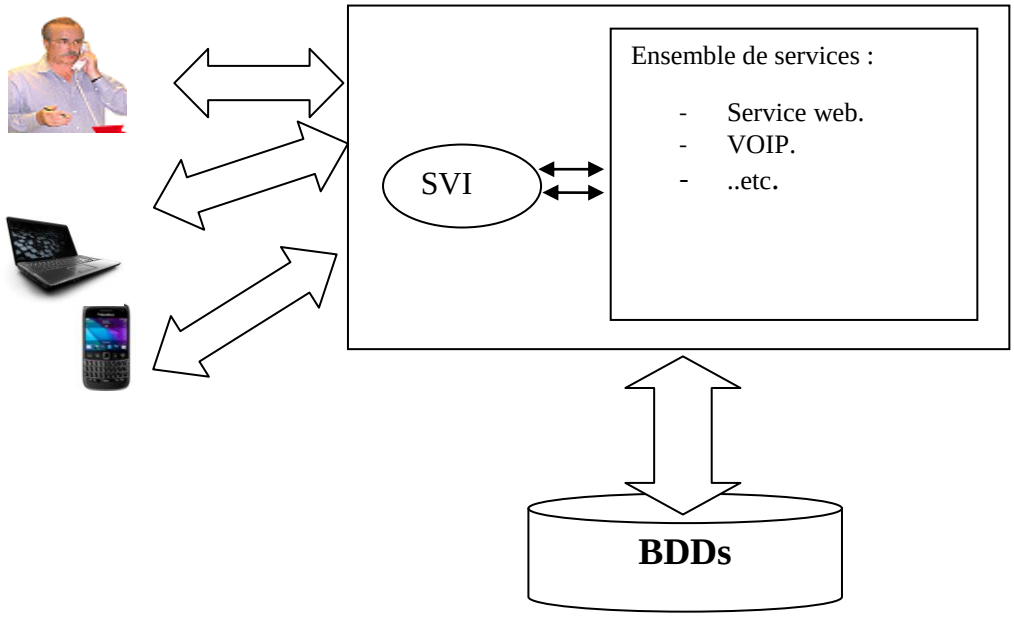


Figure 1 ***Présentation générale d'un SVI***

Donc un SVI englobe un ensemble de services où l'on trouve les services web, les applications VOIP (Voice Over IP) ou voix sur IP en français, la reconnaissance vocale (reconnaissance du locuteur ou reconnaissance de la parole ou même reconnaissance de la langue), et d'autres.

Ce mémoire traite la reconnaissance vocale des chiffres arabes en vue de construire un système de dictée pour les comptes bancaires, elle s'articule autour de deux applications des SVIs, qui sont la VOIP (Voice Over IP) et la reconnaissance vocale ; pour la VOIP, on a essayé de développer une application qui enregistre les appels des clients dans le serveur, donc chaque appel établi s'enregistre automatiquement dans le serveur, puis déclenche la deuxième application, celle de la reconnaissance du mot parlé, une fois le mot est reconnu la réponse sera envoyée au client et elle s'affichera sur son interface, ce qui lui donnera le droit de prononcer le deuxième mot et ainsi de suite.

Ce mémoire se compose de cinq chapitres dont le premier représente une introduction aux modèles utilisés dans la reconnaissance vocale, le deuxième nous donne l'ordre chronologique de la production et le traitement du son depuis sa sortie du larynx jusqu'à sa numérisation et l'extraction de ses coefficients caractéristiques pour la reconnaissance. Le troisième chapitre est une représentation d'outils mathématiques sur lesquels se base la reconnaissance de la parole. Le quatrième chapitre est dédié à la VOIP et les protocoles (H323, SIP) célèbres dans le domaine de la VOIP .En fin le dernier chapitre concerne notre contribution personnelle avec des démonstrations de l'application développée.

Chapitre 1 : **la reconnaissance automatique de la parole** dans lequel on essaye d'introduire le domaine de la reconnaissance automatique de la parole par la présentation des méthodes connues dans la littérature. La reconnaissance automatique de la parole passe par des étapes parmi elles on cite l'étape acoustique, l'étape acoustico-phonétique, et l'étape syntaxique, où les modèles acoustiques sont utilisés pour la reconnaissance des mots isolés, et le modèle de langage (désigné au module syntaxique) pour la reconnaissance de la parole continue.

Chapitre 2 : **Le traitement de signal en vue de sa reconnaissance** dans ce chapitre on démontre les étapes de la production du son à partir de la bouche humaine jusqu'à son enregistrement dans une forme électronique et les transformations appliquées sur la voix dans sa forme électronique pour avoir des données permettant l'application de la méthode de reconnaissance. La transformation usagée est la transformation de Fourier rapide (TFR, ou

FFT en anglais) qui donne comme résultat un graphe représente tant la variation de l'amplitude de la voix en fonction de sa fréquence.

On passe à la représentation de quelques paramètres dits caractéristiques, sur lesquels se base la caractérisation et la classification de la voix.

Chapitre 3 **: les modèles de Markov cachés (HMM**) dans lesquels on sort de l'informatique et on entame les processus stochastiques, ce chapitre n'est qu'un état de l'art sur la méthode adoptée (HMM Hidden Markov Model), en français le modèle de Markov caché MMC, ce modèle est un processus stochastique définis par l'ensemble des données suivantes :

- Les probabilités initiales des états $\Pi = \{\pi i = P(si)\}$
- Le modèle de transition des états
  - ✓ L'alphabet $\Sigma = \{s_1,...,s_m\}$ décrivant les états de la chaîne de Markov
  - ✓ La matrice des probabilités de transitions entre états $A = \{a_{ij} = P(s_j|s_i)\}$
- Le modèle d'observation de l'évidence
  - ✓ L'alphabet $\Omega = \{o_1,..., o_k\}$ des symboles émis par les s_i pour un HMM discret
  - ✓ Les probabilités d'émission $B = \{b_i(o_k) = P(o_k|s_i)\}$.

Son point fort est la permission de créer des modèles d'évaluation, ou d'apprentissage, ou de classification, et dans ce chapitre on va détailler ces trois modèles en se basant sur trois questions et leurs réponses, ou chaque réponse est un cas des modèles précédents.

Chapitre 4 : **la voix sur IP et les serveurs vocaux** dans ce chapitre on va présenter l'apport, d'utiliser les SVIs, et les deux principaux protocoles utilisés dans la VOIP qui sont (le SIP et le H323), avec leurs mécanismes d'envois de son sur IP et on termine le chapitre par une étude comparative entre ces deux protocoles.

Chapitre 5 **Expérimentations et résultats** dans ce chapitre nous présentons notre expérimentation avec les outils qui nous aidaient à progresser dans le travail, et les étapes suivies dès le commencement d'acquisition de la voix jusqu'à la reconnaissance, avec les difficultés rencontrés pendant l'évolution des programmes soient ceux de la VOIP ou ceux liés à la RAP.

Nous terminerons ce manuscrit par la présentation de nos perspectives concernant l'utilisation du mélange VOIP et reconnaissance de la parole dans le cadre de la situation actuelle et les nouvelles technologies.

Chapitre I : la reconnaissance automatique de la parole

1.1. Introduction

La reconnaissance automatique de la parole (RAP) consiste à extraire, à l'aide d'un ordinateur, l'information lexicale contenue dans un signal de parole.

Les fondements de la technologie récente en reconnaissance de la parole ont été élaborés par F. Jelinek et son équipe à IBM dans les années 70 [F. Jelinek 1976]. Les premiers travaux (années 80) se sont intéressés aux mots, et ce, pour des applications à vocabulaire réduit.

Au début des années 90, les systèmes de reconnaissance automatique de parole continue à grand vocabulaire et indépendants du locuteur ont vu le jour. La technologie s'est développée rapidement et, déjà vers le milieu des années 90, une précision raisonnable était atteinte pour une tâche de dictée vocale. Une partie de ce développement a été réalisée dans le cadre de programmes d'évaluation de la DARPA (Defense Advanced Research Projects Agency).

Différents systèmes de reconnaissance de la parole ont été développés, couvrant des domaines variés : reconnaissance de différents types de parole (téléphonique, continue, mots isolés, etc.), systèmes de dictée vocale, systèmes de commande et contrôle sur PC, systèmes de compréhension en langage naturel.

Les premiers travaux de reconnaissance de la parole ont essayé d'appliquer des connaissances expertes en production et en perception. De nos jours, les techniques de modélisation statistique apportent encore les meilleures performances.

Avant de présenter les principes généraux des différents modules constituant un système de reconnaissance automatique de la parole (voir Figure I.1) nous présentons d'abord les différents domaines d'application d'un système de reconnaissance automatique de la parole.

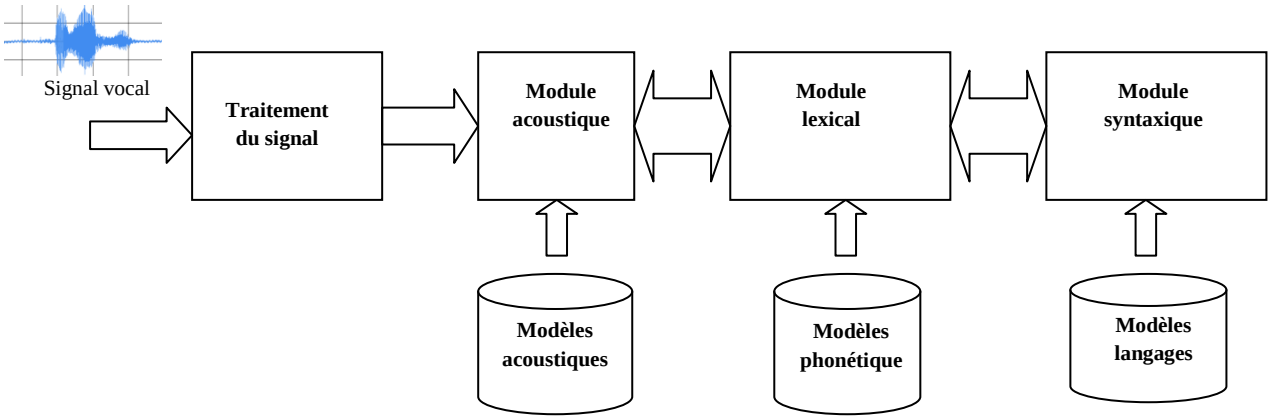


Figure I. 1 Architecture globale d'un Système de Reconnaissance de la parole

1.2. Domaines d'applications de la RAP

L'application de la RAP peut avoir plusieurs domaines, selon la nécessité ; en général la RAP est un outil d'allégement des taches de vie, en outre elle diminue la saisie à partir dun clavier dans la rédaction, les commandes de machine ou aussi la détection des maladies[1], selon [M .Chetouani 2004] on peut énumérer les domaines d'application en :

1.2.1. Dictée vocale

Le marché de l'informatique propose maintenant des logiciels sous Windows, capables, pour une modique somme à l'achat, de proposer des fonctionnalités de création de documents avec une seule interface parole. Le texte est dicté au lieu d'être saisi à partir d'un clavier.

1.2.2. Commandes de machines et contrôle de processus

Dans l'industrie, il n'est pas toujours possible de piloter les machines avec les moyens habituels (série de boutons, clic avec la souris, choix dans un menu, etc.), d'autant plus que les machines deviennent de plus en plus sophistiquées et que leur usage (ou leur processus de commande) devient de plus en plus complexe. La parole, grâce à un système de commande orale, peut s'avérer un mode de commande rapide et concis, en trouvant un espace de liberté supplémentaire à l'utilisateur.

[1] Exemple de la détection de l'enrouement dans la voix d'un patient

1.2.3. Reconnaissance automatique de la parole dans les télécoms

Depuis les années 90, nous sommes entrés dans un monde où l'usage des nouvelles technologies de l'information se développe de façon exponentielle. Le téléphone portable, devient un objet courant et traduit une volonté de nomadisme et de liberté pour l'usager. L'une des conséquences est que les technologies vocales sont maintenant sorties des laboratoires de recherche et permettent aux "providers" de services de les mettre à disposition de leurs clients pour faciliter l'usage des moyens de télécommunication : la composition automatique des numéros de téléphone, les serveurs vocaux pour la réservation des billets de transports ou pour la consultation de services bancaires, sont des exemples devenus maintenant courants.

1.2.4. Traduction automatique

Ce dernier type d'applications conjugue les nouvelles Technologies de la traduction automatique de la langue avec les technologies de la reconnaissance automatique et de la synthèse de la parole. Même si ces systèmes ainsi conçus ne permettent pas la traduction exacte au mot près, ils permettent d'aider au dialogue entre deux personnes de langue maternelle différente.

1.3. Les modèles de la RAP

1.3.1. Modèles acoustiques

1.3.1.1. Du signal aux vecteurs acoustiques

Le signal de parole ne peut être exploité directement. En effet, il contient non seulement le message linguistique, mais aussi de nombreux autres éléments comme des informations liées au locuteur, aux conditions d'enregistrement, etc. Toutes ces informations ne sont pas nécessaires lors du décodage de parole et rajoutent même du bruit. De plus, la variabilité et la redondance du signal de parole le rendent difficilement exploitable tel quel [S.Seng 2010].

Il est donc nécessaire d'en extraire uniquement les paramètres qui seront dépendants du message linguistique.

Généralement, ces paramètres sont estimés via des fenêtres glissantes sur le signal. Cette analyse par fenêtrage permet d'estimer le signal sur une portion jugée quasi-stationnaire, généralement 10 à 20 ms, en limitant les effets de bord et les discontinuités du signal via une fenêtre de Hamming. La majorité des paramètres représente le spectre fréquentiel et son évolution sur une fenêtre de taille donnée. Les techniques de paramétrage les plus utilisées sont PLP (Perceptual Linear Prediction dans le domaine spectral) [H.Hermansky 1991], LPCC Linear Prediction Cepstral Coefficients dans le domaine temporel) [J. Markel 1982] et MFCC (Mel Frequency Cepstral Coefficients dans le domaine cepstral).

1.3.1.2. Modélisation acoustique à base de modèles de Markov cachés

Pour la modélisation statistique acoustique, [F. Jelinek 1976] et [J. Baker 1975] introduisent les modèles de Markov cachés (Hidden Markov Model, HMM), qui sont aujourd'hui utilisés dans un très grand nombre des systèmes de reconnaissance automatique de la parole.

Chaque unité acoustique, en effet, est modélisée par un HMM. Dans le cas de petits lexiques, ces unités peuvent être les mots. Dans le cas de grands lexiques, l'unité préférée est le phonème (ou polyphone) ce qui limite le nombre de paramètres à estimer. Dans ce dernier cas, lors de la reconnaissance, les mots sont construits (dynamiquement) en termes de séquences de phonèmes et les phrases en termes de séquences de mots. C'est la méthode utilisée dans notre cas et on va la détailler dans le chapitre III.

1.3.1.3. Dictionnaire de prononciation

Le dictionnaire de prononciation (ou dictionnaire phonétique) fournit le lien entre les séquences des unités acoustiques et les mots représentés dans le modèle de langage. Il est important de noter que les performances du système de reconnaissance sont directement liées au taux de mots hors vocabulaire. Bien qu'un dictionnaire de prononciation créé manuellement permette une bonne performance, la tâche est très lourde à réaliser et demande des connaissances approfondies sur la langue en question. En plus, les noms propres sont l'un des problèmes majeurs pour toutes les langues. Par exemple, les 20 000 noms propres inclus dans le dictionnaire anglais COMPLEX ne représentent qu'une petite fraction des un à deux millions de noms rassemblés par [X. Huang 2001] sur des données en anglais US.

Pour résoudre ces problèmes, la littérature propose des approches qui permettent de générer automatiquement le dictionnaire de prononciation. Une des approches de la génération automatique d'un dictionnaire de prononciation consiste à utiliser des règles de conversion graphème-phonème. Cette construction nécessite une bonne connaissance de la langue et de ses règles de phonétisation, qui par ailleurs ne doivent pas contenir trop d'exceptions. Cette approche est donc applicable aux langues avec des prononciations assez régulières comme l'espagnol et l'italien. Pour l'anglais qui est très varié au niveau de la prononciation, l'approche automatique à base de règles n'est pas recommandée. Dans ce cas, les prononciations des mots hors vocabulaire d'une langue peuvent être générées en utilisant le décodage acoustico-phonétique de cette langue.

Alternativement, l'approche a été validée dans [J. Billa 2002] et [M. Bisani 2003]. Elle est simple et totalement automatique, et utilise des graphèmes comme unités de modélisation (dictionnaire de prononciation à base de graphèmes).

1.3.1.4. Décodage acoustico-phonétique

D'après [J. Haton et al. 1991], un décodage acoustico-phonétique (DAP) est défini comme la transformation de l'onde vocale en unités phonétiques ; c'est 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 pour lequel le niveau de représentation passe de continu à discret. Le décodage acoustico-phonétique est composé d'une première partie consistant à extraire les paramètres acoustiques et à les représenter sous forme de vecteurs acoustiques à partir du signal à décoder, 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

1.3.2. Modèles de langage

Un système de reconnaissance automatique de la parole continue à grand vocabulaire dépend généralement et fortement de la connaissance linguistique de la parole. Les meilleurs systèmes de décodage acoustico-phonétique qui n'utilisent aucun modèle de langage n'atteignent qu'un taux d'exactitude en phonèmes de l'ordre de 50 % environ. La modélisation du langage est donc une réelle nécessité pour la reconnaissance automatique de la parole continue à grand vocabulaire. Un module linguistique est nécessaire dans le système

pour déterminer la forme lexicale correspondante, c'est-à-dire la séquence de mots la plus probable, au sens langagier.

Dans l'équation bayésienne appliquée à la reconnaissance automatique de la parole apparaît une probabilité a priori de la séquence. Cette probabilité se calcule à partir d'un modèle de langage. Ainsi, la séquence « je suis ici » est plus probable, en terme de langage, que « jeu suis ici », ou encore «jeux suit y si », bien que l'acoustique soit quasi-similaire. Pour une même suite de phonèmes, il peut exister plusieurs dizaines de phrases possibles. Le rôle principal du modèle de langage est de les classer selon leur plausibilité linguistique [LÊ Việt 2006].

1.4. Évaluation des systèmes de reconnaissance de la parole

Les systèmes de reconnaissance de la parole sont évalués en terme de taux de mots erronés (WER : Word Error Rate). Généralement, il y a trois types d'erreurs sur les mots reconnus par le système de reconnaissance de la parole :

- substitution (Sub) ou remplacement du mot correct par un autre mot.
- suppression (Sup) ou omission d'un mot correct.
- insertion (Ins) ou ajout d'un mot supplémentaire.

Ces trois types d'erreur peuvent être calculés après alignement dynamique entre l'hypothèse du décodeur et une transcription de référence, [S. Seng 2010] à l'aide d'un calcul de distance d'édition minimale entre mots. Le résultat sera le nombre minimal d'insertions, de substitutions et d'élisions de mots, pour pouvoir faire correspondre les séquences de mots de l'hypothèse et de la référence. D'après sa définition, le WER peut être supérieur à 100 % à cause des insertions.

$$WER = \frac{nb\ de\ sub + nb\ de\ sup + nb\ de\ ins}{nb\ de\ mots\ corrects\ dans\ la\ référence} \qquad \text{I.1}$$

1.5. Outils existants

1.5.1. HTK

Hidden Markov Model Toolkit (HTK) est un ensemble d'outils portable permettant la création et la manipulation de modèles de Markov cachés. HTK est principalement utilisé dans le domaine de la recherche de la reconnaissance vocale bien qu'il soit tout à fait utilisable dans de nombreuses autres applications telles que la synthèse vocale, la reconnaissance de l'écriture ou la reconnaissance de séquences d'ADN.

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

1.5.2. Sphinx 4

Sphinx 4 est un logiciel de reconnaissance vocale écrit entièrement en Java. Les buts de Sphinx sont d'avoir une reconnaissance vocale hautement flexible, [Philippe Galley et al. 2006] d'égaler les autres produits commerciaux et de mettre en collaboration les centres de recherche de diverses universités, des laboratoires de Sun et de HP mais aussi du MIT.

Tout en étant hautement configurable, la reconnaissance de Sphinx 4 supporte notamment les mots isolés et les phrases (utilisation de grammaires). Son architecture est modulable pour permettre de nouvelles recherches et pour tester de nouveaux algorithmes.

La qualité de la reconnaissance dépend directement de la qualité des données vocales. Ces dernières étant les informations relatives à une voix propre. Ce sont par exemple les différents phonèmes, les différents mots (lexique), les différentes façons de prononciation. Plus ces informations ne seront importantes et connues par le système, meilleure sera sa réaction et ses choix à faire.

1.6. Conclusion

Dans ce chapitre on a fait une introduction à la RAP dans le but de spécifier une problématique concernant ce domaine, et dont elle va nous prouver que ce domaine est un domaine ouvert à toute innovation.

La RAP est une partie des deux parties de ce projet, elle intervient par la conception d'un modèle acoustique à base des chaines de Markov cachées, pour la reconnaissance des chiffres arabe, c'est vrai que ce thème on le trouve fréquemment dans les thèmes de la RAP réalisés par des gens d'origine arabes mais la spécification de notre projet est que la reconnaissance ce fait via un serveur vocal interactif et la plateforme utilisée est différente de celle utilisée dans la plupart de ces thèmes, ce qui augmente la difficulté de la réalisation.

Chapitre II : le traitement automatique du signal audio en vue de sa reconnaissance

2.1. Introduction

Bien comprendre les processus tels que le codage du son dans l'ordinateur ou sa synthèse, c'est d'abord bien comprendre le son lui-même. Nous commencerons donc par définir le son.

Qu'est ce que le son ?

Le son est une vibration de l'air. A l'origine de tout son, il y a mouvement (par exemple une corde qui vibre, une membrane de haut-parleur...). Il s'agit de phénomènes oscillatoires créés par une source sonore qui met en mouvement les molécules de l'air. Avant d'arriver jusqu'à notre oreille, ce mouvement se transmet entre les molécules à une vitesse de 331 m/s à travers l'air à une température de 20°C : c'est ce que l'on appelle la propagation[2].

Un son est d'abord défini par son volume sonore et sa hauteur tonale. Le volume dépend de la pression acoustique créée par la source sonore (le nombre de particules d'air déplacées). Plus elle est importante et plus le volume est élevé. La hauteur tonale est définie par les vibrations de l'objet créant le son. Plus la fréquence est élevée, plus la longueur d'onde est petite et plus le son perçu est aigu. En doublant la fréquence d'une note, on obtient la même à l'octave supérieure. Et donc, en divisant la fréquence par deux, on passe à l'octave inférieure. Ce n'est qu'au-delà de 20 vibrations par seconde que l'oreille perçoit un son. Les infrasons, de fréquence inférieure à cette limite de 20 Hz sont inaudibles, de même que les ultrasons, de fréquence supérieure à 20 000 Hz, soit un peu plus de 10 octaves. Chacun peut constater que le niveau sonore diminue à mesure que l'on s'éloigne de la source. Cette diminution est la même chaque fois que la distance est doublée. Cependant, les hautes fréquences ne se propagent pas aussi loin que les sons graves. Il faut plus d'énergie pour restituer les basses que les aigus.

2.2. Production de la parole [3]

Le signal de parole est provoqué par des mécanismes complexes issus de plusieurs sources. Les sons de la parole se produisent normalement lors de la phase de l'expiration grâce à un flux d'air contrôlé, en provenance des poumons et passant par la trachée-artère (=conduit respiratoire). Ce flux d'air s'appelle « air pulmonaire (ou pulmonique) égressif ». Il va

[2] http://jarrologie.free.fr/pratique/mao.pdf
[3] Cours (CM) Lolke J. Van der Veen , Université Lyon2

rencontrer sur son passage plusieurs obstacles potentiels qui vont le modifier de manière plus ou moins importante

2.2.1. Un premier obstacle

Après passage par la trachée-artère, le flux entre dans un conduit cartilagineux, appelé le « larynx ».

Le larynx se compose de 4 cartilages différents, dont le cartilage thyroïde (‘pomme d’Adam’) et l’épiglotte (cartilage en forme de lame, pouvant fermer par un mouvement de bascule en arrière l’entrée du larynx afin d’empêcher le bol alimentaire d’entrer dans le larynx et la trachée-artère).

Le larynx peut se déplacer vers le bas ou vers le haut (cf. le mouvement de la pomme d’Adam chez certains sujets masculins). De ce fait, la longueur de la cavité pharyngienne (située juste au-dessus) peut se trouver modifiée.

A l’intérieur du larynx se situent les « cordes vocales », des organes vibratoires constitués de tissu musculaire et de tissu conjonctif résistant. Les cordes vocales sont reliées à l’avant au cartilage thyroïde. Elles peuvent s’écarter ou s’accoler. L’espace entre les cordes vocales est appelé « glotte ».

Positions des cordes vocales :

- inspiration profonde (écartement maximal) ;
- respiration normale (écartement moyen) ;
- voisement (= phonation) (accolées) ;
- chuchotement (partiellement accolées).

Les cordes vocales constituent une source vibratoire en puissance. Au passage de l’air, les cordes vocales peuvent se mettre à vibrer à condition d’être suffisamment rapprochées et relâchées (tension faible). L’état vibratoire s’appelle voisement (ou phonation). Ce dernier accompagne de nombreux sons de la parole (= sons voisés). Le mouvement vibratoire correspond à une succession plus ou moins rapide de cycles d’ouverture et de fermeture de la glotte. (Nombre de cycles par seconde = fréquence.) La hauteur mélodique dépend également de ce mouvement ainsi que de quelques autres facteurs physiologiques (longueur et tension des cordes).

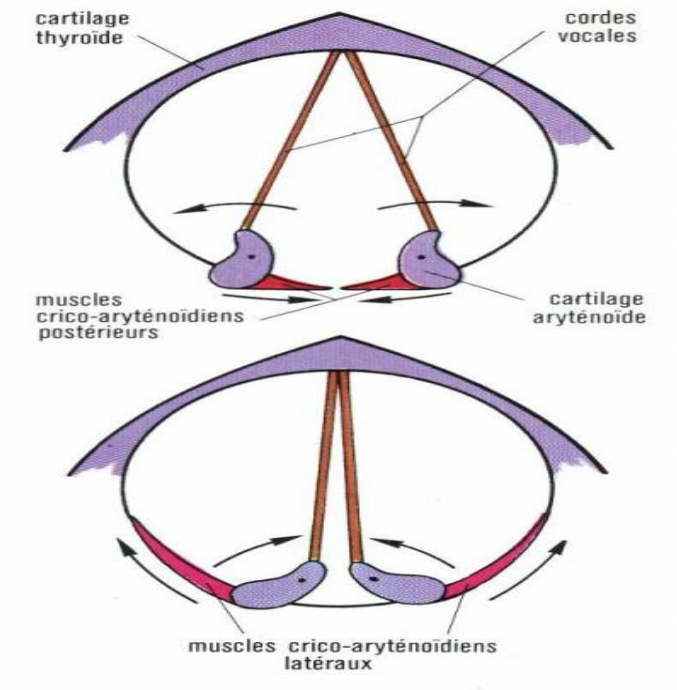


Figure II. 1 Schéma montrant le rapprochement et l'écartement des cordes vocales

2.2.2. Le conduit vocal

L'aire laryngée passe dans le conduit vocal qui comprends plusieurs cavités parmi elles on trouve le pharynx qui intervient dans la déglutition et la respiration et la phonation et l'audition ; les cavités nasales qu'ont comme rôle de réchauffer et assainir l'air inhalé. De plus, elles contiennent les organes impliqués dans l'olfaction ; la bouche (cavité buccale) dans cette cavité se situent des articulateurs, certains fixes (= passifs), d'autres mobiles (= actifs) ; et les lèvres (cavité buccale) une cavité que l'on crée lorsqu'on projette en avant les lèvres

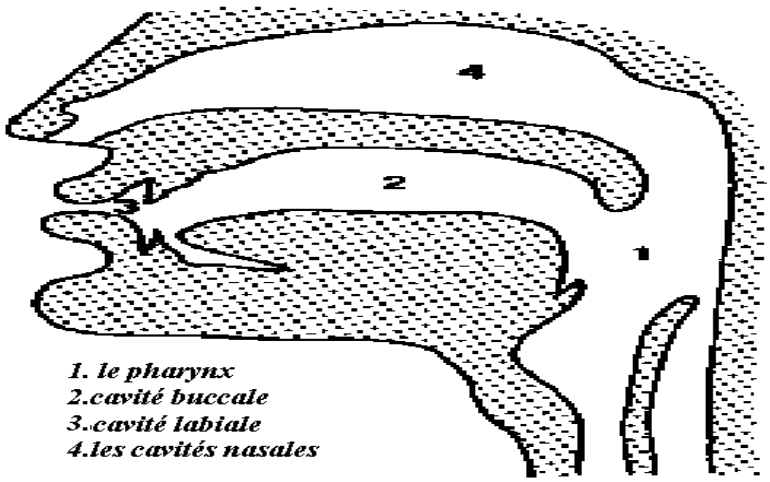


Figure II. 2 Appareil phonatoire

2.3. Caractéristiques phonétiques

2.3.1. Phonème

La plupart des langues naturelles sont composées à partir de sons distincts, les phonèmes. Un phonème est la plus petite unité présente dans la parole [Anne 2007]. Le nombre de phonèmes est toujours très limité (normalement inférieur à cinquante) et ça dépend de chaque langue. Les phonèmes peuvent être classés en fonction de trois variables essentielles : le voisement (activité des cordes vocales), le mode d'articulation (type de mécanisme de production) et le lieu d'articulation (endroit de resserrement maximal du conduit vocal)

2.3.1.1. Voyelles

Les voyelles sont des sons voisés qui résultent de l'excitation du conduit vocal par des impulsions périodiques de pression liées aux oscillations des cordes vocales. Chacune des voyelles correspond à une configuration particulière du conduit vocal. Les voyelles se différencient principalement les unes des autres par leur lieu d'articulation, leur aperture, et leur nasalisation. On distingue ainsi les voyelles antérieures, moyennes et postérieures, selon la position de la langue, et les voyelles ouvertes et fermées, selon le degré d'ouverture du conduit vocal.

Il y a deux types de voyelle : les voyelles orales (a,i,e, u, ...) qui sont émises sans intervention de la cavité nasale et les voyelles nasales(ã, ɛ~, …) qui font intervenir la cavité nasale.

2.3.1.2. Consonnes

Les consonnes sont des sons qui sont produits par une turbulence créée par le passage de l'air dans une constriction du conduit (les consonnes non voisées) ou une source périodique liée à la vibration des cordes vocales s'ajoute à la source de bruit (les consonnes voisées). Il y a trois types de consonnes : les fricatives(ou constrictives), les occlusives et les nasales.

2.4. Les caractéristiques du son :

Le son est défini par trois paramètres acoustiques : l'intensité, la fréquence et la durée.

L'intensité :

Elle est définie par le Dictionnaire d'Orthophonie (2004) comme «la puissance du son, de la voix, mesurée en décibels (dB) grâce à un sonomètre ».

La fréquence :

Elle correspond au « nombre de vibrations par seconde d'un son pur périodique déterminant sa hauteur physique » (Ibid.). Plus la fréquence d'un son est élevée, plus ce son est aigu. La fréquence se mesure en Hertz (Hz).

La durée :

La durée se définit comme l'intervalle séparant deux évènements. Selon Lienard (cité par [G. Laboulais 2007], un événement sonore dure au moins soixante millisecondes pour être perçu par l'oreille.

Ces trois paramètres composent le signal acoustique. Leur traitement par le système auditif permet notamment de reconnaître et d'identifier une source sonore, de la localiser dans l'espace, de décoder la parole et d'en analyser la prosodie [G. Laboulais 2007]. Une représentation courante est l'amplitude de l'onde en fonction du temps :

Figure II. 3 représentation bidimensionnelle de son.

2.4.1. Comment stocker le son sur l'ordinateur :[4]

Il faut d'abord différencier les deux types de sons : le son analogique et le son numérique.

Le son analogique est représenté sous la forme de signaux électriques d'intensité variable. Ces signaux sont issus d'un micro qui transforme le son acoustique d'une voix ou la vibration des cordes d'une guitare en impulsions électriques. Ces signaux sont enregistrables tels quels sur une bande magnétique (K7 audio par exemple) et peuvent être ensuite amplifiés, puis retransformés en son acoustique par des haut-parleurs. Le son analogique n'est pas manipulable tel quel par un ordinateur, qui ne connaît que les 0 et les 1.

Le son numérique est représenté par une suite binaire de 0 et de 1. L'exemple le plus évident de son numérique est le CD audio. Le processus de passage du son analogique en son numérique est appelé "échantillonnage". Celui-ci consiste à mesurer la tension (en Volt) du signal analogique à intervalles réguliers. La valeur obtenue est enfin codée en binaire (suite de 0 et de 1).Le composant qui réalise cette tache est appelée convertisseur A/N. Évidemment, ce processus de mesure et de conversion binaire doit être très rapide. C'est là qu'intervient la fréquence du son à numériser. Par exemple, pour une voix dont la fréquence est de 8000 Hz (Hertz), le signal électrique issu du micro aura aussi une fréquence de 8000 Hz. Pour transformer ce signal en numérique et à qualité équivalente, le mathématicien Shannon a démontré qu'il fallait que le prélèvement de mesures soit fait à une fréquence au moins 2 fois plus rapide que la fréquence originale, soit pour l'exemple de la voix, 16000 fois par seconde (16000 Hz ou 16 kHz). Un autre paramètre très important de l'échantillonnage est la précision avec laquelle la tension du signal électrique sera lue et codée. Le codage peut, en effet se faire sur 2 n bits. Une précision de 8 bits donnera une tension codée parmi 256 valeurs possibles, alors que 16 bits donneront 65 536 valeurs. Les CD sont ainsi échantillonnés à 44,1 kHz sur 16 bits.

La conversion inverse de numérique vers analogique, se fait par le processus inverse. La tension du signal électrique est recréée à partir des valeurs codées, lues à la même vitesse qu'elles avaient été enregistrées. L'avantage évident de ce type de son, c'est qu'étant codé sous

[4] http://jarrologie.free.fr/pratique/mao.pdf

la forme de 0 et de 1, il est directement manipulable par un ordinateur et son stockage ne pose aucun problème sur un disque dur. En revanche, le nombre de valeurs enregistrées étant énorme (44 100 valeurs/s), ce type de son occupe beaucoup de place dans la mémoire ainsi que sur le disque dur de l'ordinateur. A titre d'exemple, un CD audio de 74 minutes représente 650 Mo (Mégaoctets) et le débit d'un lecteur de CD est de 150 Ko/s (Kilo-octets/s). Une seule seconde de son stéréo échantillonné à 44,1 kHz et en 16 bits prend 172 Ko.

2.4.2. Qu'est-ce qu'un fichier audio numérique.

La reconnaissance vocale se base sur la comparaison de fichiers audio ainsi nous devons tout d'abord maîtriser le format d'enregistrement utilisé avant d'effectuer des opérations de transformation du signal.

On distingue deux types de format, les formats compressés et les formats non compressés.

WAV (format largement utilisé depuis l'apparition de Windows). La norme de qualité du format WAV est 44.100 KHz à 16 Bits/s qui correspond à l'échantillonnage des CDs Audio. Sa structure est très simple, et elle est comme suit :

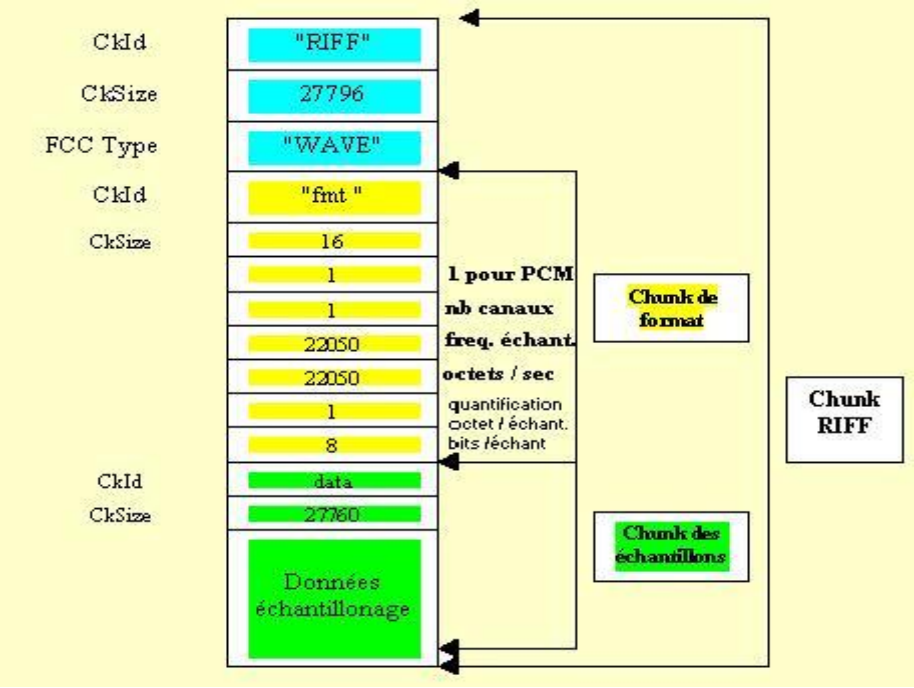


Figure II. 4 Architecture d'un fichier wav

Le fichier est constitué de blocs hiérarchisés appelés "chunk". Le bloc "Riff " ("Ressource Interchange File Format", qui est le type des fichiers WAV) englobe l'ensemble du fichier, il permet de l'identifier comme étant un fichier WAV. Le bloc Format identifie les différents paramètres du format: fréquence de l'échantillonnage, nombre de bits etc. Le bloc data contient les données échantillonnées. Tous les champs sont facilement compréhensibles avec les notions vues précédemment, excepté la valeur du champ correspondant au format Microsoft qui est à 1 pour le format PCM, Pulse Code Modulation, format des données audio non compressés.

Les données du son, les échantillons, sont alignés les uns après les autres dans l'axe des temps (dans l'ordre où ils arrivent dans le temps). En stéréo, les canaux sont multiplexés (entrelacés). Ce multiplexage offre deux avantages : lecture/écriture des deux canaux en une seule opération disque (évite surtout les déplacements de têtes du disque) et l'augmentation aisée du nombre de canaux tout en restant dans la norme du format (par exemple un format quadriphonique).

2.5. Le traitement automatique du signal

L'analyse des données issues du signal parole est très complexe, ceci est dû à la multitude ou la redondance de l'information, c'est l'une des particularités de ce signal, en prenant en compte les informations citées auparavant, l'analyse sera orientée à des paramètres généraux discriminants, englobant différentes occurrences de l'information du coté temporel, spectral, spatial, perceptuel et/ou prosodique,

2.5.1. Prétraitement

Un prétraitement consiste à :

- Une conversion analogique numérique.
- Une préaccentuation.
- Une segmentation et un chevauchement.
- Un fenêtrage.

2.5.1.1. La conversion analogique numérique

Le traitement de la parole suppose toujours en premier lieu une analyse du signal vocal converti au préalable en signal électrique par un microphone ; puisque les ordinateurs ne peuvent pas manipuler des sources analogiques, on doit convertir les signaux au format numérique avec un convertisseur (A/N) le processus inverse sera fait par le convertisseur (N/A).[5]

2.5.1.2. Segmentation et chevauchement

Après l'acquisition et le stockage du corpus d'étude suivant un codage bien défini, chaque mot ou phrase du signal enregistré est segmenté en fenêtres de durée fixe, obéissant aux deux contraintes suivantes :

- Stationnarité du signal parole (moyenne et variance constantes durant la trame ou fenêtre temporelle d'analyse)
- Durée supérieure à l'inverse de la fréquence fondamentale [Y.laprie 2002].

Travaux	Durée	Chevauchement
Rabiner en 1989 [14]	45 ms	30 ms
Ahmed en 1998 [15]	20 ms	5 ms
Yasuhi en 2002 [16]	20 ms	10 ms
Alizera en 2002 [17]	10 ms	2.5 ms
Stemmeren 2001 [18]	20 ms	10 ms

Table II. 1 Durée d'analyse primaire ainsi que la durée de chevauchement

2.5.1.3. Préaccentuation :

Il y a deux explications pour justifier l'utilisation du module de préaccentuation. Pour la première, la partie voisée du signal de la parole présente une accentuation spectrale approximative de -20 dB par décade. Le filtre de préaccentuation permet de compenser cette accentuation avant d'analyser le spectre, ce qui améliore cette analyse. La deuxième considère

[5] http://users.dsic.upv.es/~jorallo/escrits/MEMOIRE.pdf

que l'audition est plus sensible dans la région du spectre autour de 1 KHz. Le filtre de préaccentuation va donc amplifier cette région centrale du spectre» [M .Chetouani 2004].

Le filtre utilisé pour la préaccentuation, est un filtre numérique de premier ordre :

$$H(z) = 1 - \alpha \quad z^{-1} \qquad \text{II.1}$$

Le facteur de préaccentuation est pris entre 0.9 et 1 (souvent 0.95).

2.5.1.4. Fenêtrage

Le découpage du signal en trames produit des discontinuités aux frontières des trames, qui se manifestent par des lobes secondaires dans le spectre ; ces effets parasites sont réduits en appliquant aux échantillons de la trame une fenêtre de pondération comme par exemple la fenêtre de Hamming» [M. Chetouani 2004].

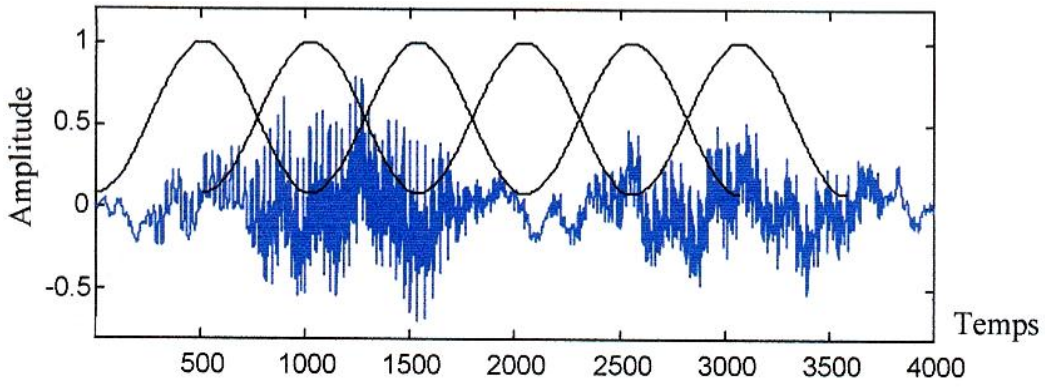


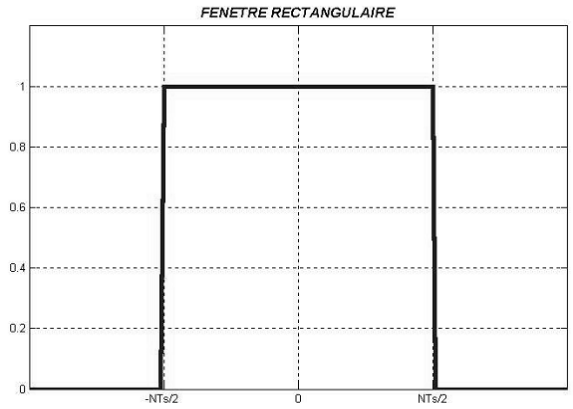
Figure II. 5 Le recouvrement des fenêtres dans le temp

Il existe plusieurs types de fenêtres d'analyse, comme la fenêtre rectangulaire, Hamming , triangulaire, Hanning et d'autres on présente les deux premiers :

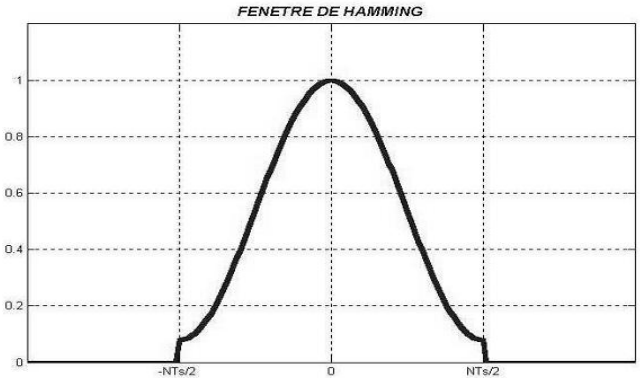
Fenêtre rectangulaire : C'est la troncature simple, son lobe central est de largeur $2\Delta f = \frac{2}{NT_0} = \frac{2}{NT_S}$.[6] Et l'amplitude du premier lobe de l'ordre de 22%.

[6] $2\Delta f$ *largeur d'une raie*
T_0 *Temps total.*
T_s *Temps d'échantillonnage.*

L'effet des lobes latéraux se met en évidence sur le traitement d'un signal composé de deux raies théoriquement résolues mais d'amplitudes de rapport 10. La TFD donne le résultat ci contre où la "petite" raie est non détectable.[7]



Fenêtre de Hamming : Dite en "cosinus rehaussé", son lobe central est de largeur $2\Delta f = \frac{4}{NT_s}$ et ses lobes latéraux d'amplitude relative inférieure à 1%.[7]



2.5.1.5. 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 [M. Bellanger 1980].

$$X_n = \frac{1}{N} \sum x(k) e^{-j2\pi(n/N)} \qquad \text{II.2}$$

[7] Cours Traitement Numérique du Signal « Digital Signal Processing ». M. Frikel . École nationale supérieure d'ingénieur de Caen 2011/2012.

L'équation II.2 donne le calcul de la FFT pour une séquence X(n) comportant N échantillons.

Dans le cas d'une séquence d'échantillons, il est alors possible de calculer une Transformée de Fourier Discrète (TFD, Discret Fourier Transform-DTF-enanglais).

En 1965, [Cooley et tukey 1965] 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.

2.6. Analyse et traitement de la parole

Il existe plusieurs méthodes d'analyse qui sont utilisés en vue de reconnaître la parole ; comme l'analyse cepstral, l'analyse temporel et l'analyse spectrale.

2.6.1. Analyse temporel

L'ordinateur appréhendait un signal sonore dans ses formes temporelles ou fréquentielles ,qui ne sont pas les plus adéquates pour la reconnaissance de la parole. Il est nécessaire de calculer plusieurs paramètres [Vaufreydaz 2002] dérivés de ce signal.

Nous n'aborderons ici que les principaux utilisés dans la littérature :

- **Energie du signal**

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 [J. Taboada et al. 1994]. La formule de calcul de ce paramètre est :

$$E_N = \sum_{n=0}^{N-1} S(n)^2 \qquad \text{II.3}$$

S (n) : signal parole échantillonné

- **Taux de passage par zéro**

Le taux de passage par zéro (zero crossing rate) représente le nombre de fois que le signal, dans sa représentation amplitude/temps, passe par la valeur centrale de l'amplitude

(généralement zéro). Il est fréquemment employé pour des algorithmes de détection de section voisée/non voisée dans un signal.

La formule du (bond-crossing) proposée par [J. Taboada et al. 1994] pour chaque fenêtre analysée est donc :

$$\sum_{N=0}^{N-1} |f(n) - f(n-1)|$$

$$\text{Avec} \quad fn = \begin{cases} 1 & \text{si } n > S \\ f(n-1) & \text{si } -S < n < S \\ -1 & \text{si } n < -S \end{cases} \qquad \text{II.4}$$

Où S est un seuil d'amplitude permet de définir une zone autour du zéro de largeur 2xS au sein de laquelle les oscillations ne sont pas prises en compte.

Cette mesure se montre très intéressante, dans le cadre d'une détection de parole en amont d'un système de reconnaissance, pour la détection de fricative en fin de signal à reconnaître ou d'attaque de Plosive [M. Aubry 2000].

- **Premier coefficient d'autocorrélation [J. RACHEDI]**

L'autocorrélation est la convolution du signal avec lui même. C'est une méthode couramment utilisée pour déterminer la fréquence fondamentale. Le résultat d'une autocorrélation est dans le cas d'un son voisé (possédant une f0) une suite de lobes espaces de n_0 échantillons. La fréquence fondamentale peut être déterminée en prenant l'inverse de la distance entre les deux premiers lobes de la deuxième moiti´e de la courbe. Le premier coefficient correspond `a la corrélation de deux ´échantillons consécutifs du signal. Un échantillon d'écart correspond `à une très haute fréquence. Or un bruit contient beaucoup plus d'´energie dans les tr`es hautes fréquences qu'un son voisé.

Deux échantillons consécutifs sont donc fortement décorellés dans le cas de bruit blanc. Ainsi, ce coefficient donne une information très fiable du rapport signal/bruit, pour de tr`es faibles temps de calculs. Pour un signal x de longueur N, sa formule normalisée en énergie est donnée par :

$$Fac(x) = \frac{\sum_{n=0}^{N-2} x_n . x_{n+1}}{\sqrt{\sum_{n=0}^{N-2} x_n^2} \sqrt{\sum_{n=0}^{N-2} x_{n+1}^2}} \qquad \text{II.5}$$

- **Analyse cepstrale**

L'analyse cepstrale résulte du modèle de production, son but est d'effectuer la déconvolution (source/ conduit) par une transformation homomorphique: les coefficients sont obtenus en appliquant une transformée de Fourier numérique inverse au logarithme du spectre d'amplitude[8]. Le signal ainsi obtenu est représenté dans un domaine appelé cepstral ou quérulent ; les échantillons se situant en basses fréquences correspondent à la contribution du conduit vocal et donnent les paramètres utilisés en RAP, tandis que la contribution de la source n'apparaît qu'en hautes fréquences.

- Lorsque le spectre d'amplitude résulte d'une FFT sur le signal de parole prétraité, lissé par une suite de filtres triangulaires répartis selon l'échelle Mel, les coefficients sont appelés Mel Frequency Cepstral Coefficients (MFCC).

$$M_{mel} = \mathrm{x}.\ \log\left(1 + \frac{\mathrm{f_h}}{\mathrm{y}}\right) \qquad \text{II.6}$$

x=l000/log (2) et y=l000. [Calliope 1989]

De nos jours, les valeurs les plus couramment utilisées sont celles de [S. Umesh 1999]

x=2595 et y=700.

$$\text{Donc :} \quad B(f) = 2595.\log\left(1 + \frac{f}{700}\right) \qquad \text{II.7}$$

Ou f est la fréquence en Hz et B (f) la fréquence suit l'échelle de Mel

[8] http://rsa.esigetel.fr/Doc/Supports_Rsa/Information/parole/speech2002/%C9tude%20Du%20signal.pdf

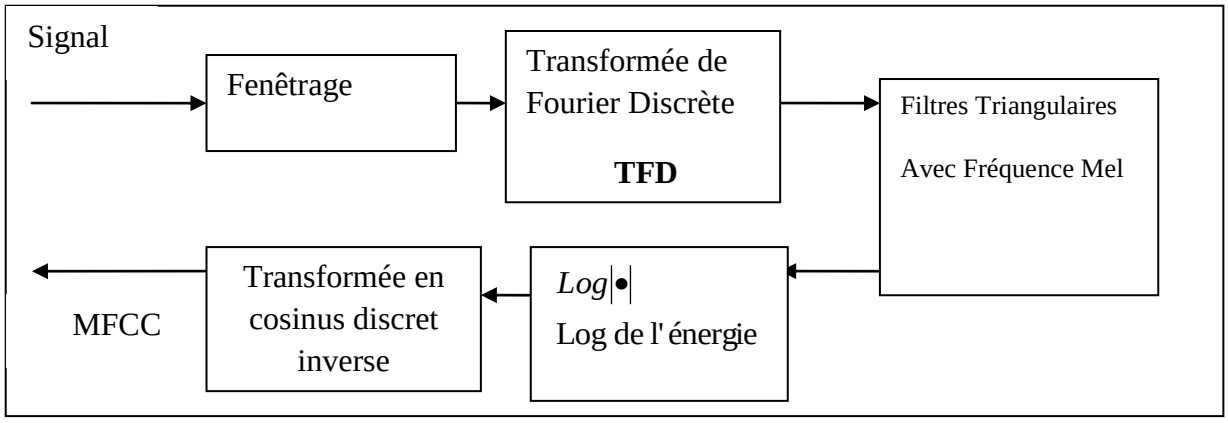


Figure II.6 chaîne d'analyse du signal produisant les coefficients MFCC (M. Chetouani 2004.)

2.7. Conclusion

Dans ce chapitre on a fait une étude, qui suit le son depuis sa production jusqu'à son analyse, dans le but de générer des vecteurs acceptant l'application de la méthode HMM pour la reconnaissance du mot prononcé.

Les difficultés rencontrées dans cette partie commence par l'acquisition de son, et sa transformation de la représentation temporelle à la représentation fréquentielle, et comme ces deux représentations ne sont pas les plus adéquates pour la reconnaissance de la parole surtout la parole continue il était nécessaire de calculer plusieurs paramètres dérivés de ce signal. Et sans oublier que la voix est transmise d'un pc à un autre via un réseau IP qui nous oblige de veiller à la crédibilité du fichier reçu.

Comme cette partie est la base de notre travail ses résultats seront entièrement consacrés à la partie suivante (le processus de reconnaissance).

Chapitre III : Les modèles de Markov cachés

3.1. Introduction

La reconnaissance automatique de la parole (RAP) est un processus qui converti une forme en ondes (waveform) en un mot ou en une séquence de mots « texte » ; elle est utilisée généralement dans les logiciels de dicté, ou les transcriptions médicales..

Le problème au quel est confronté à la RAP est l'incertitude, due aux locuteurs et autres variables liées à l'environnement telles que le dispositif d'acquisition, l'environnement d'appel qui peut avoir de bruit, et autres. Toutes ces variables doivent être modélisées pour permettre la RAP. En utilisant un modèle probabiliste ; le meilleur et le plus populaire modèle utilisée jusqu'à nos jours ce sont les modèles basés sur les chaines de Markov cachées.

Le modèle de Markov cachée est un outil statistique très efficace pour la modélisation des séquences génératives [A. Markov 1913], il a trouvé ses applications dans plusieurs domaines, spécifiquement dans le traitement de signal, et particulièrement le traitement de la parole, aussi l'analyse documentaires « tel que la recherches des informations clés dans un ensemble de documents », et le traitement d'image « reconnaissance des formes ou des objets ».

Dans ce chapitre on commence par une présentation de la théorie des chaines de Markov et nous nous étendons ensuite aux modèles de Markov Cachés. Rappelons que le but de l'utilisation des HMM dans notre cas est de connaitre des séquences d'observations qui sont à l'origine un signal audio reçu à partir d'un dispositif d'acquisition simple tel qu'un microphone ou un softphone.

3.2. Le modèle de Markov discret

Une chaine de Markov est un processus aléatoire (suite de variables aléatoires) dont la distribution conditionnelle de probabilité de l'état présent ne dépend que de l'état qui le précède[9]. Une chaine de Markov est dite discrète si les variables aléatoires qui représentent les états de la chaine (processus, système) sont des variables aléatoires discrètes (prennent leurs valeurs dans un espace d'´etats discret).

Considérons un système qui peut être dans N états différents $S_1;S_2;\ldots;S_N$

[9] Chaine de Markov d'ordre 1.

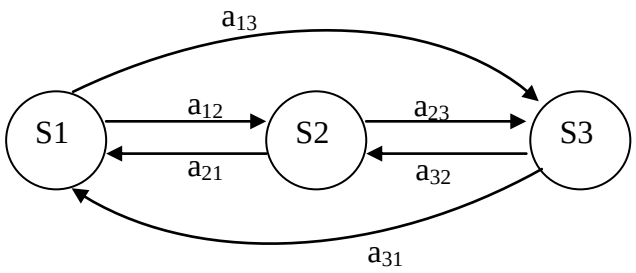


Figure III. 1 Un exemple de chaine de Markov à 3 états S1, S2, S3

Les flèches représentent les transitions d'un état à un autre, les $[a_{ij}]$ où $1 \leq i;j \leq 3$ sont les probabilités de passage d'un état *i* vers un état *j*

Le passage d'un état à un autre est régi par des probabilités. A un instant *t*, le système se trouve dans un état S_t. Dans le cas des chaînes de Markov d'ordre un, la description stochastique du modèle s'arrête à l'état courant et l'état précédent :

$$P(s_t = S_j | s_{t-1} = S_i, s_{t-2} = S_k, \dots) = P(s_t = S_j | s_{t-1} = S_i) \qquad III.1$$

De plus, nous considérons les chaînes de Markov dont les probabilités de transition a_{ij} d'un état *i* à un état *j* sont indépendantes du temps :

$$P(s_t = S_j | s_{t-1} = S_i) = a_{ij}, 1 \leq i, j \leq N \qquad III.2$$

Les coefficients a_{ij} obéissent aux contraintes suivantes :

$$\begin{cases} a_{ij} \geq 0 & III.3 \\ \sum_{j=1}^{N} a_{ij} = 1 & III.4 \end{cases}$$

Qui forment une matrice appelée matrice de transition : $A = [a_{ij}]_{1 \leq i,j \leq N}$ *III.5*

A l'instant initial *t = 1*, on doit définir pour chaque état *i* une probabilité dite probabilité initiale

$$\pi_i = P(s_1 = S_i) \qquad III.6$$

Et l'ensemble de ces probabilités forme la matrice des probabilités initiales :

$$\boldsymbol{\pi} = [\pi_i]_{1 \le i \le N} \qquad III.7$$

Ce processus stochastique est un modèle de Markov observable car la sortie du processus est une séquence d'états où chaque état correspond à un événement physique observable [Rabiner 1989].

3.3. Les Modèles de Markov Cachés

Definition: La Chaine de Markov et les Modèles de Markov Cachés

- Une chaîne de Markov est un automate de *L* états discrets.
- Un processus de Markov est un système à temps discret se trouvant à chaque instant dans un état pris parmi *L* états distincts. Les transitions entre les états se produisent entres deux instants discrets consécutifs, selon une certaine loi de probabilité.
- Les Modèles de Markov Cachés diffèrent des modèles discrets par la distribution des probabilités sur l'espace des observations possibles, ces états ne sont plus alors observés directement mais dits cachés.

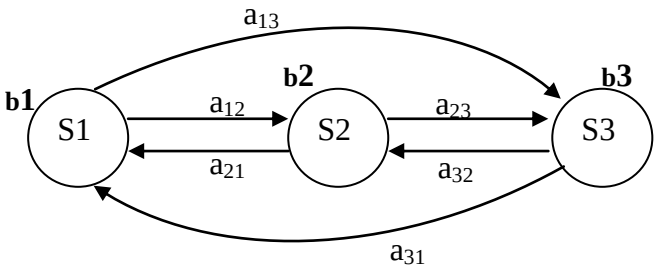


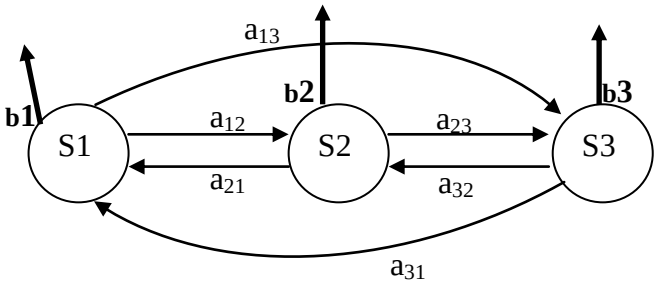
Figure III. 2 Un exemple de chaine de Markov à 3 états S1, S2, S3

Les flèches représentent les transitions d'un état à un autre, les $[a_{ij}]$ où $1 \le i;j \le 3$ sont les probabilités de passage d'un état i vers un état j et les b_i sont les probabilités d'émission.

Un modèle de Markov caché (HMM) est représenté de la même façon qu'un modèle de Markov discret, un ensemble de séquences d'observations dont l'état de chaque observation n'est pas observé, mais associé à une fonction de densité de probabilité.

Un modèle de Markov caché est un processus doublement stochastique, dans lequel les observations sont une fonction aléatoire de l'état et dont l'état change à chaque instant en fonction des probabilités de transition issues de l'état antérieur [J. Bruno 1995].

Et de même que les modèles de Markov discrète les modèles de Markov cachés se caractérisent par des propriétés dans la globalité sont identique aux propriétés des modèle de Markov discrètes ajoutant une matrice dite matrice d'émission.



✓ La matrice de transition : $A = [a_{ij}]_{1 \le i,j \le N}$

$$A = \begin{pmatrix} a11 & \cdots & a13 \\ \vdots & \ddots & \vdots \\ a31 & \cdots & a33 \end{pmatrix} \qquad III.8$$

Où la somme des a_{ij} est égale à un.

✓ Le vecteur des probabilités initiales :

$$\pi = \begin{bmatrix} P(s_1 = S_1) \\ P(s_1 = S_2) \\ P(s_1 = S_3) \end{bmatrix} \qquad III.9 \quad .$$

La somme des $P(s_1 = S_i)$ est égale à un.

✓ La matrice d'émission :

$$B = \begin{bmatrix} b_1(o_t) \\ b_2(o_t) \\ b_3(o_t) \end{bmatrix} \qquad III.10$$

Où $\boldsymbol{b_1(o_t)}$ est **la** probabilité d'apparition de l'observation *t* sachant que le système se trouve dans l'état un.

A partir de cet exemple on peut caractériser un MMC par les trois matrices (A, B etπ), [Rabiner 1989] et on le note :

$$\lambda = (A, B, \pi) \qquad III.11$$

Dont les formes généralisées de leurs matrices sont :

La matrice de transition généralisée $A = \begin{pmatrix} a11 & \cdots & a13 \\ \vdots & \ddots & \vdots \\ ai1 & \cdots & aij \end{pmatrix}$ *III.12*

La matrice d'émission généralisée $B = \begin{bmatrix} b_1(o_1) & \cdots & b_1(o_t) \\ \vdots & \ddots & \vdots \\ b_i(o_1) & \cdots & b_i(o_t) \end{bmatrix}$ *III.13*

$$b_i(o_t) = P(x_t \,|s_t = S_i) \;\; o\text{ù } 1 \leq i \leq \boldsymbol{N} \;, 1 \leq t \leq \boldsymbol{T}$$

- ✓ N est le nombre d'états dans le modèle
- ✓ T est le nombre d'observations distinctes dans chaque état

La matrice des probabilités initiales généralisée $\pi = \begin{bmatrix} P(s_1 = S_1) \\ \vdots \\ P(s_1 = S_i) \end{bmatrix}$ *III.14*

3.3.1. Exemple 1 :

Etant donnée un MMC défini par trois état {1, 2,3} et par une matrice de transition

$$T = \begin{matrix} 0.3 & 0.2 & 0.5 \\ 0.6 & 0.1 & 0.3 \\ 0.2 & 0.4 & 0.4 \end{matrix}$$

Et une matrice d'émission

$$B = \begin{matrix} 0.6 & 0.2 & 0.2 \\ 0 & 0.5 & 0.5 \\ 0.3 & 0 & 0.7 \end{matrix}$$

Et une matrice des probabilités initiales

$$\pi = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

La représentation graphique de ce modèle est la suivante :

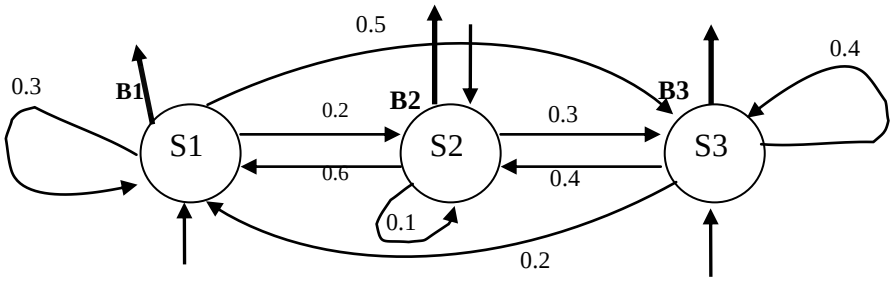


Figure III. 3 Automate d'états pour la chaine de Markov cachée de l'exemple

3.3.2. Exemple de modèle de l'urne et des boules[10]

Supposons qu'il existe *N* urnes dans une pièce. Dans chaque urne il y a un grand nombre de boules colorées. Supposons qu'il y a *M* couleurs de boules distinctes.

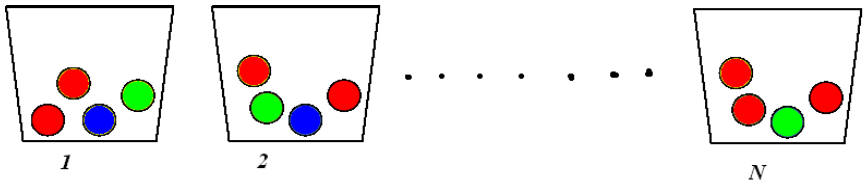


Figure III. 4 L'exemple de l'urne et des boules

$P(Vert) = b_1(1)$ $P(Vert) = b_2(1)$ $P(Vert) = b_3(1)$

$P(Blue) = b_1(2)$ $P(Blue) = b_2(2)$ $P(Blue) = b_3(2)$

$P(Rouge) = b_1(M)$ $P(Rouge) = b_2(M)$ $P(Rouge) = b_N(M)$

Les étapes pour générer une séquence d'observations pour cet exemple est les suivantes :

1- Une personne est dans la pièce et choisit une urne initiale $q_1=i$ comme un état initial, on l'attache π comme distribution.

[10] Le modèle de l'urne et des boules, introduit par Jack Ferguson et ses collègues (Rabiner 1989)

2- On initiale t=1 (l'indice des observations).

3- Dans cette urne, une boule est choisie aléatoirement et le tirage est enregistré comme première observation (avec laquelle sa probabilité d'occurrence est$b_i(o_t)$).

4- La boule est ensuite replacée dans l'urne de laquelle elle a été tirée.

5- Une nouvelle urne est ensuite choisie

6- Incrémenter t, t = t+1, et retourner à l'étape 3 tan que t≤ T; si non on arrête la procédure.

La question posée est la suivante. Si on choisit une observation (une boule tirée à $t=t_i$), et on demande à une personne qui n'est pas dans la pièce de nous dire cette boules est tirée de quelle urne comment puerait-elle nous répondre ? Pour que cette personne puerait répondre à notre question il faut qu'elle utilise une CMC soit d'ordre *0* ou d'ordre *1* ou d'ordre *2* …etc., (c.-à-d. Ordre *1* cette personne doit connaître l'observation précédente était dans quel état, et d'ordre *2* elle doit connaître les deux observations précédentes étaient dans quels états …etc.).

3.3.3. Les types des MMCs

La structure d'un MMC est définie par la matrice de transition, A. En général la structure des MMCs est ergodique ou fortement connexe Figure III.5.a

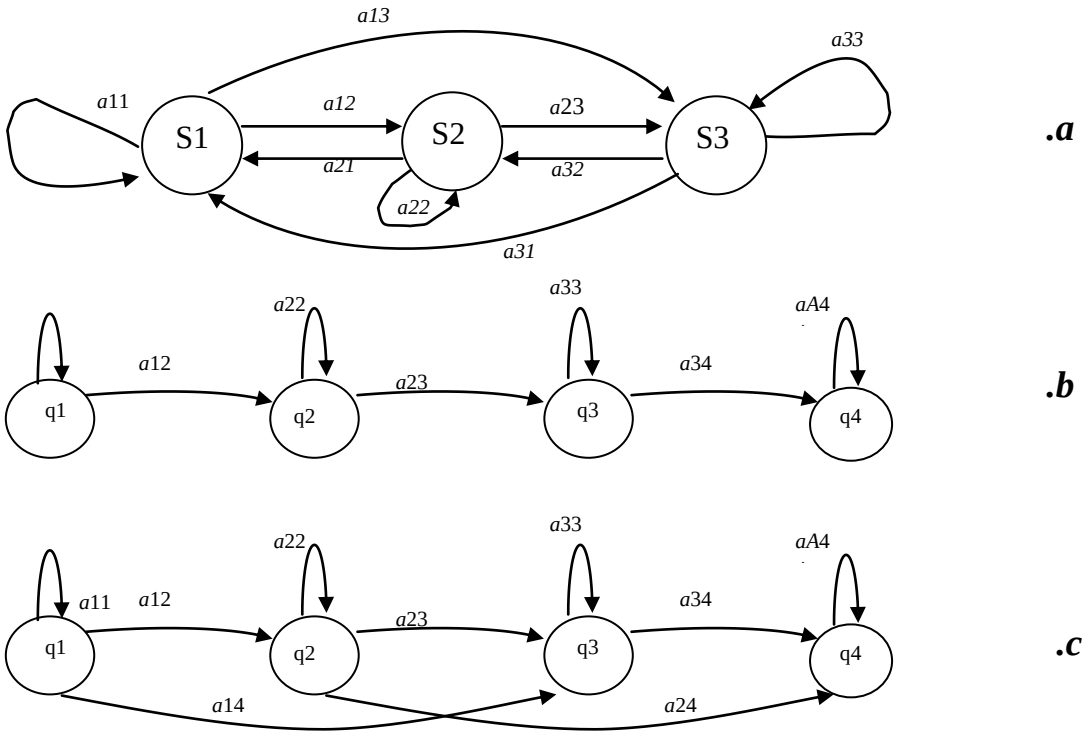


Figure III. 5 les déférentes structures des MMCs

Dans la reconnaissance de la parole il est désirable d'utilisés un modèle avec la propriété left–right

$$a_{ij} = 0, \quad j < i \qquad \textit{(Figure III.5.b et .c)}$$

C'est-à-dire que n'aura pas des Sauts entre les états, ou il n'y aura pas de retour arrière, on note que les coefficients des transitions pour les états finaux sont caractérisés par

$$a_{NN}=1 \quad \text{et} \quad a_{Nj}=0 \quad 1 \leq j < N$$ (où q_N c'est un état final)

3.3.4. Les trois problèmes fondamentaux des MMCs

L'utilisation de MMC nous faire face à trois types de problèmes

3.3.4.1. Evaluation

Sachant ou ayant des vecteurs d'observation $O = (o_1, o_2,...,o_N)$ et $\lambda = \{A, B, \pi\}$

Comment évaluer $P(O/\lambda)$?

Comment trouver le modèle qui a pu générer la séquence observée ?

3.3.4.2. Optimisation

Ayant les vecteurs d'observations $O = \{o_1, o_2, ..., o_N\}$ ainsi que les paramètres du $\lambda = (A, B, \pi)$ du modèle, comment trouver la séquence (cachée) optimale d'états qui explique aux mieux ces observation?

3.3.4.3. Apprentissage

Sachant un corpus d'entraînement O, comment ajuster les paramètres λ du modèle pour maximiser $P(O/\lambda)$?

Prenons un exemple de m modèles, $(\lambda_i, \forall i \in [1, m])$ qui modélise chacun une entité donnée (un mot ou un phonème, par exemple...), soit O une observation dont on veut connaître l'identité

$$i = \arg Max_i \left(p(O/\lambda_i)\right) \qquad \textit{III.15}$$

Soit $Q = q_1, q_2, ..., q_T$ une séquence d'états du modèle de Markov cacher pouvant expliquer O :

$$p(O/\lambda) = \sum_{tous\ les\ Q} p(O,Q/\lambda) = \sum_{tous les\ Q} p(O/Q,\lambda).p(Q/\lambda) \qquad III.16$$

$$p(O/Q,\lambda) = \prod_{t=1}^{T} p(O_t, Q_t/\lambda) = b_{q1}(o_1) \times b_{q2}(o_2) ..... \times b_{qT}(o_T) \qquad III.17$$

Et $p(Q/\lambda) = \pi_{q1} \times a_{q1q2} \times a_{q2q3} \times ..... \times a_{qT-1qT}.$ *III.18*

$$p(O/\lambda) = \sum_{tous\ les\ Q} \pi_{q1} b_{q1}(o_1) \times a_{q1q2} b_{q2}(o_2) \times a_{qT-1qT} b_{qT}(o_T) \qquad III.19\ [11]$$

3.3.5. Exemple 2[12]

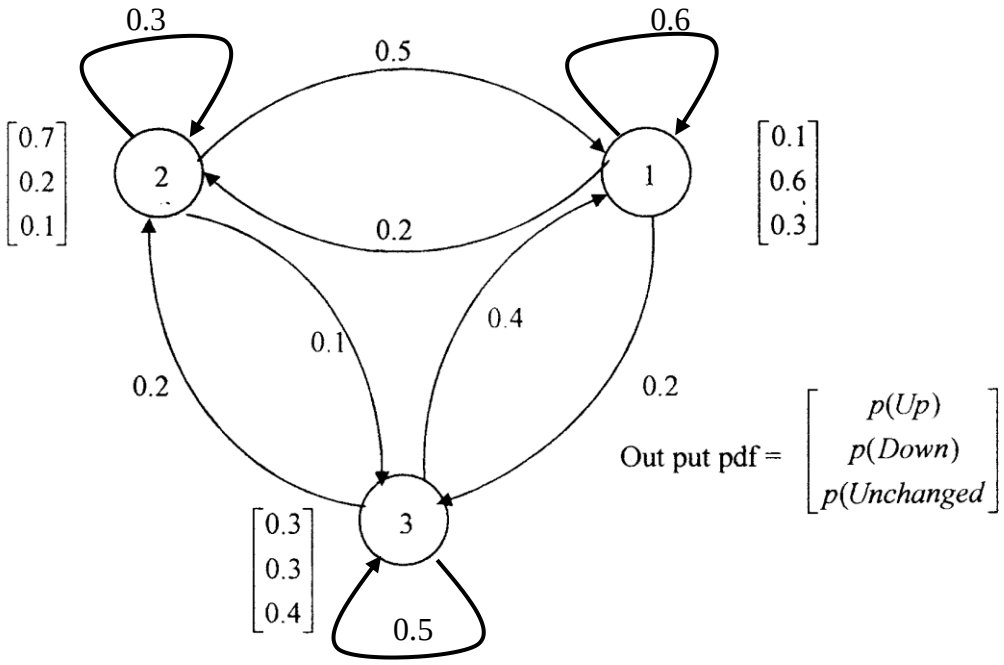
On applique ce problème sur l'exemple suivant :

- Considérant trois marchés {1, 2, 3}, représentés respectivement par {q_1, q_2, q_3} comme des états de chaîne de Markov.
- Les évènements observés sont {UP, UNCHANGED, DOWN}.
- Chaque état est défini par une distribution de probabilité (PDF).
- On peut observer la séquence {UP, UNCHANGED, DOWN}, mais la séquence d'état correspondant est cachée.[13]

La matrice de transition : $A = \begin{bmatrix} 0.6 & 0.2 & 0.2 \\ 0.5 & 0.3 & 0.2 \\ 0.4 & 0.1 & 0.5 \end{bmatrix}$

Le vecteur d'initialisation : $\pi = (0.5, 0.2, 0.3)$

[11] Complexité de calcul : $(2*T-1).N^T$ multiplications, N^{T-1}
[12] http://digital.cs.usu.edu/~cyan/CS7960/hmm-tutorial.pdf
[13] On ne peut pas connaître les marchés dont leurs états sont observés.



Quelle est la probabilité d'avoir 3 jours successives {Down, Down, Down} ?

$$A=\begin{bmatrix}0.6 & 0.2 & 0.2\\ 0.5 & 0.3 & 0.2\\ 0.4 & 0.1 & 0.5\end{bmatrix},\quad \pi=(0.5,0.2,0.3)\quad ,B=\begin{bmatrix}0.7 & 0.2 & 0.1\\ 0.1 & 0.6 & 0.3\\ 0.3 & 0.3 & 0.4\end{bmatrix}.$$

Il existe 27 chemins qui génèrent O :

$C_1=\{q_1,q_2,q_3\}$ $C_2=\{q_1,q_1,q_1\}$ $C_3=\{q_2,q_2,q_2\}$ $C_4=\{q_3,q_3,q_3\}$

$C_5=\{q_2,q_1,q_3\}$ $C_6=\{q_2,q_1,q_1\}$ $C_7=\{q_2,q_1,q_2\}$ $C_8=\{q_2,q_2,q_3\}$

$C_{13}=\{q_1,q_2,q_1\}$ $C_{10}=\{q_2,q_3,q_1\}$ $C_{11}=\{q_2,q_3,q_2\}$ $C_{12}=\{q_2,q_3,q_3\}$

$C_{17}=\{q_1,q_3,q_3\}$ $C_{14}=\{q_1,q_3,q_1\}$ $C_{15}=\{q_1,q_2,q_2\}$ $C_{16}=\{q_1,q_3,q_2\}$

$C_{21}=\{q_3,q_2,q_2\}$ $C_{22}=\{q_1,q_1,q_2\}$ $C_{19}=\{q_3,q_2,q_3\}$ $C_{20}=\{q_3,q_2,q_1\}$

$C_{25}=\{q_3,q_3,q_1\}$ $C_{26}=\{q_1,q_1,q_3\}$ $C_{27}=\{q_3,q_3,q_3\}$ $C_{24}=\{q_2,q_3,q_2\}$

$C_9=\{q_2,q_2,q_1\}$ $C_{18}=\{q_3,q_1,q_1\}$ $C_{23}=\{q_3,q_2,q_3\}$

On peut calculer la probabilité des observations sachant le modèle par l'équation :

$$P(O|\lambda) = \sum_{i=1}^{27} P(O|C,\lambda)$$

Où par le calcule des deux probabilités

$la\ probabilité\ des\ observations\ Q\ sachant\ le\ modèle : P(O|Q,\lambda) = \prod_{t=1}^{T} P(o_t|q_t)$
$Et\ probabilité\ de\ séquence\ d'états : P(Q|\lambda) = \pi_1\ a_{12}\ a_{23}\ ....\ a_{T-1\,T}$ *III.21*

$$P(O|\lambda) = \sum P(O|Q,\lambda)\ P(Q|\lambda)$$ *III.22*

3.3.6. Comment résoudre les trois problèmes

3.3.6.1. Solution au problème 1 Algorithme forward-backward

Soit, $\alpha_t(s) = P(o_1 \cdot \cdot \cdot o_t, s_t = s \mid H)$ la probabilité d'avoir généré la séquence $O = o_1 \cdot \cdot \cdot o_t$ et d'être arrive sur l'état s à l'instant t. Le calcul de cette variable se fait d'une manière inductive comme suit :

- Initialisation : $\alpha_1(s) = \pi(s).P(o_1| s)$ *III.23*
- Induction : $\alpha_t(s) = \left(\sum_{s'\epsilon S} \alpha_{t-1}(s').P(s' \to s)\right)P(O|s)$ *III.24*

Connaissant $\alpha_T(s)$ la probabilité d'avoir généré la séquence O et d'être arrivé sur s pour tout s∈ S, le calcul de P (O | H) est immédiat :

$$P(O|H) = \sum_{s\in S} \alpha_T(s)$$ *III.25*

Cet algorithme est appelé forward car l'induction est réalisée en avant : on calcule tout d'abord la probabilité de générer le premier symbole de la séquence, puis à chaque étape de l'induction on rajoute un symbole et on réitère la procédure jusqu'à ce que l'on ait calculé la probabilité de génération de la séquence entière.

L'utilisation de l'équation *III.21*, même pour des valeurs de T et N peu élevées, n'est pas clairement acceptable :

avec N = 5 et T = 100 le calcul de P(O | H) demande approximativement 10^{72} opérations. Par contre avec l'algorithme forward le calcul nécessite approximativement 3000 opérations.

Un algorithme similaire, c'est l'algorithme backward, peut être utilise pour réaliser ce calcul à l' envers. On utilise alors la variable backward $\beta_t(s)$ = P ($o_{t+1}\cdot\cdot\cdot o_T|s_t$= s, H) qui exprime la probabilité de générer la séquence O = $o_{t+1}\cdot\cdot\cdot o_T$ en partant de l'état s. L'induction suit alors le schéma :

- Initialisation $\beta_T(s)=1$ *III.26*
- Induction $\beta_t(s) = (\sum_{s'\epsilon S}\beta_{t+1}(s').P(s'\rightarrow s)P(o_{t+1}|s))$ *III.27*

Connaissant la probabilité de générer la séquence O en partant de l'état s, le calcul de P(O|H) peut alors être réalisée suivant la formule

$$P(O|H) = \sum_{s\in S}\pi(s)\,\beta_1(s) \qquad III.28$$

3.3.6.2.Solution au problème 2 : Meilleur chemin par l'algorithme de Viterbi

Soit le modèle connu z, Quelle est le meilleur chemin x qui peut donner la meilleure vraisemblance, donc quelle sont les états qui ont réellement participés au calcul de la probabilité $P(O/\lambda)$?

Soit la transition ξ_k est définie comme étant la paire (x_{k+1}, x_k) telle que P (x_{k+1}| x_k) <> 0. On définit ξ comme étant la suite de transitions (ξ_0, ξ_1, . . ., ξ_{K-1}). De toute évidence, une séquence d'´etats de l'instant 0 à K peut être représentée de manière équivalente par x et ξ.

Etant donnée une observation z, on cherche l'état x le plus probable ayant engendré le vecteur observé.

$$P(x, z) = P(z|x)\,P(x) \qquad où\ \ P(z|x) = \prod_{k=0}^{K}P(\xi_k|z_k)$$

Donc $$P(x,z) = \prod_{k=0}^{K}P(x_{k+1}|x_k)\prod_{k=0}^{K}P(\xi_k|z_k) \qquad III.29$$

La séquence d'´etats x telle que la probabilité P (x|z) ou P (ξ |z) soit maximisée. La solution fournie par l'algorithme de Viterbi présente l'avantage d'être récursive et est équivalente au problème de recherche d'un chemin le plus court dans le graphe d'états des x_k.

pour la transition ξ_k, qui permet de passer de x_k à x_{k+1}, on répercute un coût ou une longueur $\lambda(\xi_k)$.

$$\lambda(\xi_k) \equiv -ln\ P(x_{k+1}|x_k) - ln\ P(z_k|\xi_k) \qquad III.30$$

Alors le coût total associé à x sera :

$$\sum_{k=0}^{K} \lambda(\xi k) = -ln\ P\ (x, z) \qquad III.31$$

On en conclut que la recherche du chemin le plus court dans le graphe permet de trouver la solution x au sens du critère du maximum a posteriori [Cornu 2007].

3.3.6.3. Solution au problème 3 Ajuster le modèle : L'algorithme de Baum Welch

L'idée générale derrière cet algorithme est d'estimer les paramètres de notre modèle HMM, ayant en main deux informations essentielles, les vecteurs d'observation en nombre suffisant et le nombre d'états gouvernant les transitions à trouver.

L'algorithme EM (Expectation Maximisation) maximiser l'espérance, calcule l'espérance d'une variable aléatoire manquante par rapport à une variable aléatoire présente, maximise l'espérance trouvée en fonction de ce qui est présent comme variable, par une méthode récursive, jusqu'à ajuster les paramètres des chaînes de Markov notamment les probabilités de transition ainsi que les paramètres des mélanges de gaussiennes.

L'algorithme EM n'est pas démontré mathématiquement, toutefois pour tout détail complémentaire, se référer à [Bilmes 1998].

Soit $\{o\}$ l'ensemble des variables aléatoires connues et $\{Y\}$ les variables aléatoires inconnues, nous supposons qu'il existe une densité de probabilité jointe *z= (0,X)* telle que:

$$p(z / \lambda) = p(Y / O, \theta) \times p(X / \lambda) \qquad III.32$$

L'algorithme

-Définissons une nouvelle quantité *Q*, représentant l'espérance joint z, Etape E de l'algorithme :

$$Q(\lambda / \lambda^{t-1}) = E[\log p(O, Y / \lambda) / Y, \lambda^{t-1}] \qquad III.33$$

Où λ^{t-1} représente le modèle utilisé à l'itération *t*-1 pour calculer λ à l'itération *t*.

-Cette valeur et alors maximisée selon λ , Etape M de l'algorithme.

Donc l'algorithme calcule le modèle :

$$\lambda^t = \arg\max_K (\lambda / \lambda^{T-1}). \qquad III.34$$

Donc à chaque itération, on cherchera si le nouveau modèle apporte une amélioration l'ajustement des données, c'est-à-dire est-ce que le modèle représente les données à l'étape *t* mieux qu'ait l'étape *t*-1.

Dans notre cas, nous allons définir deux nouvelles valeurs qui serviront lors de l'ajustement du modèle, telle que

$$\gamma_i(n) = p(q_i^n / O, \lambda) \qquad III.35$$

Qui représente la probabilité d'être à l'état *q* à l'instant *n*, générant la séquence *O*.

$$\gamma_i(n) = p(q_i^n / O, \lambda) = \frac{p(q_i^n / O, \lambda)}{p(O, \lambda)} = \frac{p(q_i^n / O, \lambda)}{\sum_{j=1}^{L} p(O, q_j^n / \lambda)} \qquad III.36$$

Remarquons que :

$$\alpha(n)\beta(n) = p(o_1, o_2, ...o_i, q_i^n / \lambda) \times p(o_{t+1}, o_{t+2}, ...o_T / q_i^n, \lambda) = p(O, q_i^n / \lambda) \qquad III.37$$

Alors *III.36* devient :

$$\gamma_i(n) = p(q_i^n / O, \lambda) = \frac{\alpha_i(n) \times \beta_i(n)}{\sum_{j=1}^{L} \alpha_j(n) \times \beta_j(n)} \qquad III.38$$

On définit une seconde valeur telle que :

$$\xi_{ij}(n) = p(q_i^n, q_j^{n+1} / O, \lambda) \qquad III.39$$

Qui représente la probabilité d'être à l'état *i* à l'instant *n* et de passer à l'état *j* à l'instant *n+1*, ceci peut être reformulé comme suit :

$$\xi_{ij}(n) = p(q_i^n, q_j^{n+1} / O, \lambda) = \frac{p(q_i^n, q_j^{n+1}, O / \lambda)}{p(O / \lambda)} = \frac{\alpha_i(n) \times a_{ij} \times b_j(o_{n+1)}) \times \beta_i(n)}{\sum_{J+1}^{L} p(O, q_j^n / \lambda)}$$ L'on peut

remarquer que :

$$\sum_{n=1}^{T} \gamma_i(n) \qquad \textit{III.40}$$

Représente la valeur espérée d'être à l'état q_i. pendant tout les instants n pour toutes les observations *O* donnant ainsi le nombre de transitions partant de l'état q_i

Et aussi que

$$\sum_{n=1}^{T} \xi_{ij}(n) \qquad \textit{III.41}$$

Représente le nombre de transitions de l'état q_i à l'état q_j pour toutes les observations *O* L'utilisation de l'algorithme EM pour estimer les nouveaux paramètres à chaque itération nécessite de mettre à jour les valeurs manquantes itérativement de la manière suivante:

La quantité

$$\pi = \gamma_i(1) \qquad \textit{III.42}$$

Qui est la fréquence relative de passage à l'état q_i à l'instant 1.

Ainsi que :

$$\widetilde{a}_{ij} = \frac{\sum_{n=1}^{N-1} \xi_{ij}(n)}{\sum_{n=1}^{N} \gamma_i(n)}$$

Qui représente le nombre de transitions de l'état q_i à l'état q_j relatif au nombre de transitions sortant de l'état q_i.

Pour le mélange des gaussiennes, les paramètres à estimer sont les moyennes et variances des gaussiennes ainsi que le taux de participation de la gaussienne à l'état q_i noté:

$$\widetilde{c}_{il} = \frac{\sum_{n=1}^{N} \gamma_{il}(n)}{\sum_{n=1}^{N} \gamma_i(n)} \qquad \textit{III.43}$$

Où *l* représente la $l^{iéme}$ gaussienne modélisant les vecteurs d'observation à un état q_i.

$$\mu_{il} = \frac{\sum_{n=1}^{N} \gamma_{il}(n) \times o_t}{\sum_{n=1}^{N} \gamma_{il}(n)} \qquad III.44$$

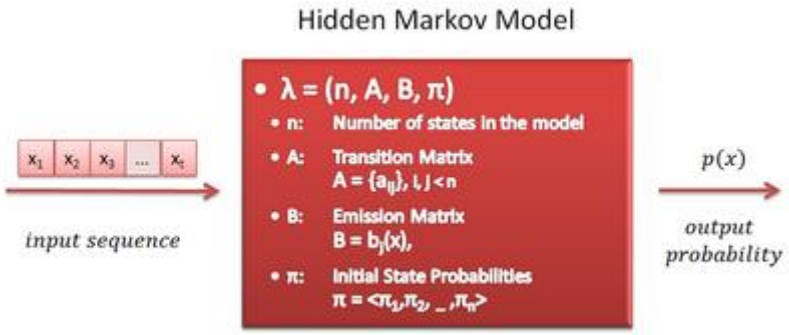
Représentant la moyenne de chaque gaussienne à l'état q_i.

$$\mu_{il} = \frac{\sum_{n=1}^{N} \gamma_{il}(n) \times (o_t - \mu_{il}) \times (o_t - \mu_{il})^T}{\sum_{n=1}^{N} \gamma_{il}(n)} \qquad III.45$$

Représentant la variance de chaque gaussienne à l'état q_i.

3.4. Conclusion

Si on veut juste utiliser la chaine de Markov caché sans apprendre son fonctionnement interne, on doit Penser à une boite noire qui a en entrée une séquence d'observations, et qui mesure en sortie la similarité aux autres séquences.



Dans ce cas il nous faut des utilitaires prédéveloppées (exemple de HTK ou CMU Sphinx) ; qui nous facilitent la grande partie de travail.

Comme on a énoncé dans les paragraphes précédents HMMs, ce sont des modèles probabilistes qui tentent de trouver la proximité entre deux choses ou leurs comportements d'une manière concise et plus facile à gérer ; et dits aussi cachés en raison qu' on est pas obligé de démontrer les états dont le processus de reconnaissance est basé sur eux.

&Dans cet article, nous avons exploré ce qu'est un Modèles de Markov cachés et ce qu'il peut faire. Après une brève théorie, le chapitre suivant va montrer comment créer, apprendre et utiliser HMM utilisant les bibliothèques crées par Accord.NET Framework.

Chapitre VI : La voix sur IP et les serveurs vocaux Interactifs (les protocoles h323 et SIP)

4.1. Introduction

Le développement rapide et l'utilisation croissante de l'internet et des réseaux informatiques pour les services de communications, y compris les applications de téléphonie, sont devenus des domaines importants pour l'industrie des télécommunications.

L'apparition récente de la transmission de la voix et de la vidéo sur IP (internet protocole) représente une avancée technologique importante dans le domaine du multimédia et offre un service conçu pour permettre aux compagnies d'utiliser leurs réseaux pour y faire passer leur trafic de la voix sans nécessiter de changement des équipements ou réseaux existants.

4.1.1. La combinaison SVI & DHM

La communication vocale homme machine offre la capacité de communication, de commande et de contrôle par la voix. Elle s'exploite au mieux dans une interaction vocale avec la machine, ayant pour but de résoudre des problèmes, de pouvoir déclencher des actions ou d'acquérir de l'information.

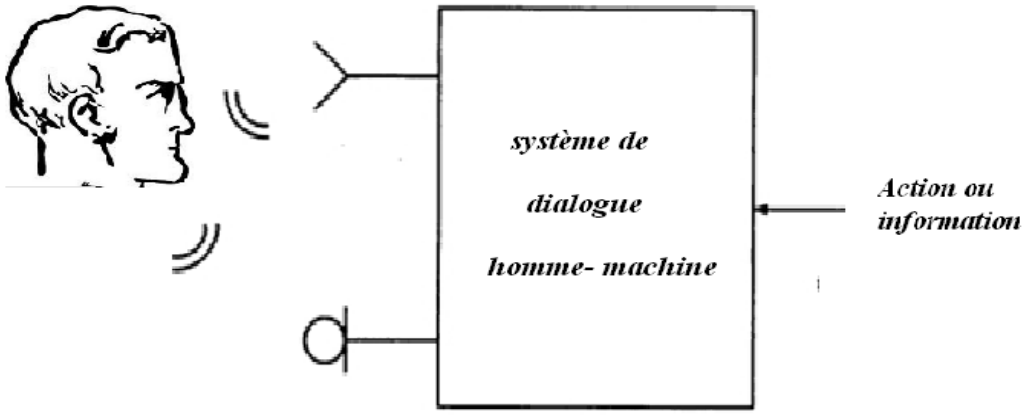


Figure IV. 1 ***système de dialogue homme machine***

Les systèmes de dialogue s'organisent comme un arrangement cohérent de sous-systèmes, qui prennent chacun en charge des traitements spécialisés comme la synthèse ou la reconnaissance vocale. Différents niveaux de composants forment donc un système de DHM comme le montre la *Figure IV.1* des parties sous-jacentes du système qui résolvent les problèmes de traitement de la parole et, d'autres parties qui servent à interpréter les dialogues souvent via un contrôleur de dialogues. Dès lors, plusieurs choix s'offrent à un concepteur de dialogues applicatifs : soit la réalisation de tout ou partie des briques élémentaires d'un

système, soit l'utilisation d'une plateforme existante avec son contrôleur possédant des références statiques ou dynamiques vers des briques élémentaires de traitement de la parole.

Une autre contrainte de conception du système de dialogue est l'obligation de fonctionner ou non dans le cadre d'une interaction téléphonique. Les solutions de communication vocale homme-machine suite à une **interaction téléphonique** sont souvent appelées **serveur vocal interactif**. Cette dénomination tient compte des capacités des systèmes à fonctionner dans le cadre d'une bande passante téléphonique c'est à dire de 300Hz à 300kHz avec comme fonctionnalités potentielles la synthèse vocale, la possibilité d'enregistrer et de jouer des enregistrements, la reconnaissance vocale, la reconnaissance des codes DTMF[14] et la commande vocale.

Figure IV. 2 Architecture fonctionnelle d'un système de dialogue homme-machine couplé à une base de données (source Vecsys)

Le concept de SVI a bien évolué, depuis sa création aux laboratoires BELL en 1941, dans sa déclinaison uniquement à touches et son application au téléphone en 1962 au Seattle World Fair. Néanmoins, le SVI est souvent resté limité à un domaine spécifique du langage et aux dialogues finalisés. Cependant, des initiatives de plus en plus courantes font émerger auprès du grand public des déclinaisons dites en **langage naturel**. La première question posée est alors une question ouverte qui appelle l'utilisateur à répondre librement. Cette question sert à découvrir dynamiquement le motif d'appel en utilisant des modèles statistiques basés sur l'étude de précédentes conversations. D'autres initiatives s'intéressent à modifier le côté statique du dialogue des SVI. La dynamique dans les dialogues permet alors, de modifier l'initiative du dialogue, de prendre en compte le contexte, c'est-à-dire, l'historique des

[14] DTMF pour Dual-Tone Multi-Frequency

réponses et les préférences de l'utilisateur puis, de construire les dialogues à la volée. La société Yseop[15] introduit pour ce genre de système la terminologie de **serveur vocal interactif intelligent** ou de **système expert virtuel**. Le système se base alors sur l'enregistrement formel du savoir faire de l'entreprise et de ses bonnes pratiques sous forme de règles et de faits puis, utilise un moteur d'inférence pour déduire les propositions à apporter à l'utilisateur en réponse à ses sollicitations et au contexte.

Le SVI peut aussi être couplé à la vidéo pour devenir alors un serveur vocal et vidéo interactif. Ce type d'interaction va dans le sens du multimédia, et nous permet de voir un interlocuteur en plus d'interagir avec un scénario applicatif vocal. Le système fait ainsi partie des services à valeur ajoutée des opérateurs de télécommunications, grâce à la perspective offerte par les réseaux haut débit comme le réseau 3G. Le SVVI est donc une initiative qui rend possible, à son niveau, l'interaction visuelle avec un agent physique ou virtuel. Le SVVI permet aussi à un agent physique de prendre le relais lorsque l'automate n'est plus qualifié pour agir ou bien d'être vu comme une perspective multimodale.

4.2. Technologies appliquées aux SVI

4.2.1. Les réseaux, principes fondamentaux

Pour mettre en œuvre un réseau informatique il est nécessaires définir un cahier des charges mettant en avant les caractéristiques d'applications souhaitées pour ce réseau, puis de choisir parmi les possibilités, dont les plus courantes sont définies dans cette proposition de classification. Les réseaux informatiques sont classés en grandes catégories en fonction de la distance :

- **Les PAN** (Personal Area Network)sont les interconnexions sur quelques mètres d'un ordinateur, d'un terminal GSM (téléphone mobile, organiseur...)... Ils correspondent à la taille d'un bureau, aux appareils d'un seul utilisateur.
- **Les LAN** (Local Area Network) ou réseaux locauxsont les plus répandus. Ils correspondent à des réseaux s'étendant sur plusieurs centaines de mètres, par exemple dans une entreprise, une salle de spectacle... Leur débit est élevé, de 10 Mb/s à 100 Mb/s, voire plus récemment 1Gb/s...

[15] http://www.yseop.com/FR/home.html

- **Les MAN** (Metropolitan Area Network) ou réseaux métropolitains permettent d'interconnecter des bâtiments, des entreprises, etc., dans les dimensions d'une ville sur des réseaux haut débits.
- **Les WAN** (Wide Area Network) ou réseaux étendussont destinés à transporter des données sur des distances allant d'un pays à plusieurs continents. Le WAN le plus utilisé est internet.

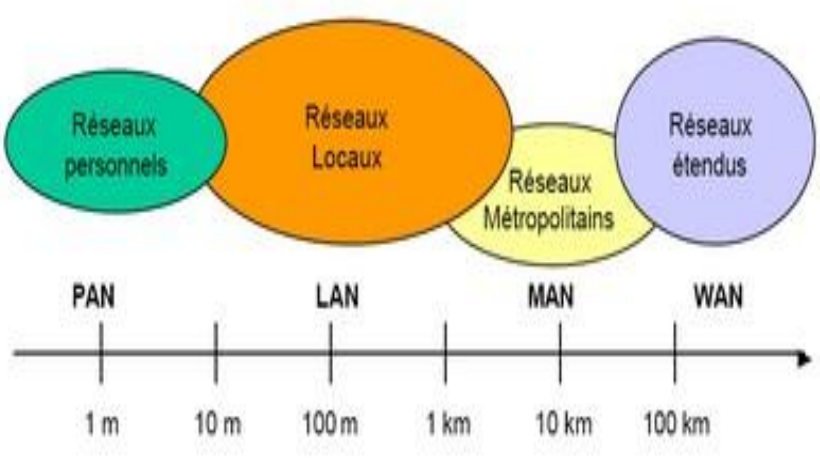


Figure IV. 3 Les grandes catégories de réseaux informatiques

4.3. Les communications en VoIP

Le terme "VoIP" est en général utilisé pour décrire des communications "Point à Point". Pour la diffusion de son sur IP en multipoints, on parlera plutôt de streaming (comme les radios sur Internet, par exemple).

Le transport de communication sur IP est très dépendant du "temps de latence" d'un réseau qui influe beaucoup sur la qualité "psycho-acoustique" d'une conversation. Avec l'avènement des réseaux 100Mb/s et ADSL, les temps de latences deviennent acceptables pour une utilisation quotidienne de la voix sur IP.

4.3.1. L'architecture TCP/IP (Transmission Control Protocol / Internet Protocol)

Elle a été mise en place dans les années 1970 par le Département Américain de la Défense, qui, voyant se multiplier les protocoles de communication et leurs incompatibilités, a décidé de créer sa propre architecture. Aujourd'hui, cette architecture est très répandue : elle est notamment à la source d'internet et de nombreux intranets.

Le protocole[16] TCP/IP assure le transport de paquets d'une extrémité à l'autre du réseau avec une certaine sécurité, en s'appuyant essentiellement sur deux protocoles : l'IP, correspondant au niveau 3 du modèle de référence, et le TCP (ou UDP) de niveau 4 du modèle de référence OSI.

✓ **IP** : Les paquets IP sont indépendants les uns des autres et routés individuellement dans le réseau par des routeurs. Le protocole IP propose une qualité de service très faible : les paquets perdus ne sont pas détectés et rendent impossible la reprise de la communication sur une erreur.

✓ **TCP et UDP** : Au niveau message, correspondant à la couche 4 du modèle de référence OSI, deux protocoles sont utilisés en fonction du mode de connexion. Le mode avec connexion correspond à un cas où avant l'envoi d'un message, l'émetteur et le récepteur doivent se mettre en accord, par opposition au mode sans connexion qui permet l'envoi de messages sans l'accord du destinataire. Dans le mode avec connexion, les paquets sont transportés sous forme de datagrammes, dans le mode contraire, sous forme de segments. Le protocole TCP (Transmission Control Protocol) en mode avec connexion est assez complexe et comporte en plus des fonctionnalités de niveau message de nombreuses options permettant de résoudre les problèmes de pertes de paquets dans les niveaux inférieurs. L'UDP (User Datagram Protocol) fonctionne en mode sans connexion et n'assure pas la retransmission des segments si ceux-ci n'arrivent pas à destination. L'émetteur n'a aucun moyen de savoir si un message a été reçu correctement ou non. UDP est donc moins fiable que le TCP, mais plus rapide et facile à mettre en œuvre. Il n'a pas non plus de fonction de réparation d'erreurs [G.Pujolle 1998].

4.3.2. Les protocoles du VoIP

Les principaux protocoles utilisés pour l'établissement de connexions en voix sur IP sont :

- H323
- SIP

[16] *Un protocole est une méthode standard qui permet la communication entre des processus (s'exécutant éventuellement sur différentes machines), c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Il en existe plusieurs selon ce que l'on attend de la communication. Certains protocoles seront par exemple spécialisés dans l'échange de fichiers, d'autres pourront servir à gérer simplement l'état de la transmission, et des erreurs.*

- MGCP

Le protocole H323 est le plus connu et se base sur les travaux de la série H.320 sur la visioconférence sur RNIS. C'est une norme stabilisée avec de très nombreux produits sur le marché (terminaux, gatekeeper, gateway, logiciels). Il existe actuellement 5 versions du protocole (V1 à V5).

Le protocole SIP est natif du monde Internet (HTTP) et est un concurrent direct de l'H323. A l'heure actuelle, il est moins riche que H.323 au niveau des services offerts, mais il suscite actuellement un très grand intérêt dans la communauté Internet et télécom.

Le protocole MGCP est complémentaire à H.323 ou SIP, et traite des problèmes d'interconnexion avec le monde téléphonique (SS7, RI).

Les principaux protocoles utilisés pour le transport de la voix en elle-même sont :

- RTP
- RTCP

Situé entre la couche UDP et la couche application, RTP (*Real-time Transfer Protocol*) a pour but principal d'offrir une connexion de bout-en-bout et en temps réel sur Internet. Le protocole de contrôle RTCP sert pour surveiller la qualité des services offerts et pour fournir des informations concernant les partenaires de la conversation. Le contrôle se résume à des aspects "simples", c'est-à-dire qu'il ne supporte pas tous les besoins de contrôle demandés par l'application.

4.3.3.1. Le protocole H 323

Il est devenu nécessaire de créer des protocoles capables de supporter l'arrivée des technologies du multimédia sur les réseaux, telles que la visioconférence, qui est une opération d’envoi des données en temps réel.

Le protocole H 323 fait paraitre pour permettre entre autres de faire de la visioconférence sur des réseaux IP.

H.323 est un protocole de signalisation défini par l'ITU-T[17] en 1996 permettant l'établissement, la libération et la modification de sessions multimédia (voix, vidéo, données). Il hérite du protocole Q.931 du RNIS qu'il enrichit pour son fonctionnement dans des réseaux de transport en mode paquet.

Le protocole H.323 supporte un ensemble de services complémentaires similaires à ceux mise en œuvre dans un réseau RNIS.

D'abord H.323 est définie pour la transmission de la voix sur réseau local (LAN) mais de plus en plus avec le développement des techniques, la norme H.323 est amélioré et appliqué sur les réseaux d'ordinateur plus grand (Internet, Intranet)[18].

Historique des protocoles existants

- **1ère génération** (jusqu'en 1992) :
- H.320 : Adapté pour le RNIS (inférieur à 2Mbs)
- H.321 : Adaptation de H.320 pour l'ATM
- H.322 : Pour les LAN avec de la QoS
- **2ème génération** (1992 à 2007) :
- H.310 : ATM
- H.323 : Pour les LAN sans QoS (sur IP,Eth.)
- H.324 : RTC
- SIP (Session Initiation Protocol)
- **3ème génération** (depuis 2007) :
- H.325.

H.323 est un regroupement de plusieurs protocoles qui concernent trois catégories distinctes la signalisation, la négociation de codecs et le transport de l'information.

[17] L'ITU (International Télécommunication Union) est la plus ancienne organisation internationale technique de coordination. L'**ITU-T** traite les questions techniques et de normalisation. À chaque catégorie de normes correspond à une lettre de l'alphabet.

[18] http://www.frameip.com/voip/#6.1_-_Protocole_H323

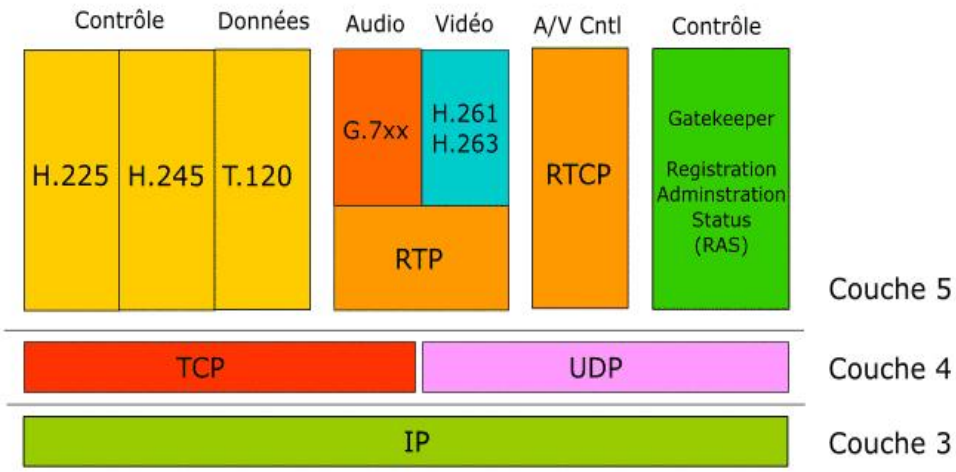


Figure IV. 4 L'établissement d'un appel point à point H.323

Le respect du standard H.323 permet de garantir un contrôle sur l'utilisation des ressources réseaux et des contraintes de qualité de service. Tous les terminaux H.323 doivent supporter :

- Le protocole H.245 qui négocie l'ouverture et l'utilisation des canaux ainsi que les paramètres de la communication voix. La négociation est utile pour mettre d'accord les terminaux et les équipements voix qui communiquent entre eux sur les choix du type des données transportées, les langages utilisés entre les équipements doivent s'adapter aux contraintes imposées par le support de transmission notamment et par les équipements eux-mêmes. Le choix du codec est très important (G7xx et H26x sur le schéma), du moins gourmand en bande passante à celui qui offre la meilleur qualité vocale.
- Le protocole H.225 (SIG) pour la signalisation et l'établissement d'appels.
- Le protocole H.225 (RAS) (Registration/Admission/Status), qui est le protocole utilisé par le terminal pour communiquer avec le serveur de contrôle d'appels.
- Les protocoles RTP/RTCP (Real Time Protocol/Real Time Control Protocol) transportent les flux audio et vidéo.

Le T.120 permet l'ouverture d'un canal pour le partage d'applications.

4.3.3.1.1. Appel base d'un terminal à un terminal

L'établissement d'un appel point à point H.323 on utilise deux connexions TCP entre les terminaux, l'une pour établir d'appel (*Q.931* port *N°port*) et l'autre pour les messages de contrôle des flux média(H.245).

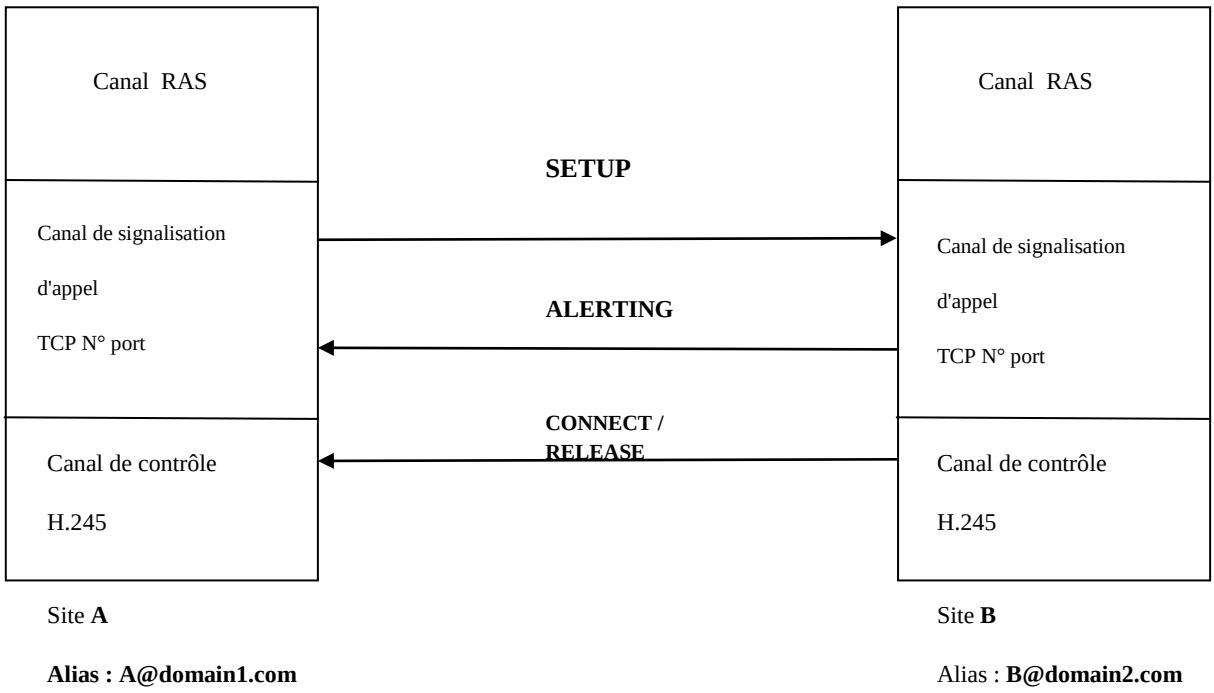


Figure IV. 5 L'établissement d'un appel point à point H.323

Le terminal ***A*** envoie au terminal ***B*** un message Q.931[19] ***SETUP*** sur le port ***N°port*** pour établir l'appel. Dès la réception de message Q.931le terminal ***B*** doit répondre par un message ***ALERTING***.

Ensuit utilisateur ***B*** a jusqu'à 3 minutes pour accepter ou refuser l'appel par le message CONNECT ou le message ***RELEASE COMPLETE*.**

[19] Q.931 est défini par l'UIT-T (ITU-T en anglais) comme, Q.931 est utilisée pour transmettre et recevoir des messages de signalisation d'appel selon le protocole H.225.

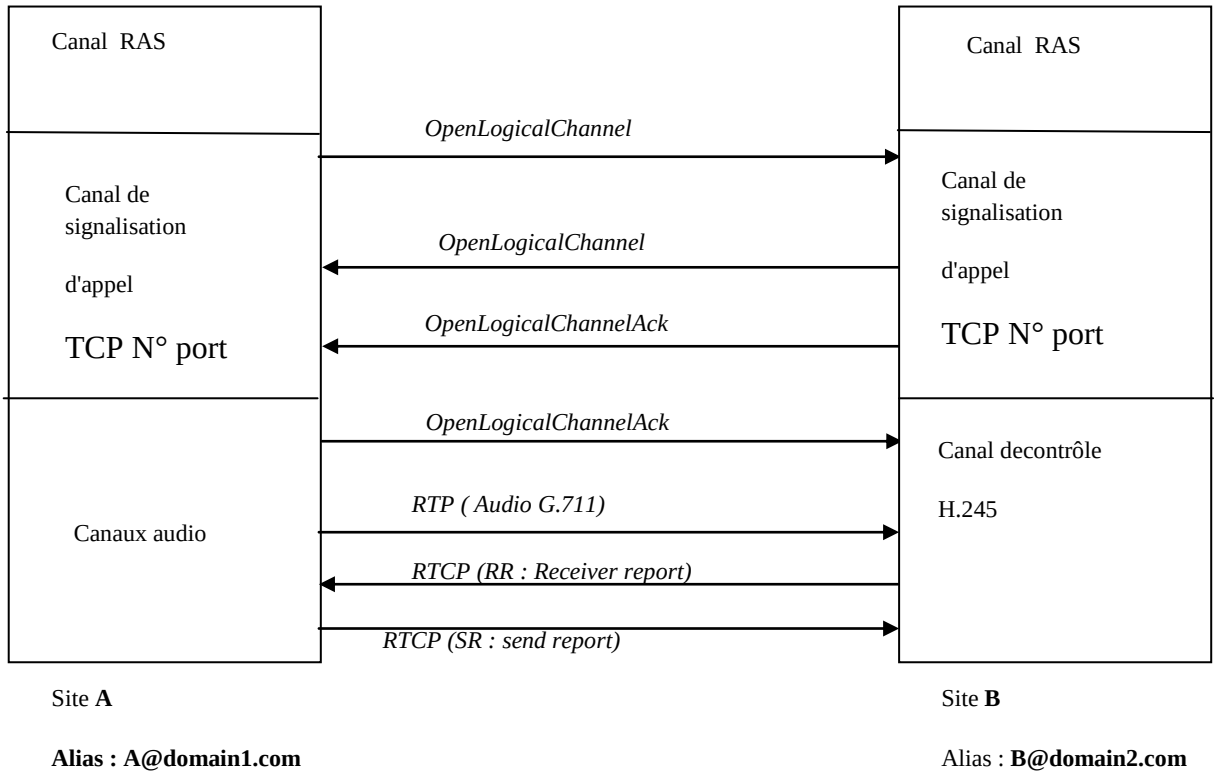


Figure IV. 6 L'établissement d'un appel point à point H.323

Une fois l'appel est accepté, il y aura un message envoyé sur le canal de contrôle *H245* ***TerminalCapabilitySet*** pour négocier les capacités des canaux médias. Après la négociation des capacités, les terminaux doivent ouvrir des canaux médias pour la voix. Pour ouvrir un canal logique vers terminal ***B***, le terminal ***A*** va envoyer un message H.245 ***OpenLogicalChannel.*** Dès recevoir le message ***OpenLogicalChannel*** le terminal ***B*** renvoie le message ***OpenLogicalChannelAck*** pour acquitter l'ouverture de ce canal logique et renvoyer les autres informations.

A ce stade, les terminaux peuvent se parler par les canaux audio. Le flux média est envoyé dans des paquets RTP et les rapports de réception RTCP permettent à chaque terminal de mesurer la qualité de service du réseau.

Pour le relâchement de l'appel un terminal ***A*** doit envoyer un message H.245 ***CloseLogicalChannel*** pour chaque canal logique qu'il a ouvert. D'autre part le terminal ***B*** en accuse réception doit répondre par un message H.245 *CloseLogicalChannelAck*. Et puis le

terminal ***A*** envoie un message H.245 *EndSessionCommand* et attende de recevoir le même message de ***B*** et enfin ferme le canal de contrôle.

Le Gateway

Un Gateway est un endpoint du réseau qui assure en temps réel des communications bidirectionnelles entre des terminaux H.323 et d'autres terminaux (e.g., terminaux RTC, RNIS, GSM).

Le Gatekeeper

Un Gatekeeper est le composant le plus important d'un réseau H.323. Il agit comme tant le point central pour tous les appels dans sa zone et contrôle les endpoints. Un Gatekeeper H.323 agit comme un commutateur virtuel.

Le Gatekeeper exécute deux fonctions importantes. La première est la translation d'adresse d'un alias LAN d'un terminal ou d'une passerelle (Gateway) vers une adresse IP ou IPX, comme le définit la spécification RAS. La deuxième fonction est la gestion de la bandepassante, aussi décrite dans la spécification RAS.

4.3.3.2. Le protocole SIP

Le protocole SIP (session Initiation Protocol) a été initié par le groupe MMUSIC (*Multiparty Multimedia Session Control*) et est désormais repris et maintenu par le groupe SIP de l'IETF.SIP est un protocole de signalisation appartenant à la couche application du modèle OSI.

Son rôle est d'ouvrir, modifier et libérer les sessions. L'ouverture de ces sessions permet de réaliser de l'audio ou la vidéoconférence, de l'enseignement à distance, de la voix (téléphonie) et de diffusion multimédia sur IP essentiellement.

Un utilisateur peut se connecter avec les utilisateurs d'une session déjà ouverte. Pour ouvrir une session, un utilisateur émet une invitation transportant un descripteur de session permettant aux utilisateurs souhaitant communiquer de s'accorder sur la compatibilité de leur media, SIP permet donc de relier des stations mobiles en transmettant ou redirigeant les requêtes vers la position courante de la station appelée. Enfin , SIP possède l'avantage de ne pas être attaché à un médium particulier et censé être indépendant du protocole de transport des couches basses.

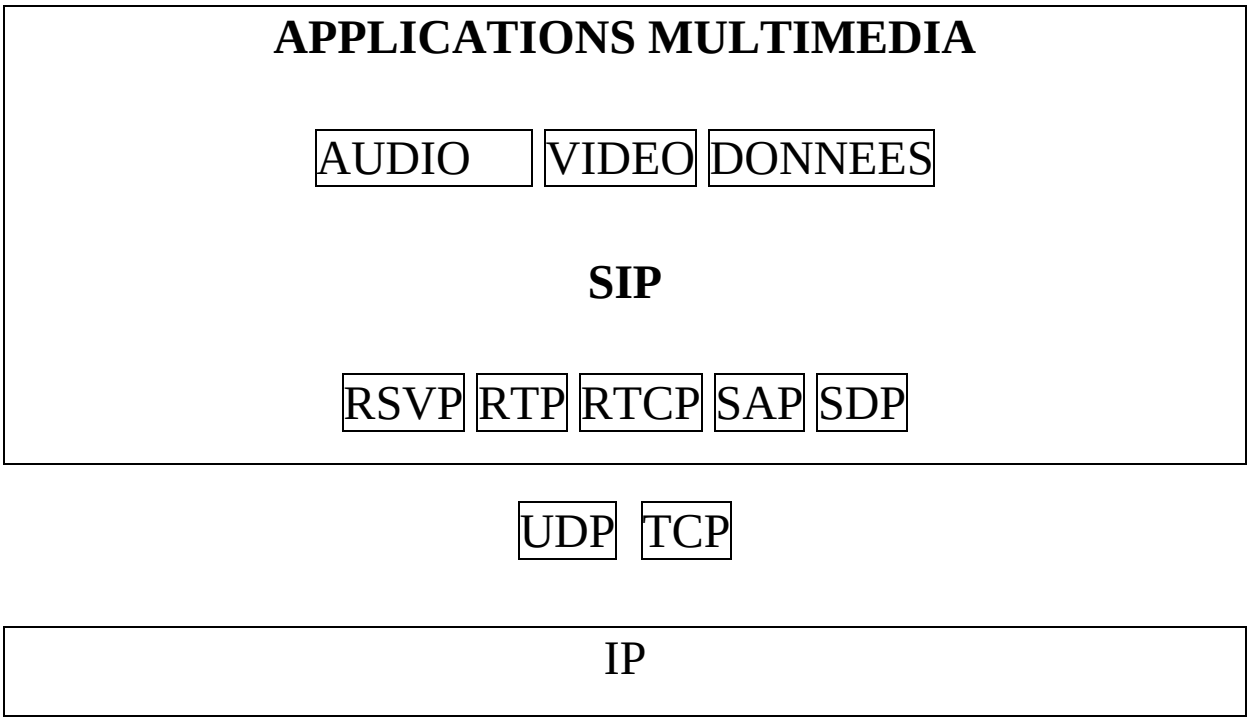


Figure IV. 7 L'architecture en couches de SIP, telle que la présente le modèle OSI

4.3.3.2.1. Fonctionnement

Le protocole SIP repose sur un modèle requête/réponse. Lorsqu'un utilisateur désire rentrer en communication avec un autre via IP. l'application utilisée fait appel au protocole SIP en précisant la nature des échanges. SIP définit ainsi le nombre de session à ouvrir et le protocole le mieux adapté à l'échange. On distingue ainsi trois modes d'ouverture de sessions

- **Point à point**

Permet une communication entre deux machines, on parle d'unicast.

- **Diffusif**

Plusieurs utilisateurs en multicast, via une unité de contrôle M.C.U (Multipoint Control Unit).

- **Combinatoire**

Plusieurs utilisateurs pleinement interconnectés en multicast via le réseau à maillage complet de connexion.

- **Les avantages du protocole SIP**

L'implémentation de la VoIP avec le protocole de signalisation SIP (Session Initiation Protocol) fournit un service efficace, rapide et simple d'utilisation. SIP est un protocole rapide et léger. La séparation entre ses champs d'en-tête et son corps du message facilite le traitement des messages et diminue leur temps de transition dans le réseau. SIP un est protocole indépendant de la couche transport. Il peut aussi bien s'utiliser avec TCP qu'avec UDP. SIP est un protocole plus rapide. Le nombre des en-tête est limité (36 au maximum et en pratique moins d'une dizaine d'en-tête sont utilisées simultanément), ce qui allège l'écriture et la lecture de requêtes et réponse.

4.3.3.2.2. Structure du protocole SIP[20]

Contrairement du H.323, SIP n'utilise pas des messages issus du protocole ISDN, mais un ensemble méthodes de réponse pour certains similaires aux méthodes de réponse du protocole HTTP. Cependant SIP diffère du protocole HTTP par un ensemble de méthodes propres dont les plus basiques sont :

- **INVITE**

Requête envoyée pour commencer un appel

- **ACK**

Requête envoyée par le client qui atteste la bonne réception de la réponse du serveur à sa précédente requête.

- **CANCEL**

Annulation de la requête précédente tant que le serveur n'y a pas répondu.

- **BYE**

Requête de relâchement de l'appel. Et par un ensemble de codes de réponses regroupées par familles.

- **1XX**

[20] http://blog.wikimemoires.com/2011/03/protocole-sip-comparation-entre-sip-h323/

Désigne une information (ex : 100 TRYING, 180 RINGING, 183 SESSION PROGRESS).

- **2XX**

Désigne que la requête a bien été reçue et accepter

- **3XX**

Désigne une redirection (ex : 305 USE PROXY)

- **4XX**

Désigne une erreur côté client (ex : 400 BAD REQUEST, 401 UNAUTHORISED, 404 NOT FOUND)

- **5XX**

Désigne une erreur côte serveur (ex :500 INTERNAL SERVER ERROR, 502 BAD GATEWAY)

- **6XX**

Désigne un problème global (ex : 600 BUSY EVERYWHERE). Il est important de préciser, du fait de sa conception peer to peer, qu'un terminal SIP pourra se comporter à la fois comme un client et un serveur, il pourra donc émettre et répondre aux requêtes qu'il reçoit. L'architecture SIP repose sur 3 entités l'user agent, le serveur d'enregistrement, le serveur Proxy SIP, ainsi qu'une Geteway chargée les appels vers le PSTN (Public Switched Téléphone Network).

4.3.3.2.3. Les différentes entités d'une architecture SIP

- **L'user Agent**

Se situe typiquement sur les terminaux abonnés (softphone, IP-phone, adaptateur, PDA). SIP, 2 users agents peuvent communiquer directement, en point à point à condition que les adresse IP soient connues et accessibles. Un user agent général génère des requêtes SIP (REGISTRAR, INVITE) mais peut également y répondre ;

Figure IV. 8 Le serveur d'enregistrement (REGISTRAR)

Est le responsable du traitement des requêtes REGISTRER envoyées par les user agents, il permet d'associer une URL à l'adresse IP de l'user agent. Elle sert également de location serveur et peut assurer des mécanismes d'authentification (username/passeword). En général, le Registar SIP stockent les informations des user agents dans une base de données.

Le Proxy SIP

Est chargé de transmettre les **INVITE** d'un agent vers un autre dans le cas où ceux-ci ne peuvent être joints directement en point à point. Le proxy SIP va pouvoir interroger la base des données d'enregistrement, récupérer l'URL/adresse IP du destinataire et ainsi transmettre l'invite de l'appelant. Il se comporte à la fois comme un serveur et un client. En plus des fonctions d'aiguillage, le proxy SIP peut également être utilisé pour du contrôle d'appels/abonnés et la facturation (billing) lorsqu'il s'agit d'un stateful proxy.

La Gateway SIP

Permet de véhiculer les appels vers le PSTN et inversement.

Les inconvénients

L'une des conséquences de cette convergence est que le trafic de voix et ses systèmes associés sont devenus aussi vulnérable aux menaces de sécurité que n'importe quelle autre véhiculée parle réseau. En effet, SIP est un protocole d'échange de messages basé sur http. C'est

pourquoi SIP est vulnérable face à des attaques de types Dos(dénis de service), associé RTP (Real Time Protocol) est lui aussi très peu sécurisé face à de l'écoute indiscrète ou des DoS.

Le SIP est une norme pour la communication de multimédia, il devient de plus en plus utilisé pour la mise en place la téléphonie sur IP, la compréhension de ce protocole aidera le professionnel à l'épreuve de la sécurité sur le réseau .Ce protocole est un concurrent direct à H.323.

4.3.4. Etude comparative entre SIP et H.323

Les deux protocoles SIP et H323 représentent les standards définis jusqu'à présent pour la signalisation à propos de la téléphonie sur Internet .Ils présentent tous les deux des approches différentes pour résoudre un même problème.

H323 est basé sur une approche traditionnelle du réseau à commutation de circuits. Quant à SIP, il est plus léger car basé sur une approche similaire au protocole http. Tous les deux utilisent le protocole RTP comme protocole de transfert des données multimédia. Au départ H323 fut conçu pour la téléphonie sur les réseaux sans QoS, mais on l'adopta pour qu'il prenne en considération l'évolution complexe de la téléphonie sur internet. Pour donner une idée de la complexité du protocole H323 par r apport à SIP, H323 est défini en un peu plus de 700 pages et SIP quand à lui en moins de 200 pages. La complexité de H323 provient encore du fait de la nécessité de faire appel à plusieurs protocoles simultanément pour établir un service, par contre SIP n'a pas ce problème.

SIP ne requiert pas de comptabilité descendante, SIP est un protocole horizontal au contraire de H323 : Les nouvelles versions de H323 doivent tenir compte des anciennes versions pour continuer à fonctionner. Ceci entraîne pour H323 de traîner un peu plus de codes pour chaque version. H323 ne reconnaît que les Codecs standardisés pour la transmission des données multimédias proprement dit alors que SIP, au contraire, peu très bien en reconnaître d'autres. Ainsi, on peut dire que SIP est plus évolutif que H323.

En résumé, La simplicité, la rapidité et la légèreté d'utilisation, tout en étant très complet, du protocole SIP sont autant d'arguments qui pourraient permettre à SIP de convaincre les investisseurs. De plus, ses avancées en matière de sécurité des messages sont un atout important par rapport à ses concurrents.

	SIP	H323
Nombre échanges pour établir la connexion	1,5 aller-retour	6 à 7 aller-retour
Maintenance du code protocolaire	Simple par sa nature textuelle à l'exemple de Http	Complexe et nécessitant un compilateur
Evolution du protocole	Protocole ouvert à de nouvelles fonctions	Ajout d'extensions propriétaires sans concertation entre vendeurs
Fonction de conférence	Distribuée	Centralisée par l'unité MC
Fonction de téléservices	Ou i, par défaut	H.323 v2 + H.450
Détection d'un appel en boucle	Oui	Inexistante sur la version 1 un appel routé sur l'appelant provoque une infinité de requêtes
Signalisation multicast	Oui, par défaut	Non

Table IV. 1 Récapitulation comparative entre SIP et H.323[21]

4.3.5. L' avenir du SIP

SIP reste un protocole jeune, mais son succès n'est plus à débattre, bien qu'il soit simple de fonctionnement et flexible, SIP souffre toujours de son manque d'interopérabilité avec les réseaux NATES.

Bien que les routeurs de dernières générations supportent le SIP dans leur table de translation NAT, il est parfois nécessaire de recourir à diverse solutions comme un serveur STUN (Simple Traversal of UDP trough Nat), une gestion des proxys keep alive (côté proxy) ou bien

[21] http://www.packetizer.com/ipmc/h323_vs_sip/

utiliser la fonction TURN (Traversal Using Nat) ou ICE (Connectivity Establishement) pour maintenir les sessions NAT valides au niveau du routeur /firewall et conserver les appels entrants fonctionnels.

Néanmoins SIP offre des possibilités d'utilisations bien au-delà de la VoIP (messagerie instantanée IMS, solution applicative unifiée), et bénéficie du support de la majorité des grands acteurs des télécommunications et IP qui permettront au SIP de gagner en stabilité et en maturité.

4.3. Conclusion

Dans ce chapitre on a essayé de faire un résumé sur les technologies utiles et nécessaire à connaitre dans le cas d'une conception d'un SVI mais on n'a pas cité autres outils qui ont aussi un rôle primordial dans le développement des SVI tel que les langages VoiceXML et SALT et les PABX.

Les langages VoiceXML et SALT ce sont des langages de balaise pour le développement des applications d'input/output de la voix, ils ont comme but de :

- Minimise les interactions client/serveur en définissant plusieurs interactions par document.
- Isole les auteurs d'applications des détails de bas niveau propres à la plateforme.
- Sépare le code d'interaction avec l'utilisateur (dans le langage VoiceXML) de la logique des services.
- Favorise la portabilité des services entre les plateformes d'implémentation. Le langage VoiceXML est commun aux fournisseurs de contenu, aux fournisseurs d'outils et aux fournisseurs de plateformes.
- Est facile à employer pour des interactions simples et offre néanmoins des fonctionnalités pour gérer des dialogues complexes.

PABX signifie Private Automatic Branch eXchange (autocommutateur téléphonique privé). Le PABX fonctionne à la base pour des lignes de téléphone traditionnelles (analogiques), et non pour des lignes de téléphonie en Voix sur IP (illimité via internet). Pour que le PABX accepte la voix sur IP il suffit d'ajouter une carte VoIP.

Chapitre V :
Expérimentations et résultats

5.1. Introduction

Depuis toujours, les scientifiques se penchent dans leurs travaux vers le domaine de la reconnaissance automatique de la parole, qui reste toujours un domaine fascinant pour eux comme pour le large publique, à travers ses applications multiples dans la vie quotidienne des gens, à titre d'exemple : communiquer avec une machine via une voix humaine, allumer ou éteindre tel ou tel appareil électrique sans se lever ; éviter de taper pendant des heures et des heures sur un clavier en se contentant de dicter.

L'homme est par nature paresseux, une telle technologie a toujours suscité chez lui une part d'envie et d'intérêt, ce que peu d'autres technologies ont réussi à faire.

Le secteur de la reconnaissance automatique de la parole est en pleine croissance due à la technologie actuelle. Où les commandes vocales et le dialogue homme machine sont possible sans être présent devant la machine commandée ; c'est le cas de commander un café avant de revenir à la maison et sans l'intervention d'une personne ; préparer un envoi administratif avant d'être présent au bureau ; consulter son compte bancaire en dialoguant avec la machine, toutes ces applications et autres, sont en marche ou en phase de réalisation.

La problématique posée dans notre projet est en effet décomposable en deux parties, la première partie est comment peut on consulter notre compte bancaire en utilisant un dialogue homme/machine et la deuxième partie c'est comment faire tout cela sans être présent à la banque ?

5.2. Coup d'œil sur les plateformes existantes

En premier lieu on a posé la question comment peut-on faire un appel distant et que la machine réceptrice ou l'ordinateur récepteur enregistre ce qu'on a dit ?

En deuxième lieu, une fois la machine réceptrice, ou ordinateur récepteur a enregistré ce qu'on a dit comment il peut le reconnaitre ?

Pour répondre à ces questions et pour réaliser un tel projet on a fait un survol sur les technologies et les outils existant dans le cadre de la reconnaissance de la parole et leur utilisation avec les serveurs vocaux interactifs, et le résultat était étonnant, pleins de logiciels et pleines de recherches dans ce domine ont été fait, sans oublier les nouvelles technologies dans le cadre d'envoi audio sur réseau (VOIP), on les résume dans ce qui suit :

- Pour l'appel distant et comme on a évoqué dans le chapitre 3, les technologies les plus utilisables sont H323 et SIP, et dans ce cadre et avec la plateforme utilisée (le langage de programmation C#, système d'exploitation Windows) on a essayé de créer une bibliothèque pour le H323 et on est arrivé à 80% de la réaliser avec quelques erreurs de bug.

  Concernant le SIP on a trouvé une bibliothèque très performante créée par la société OZEKI dont le nom est « *OZEKI VoIP SIP SDK* » et qui traite tous les types d'appels distants «audio, vidéo » et quelque soit le dispositif « pc, téléphone portables 3G, téléphone IP » la seule contrainte avec cette bibliothèque est qu'elle n'est pas gratuite et la version démo ne dure que quelque jours.

- Les outils de la reconnaissance de la parole les plus célèbres et les plus utilisés jusqu'à nos jours sont le HTK et le Sphinx.

  - ✓ Le HTK en anglais (The Hidden Markov Model Toolkit) est un outil de la construction des modèles de Markov cachés, dont les premières utilisations étaient pour les recherches de la reconnaissance automatique de la parole, bien que maintenant il est utilisé dans d'autres domaines tel que les recherches sur la synthèse vocale, la reconnaissance des caractères et le séquençage de l'ADN. Le HTK est un ensemble de bibliothèques écrit dans le langage C, et qui contient pleins des documentations et des exemples[22].
  - ✓ Sphinx est un projet lancé par l'université Carnegie Mellon (CMU) dans le but de concevoir un environnement pour la recherche dans le domaine de la reconnaissance automatique de la parole. CMU Sphinx 4 est une librairie de classes (en langage java). Sphinx est un système de RAP basé sur les Models de Markov Cachés (HMM) présente un ensemble d'outils de reconnaissance vocale (voir figure V.1) flexibles modulaires et extensibles formant un véritable banc d'essais et un puissant environnement de recherche pour les technologies de reconnaissance automatique de la parole.

[22] http://htk.eng.cam.ac.uk/

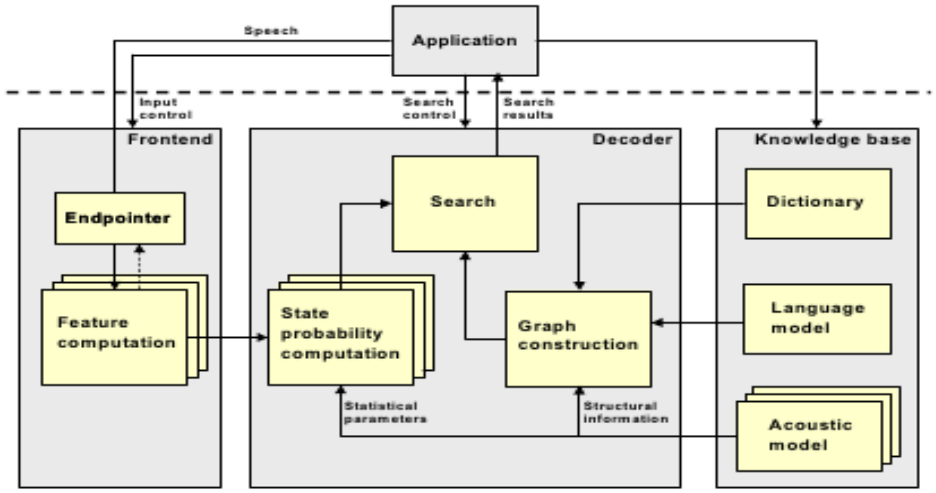


Figure V. 1 ***Architecture du CMU Sphinx-4.***

Comme notre plateforme est (le langage de programmation C# et le système d'exploitation Windows) qui nous ne permet pas d'utilisé ni le HTK ni le sphinx, on a préférer de réaliser notre propre application, en réutilisant d'autres bibliothèques qu'on a trouvé sur le net (Accord.NET).

5.3. La plateforme utilisée :

5.3.1. DOTNET

.NET (prononcez «Dotnet») est un standard proposé par la société Microsoft, pour le développement d'applications d'entreprises multi-niveaux, basées sur des composants. Microsoft .NET constitue ainsi la réponse de Microsoft à la plate-forme J2EE de Sun. La plate-forme .NET a été élaborée en s'appuyant sur une communauté d'utilisateurs et a abouti à l'élaboration de spécifications. Ces spécifications ont été ratifiées par un organisme international de standardisation, l'ECMA (European Computer Manufacturers Association), ce qui en fait un standard. Ainsi l'effort de standardisation a permis l'émergence de plates-formes portées par des entreprises tierces et disponibles sous un grand nombre de systèmes d'exploitation[23].

On parle généralement de «Framework» (traduisez «socle») pour désigner l'ensemble constitué des services (API) offerts et de l'infrastructure d'exécution. Le framework .NET comprend notamment :

- ✓ **L'environnement d'exécution :**

[23] http://www.commentcamarche.net/contents/254-net-introduction

- un moteur d'exécution, appelé CLR (Common Language Runtime), permettant de compiler le code source de l'application en un langage intermédiaire,
- baptisé MSIL (Microsoft Intermediate Language) et agissant telle la machine virtuelle Java. Lors de la première exécution de l'application, le code MSIL est à son tour compilé à la volée en code spécifique au système grâce à un compilateur JIT (Just In Time).
- un environnement d'exécution d'applications et de services web, appelé ASP .NET ;
- un environnement d'exécution d'applications lourdes, appelé WinForms.[24]

✓ **Des services**, sous forme d'un ensemble hiérarchisé de classes appelé Framework Class Library (FCL). La FCL est ainsi une librairie orientée objet, fournissant des fonctionnalités pour les principaux besoins actuels des développeurs. Le SDK (Software Development Kit) fournit une implémentation de ces classes.[25]

5.3.2. Le langage de programmation C#

C# (C sharp) est un langage orienté objet élégant et de type sécurisé qui permet aux développeurs de générer une large gamme d'applications sécurisées et fiables qui s'exécutent sur le .NET Framework. Vous pouvez utiliser C# pour créer, entre autres, des applications clientes Windows traditionnelles, des services Web XML, des composants distribués, des applications client-serveur et des applications de base de données[26].

5.4. Partie I : l'appel distant entre Serveur et client :

Comment peut-on faire un appel distant et que la machine réceptrice ou l'ordinateur récepteur enregistre ce qu'on a dit ?

Cette partie a comme but de permettre deux points de communication (par exemple deux PC reliés sous forme d'un réseau LAN) de communiquer vocalement, c'est-à-dire que deux personnes chacun devant un PC et leur PCs sont reliés par une connexion réseau, peuvent dialoguer en utilisant cette application. Et un des deux PC enregistre ce que dit la personne qui utilise l'autre PC.

[24] Utilisé dans notre cas.
[25] On a évité son utilisation parce qu'il facilite la reconnaissance vocale et il n'est pas applicable pour la langue arabe.
[26] http://msdn.microsoft.com/fr-fr/library/z1zx9t92(v=vs.80).aspx

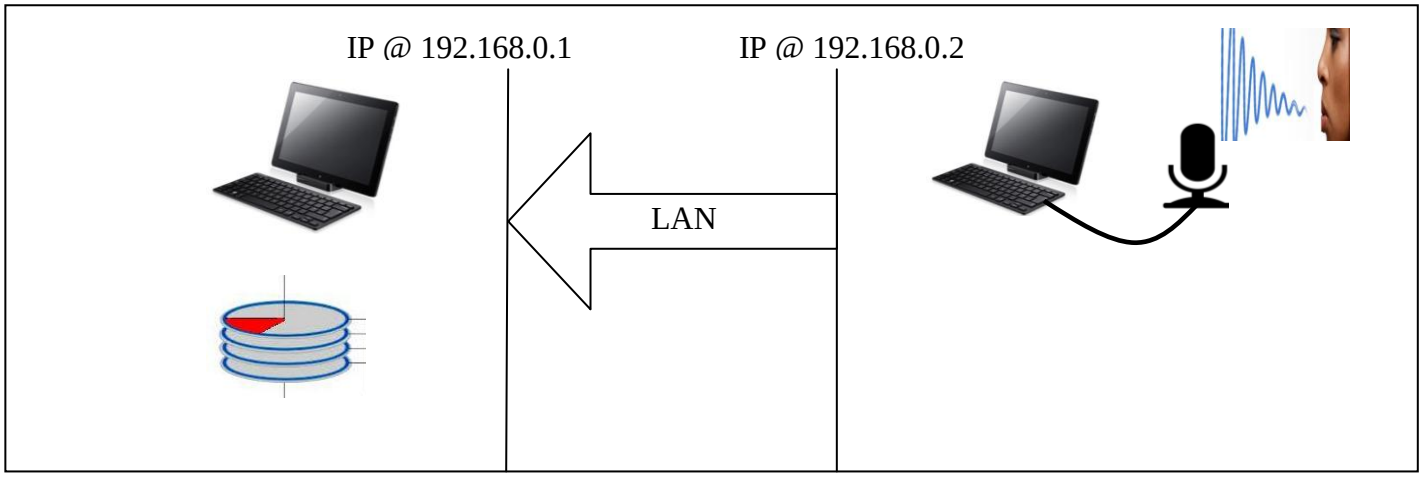


Figure V. 2 schéma représentatif de l'opération appel distant (client, serveur)

La réalisation de cette partie est décomposée en deux phases (deux applications).

5.4.1. PHASE 1- application serveur :

Cette application a le rôle d'ouvrir un port de communication entre les deux points de communication, et d'enregistrer les données reçues dans un répertoire spécifique pour être reconnues par une autre application.

Cas d'utilisation :

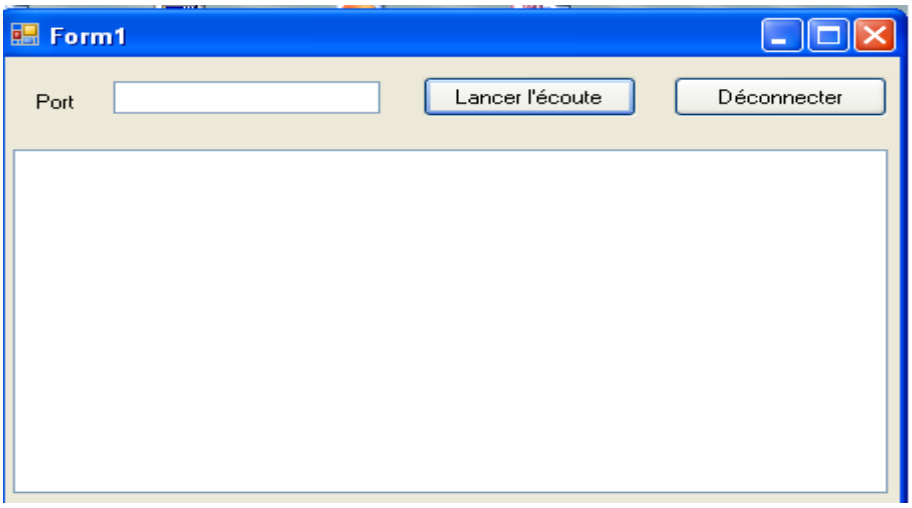


Figure V. 3 ***interface de l'application serveur.***

d'apres le WindowForm de la Figure V.3 , qui représente l'application serveur on doit indiquer un numéro de port puis on clique sur le bouton « lancer l'ecoute » pour ouvrir le port indiqué et permettre au client de parler. Quand le client termine l'appel, l'application

serveur envoie à une application cliente sitée sur le meme PC un message qui indique la présence d'un fichier d'extension wav prét à etre reconnu, cette dernière est similaire dans son interface à l'application d'appel, et c'est elle qui procède à la reconnaissance du mot parlé. (on va la voir dans la deuxieme partie)

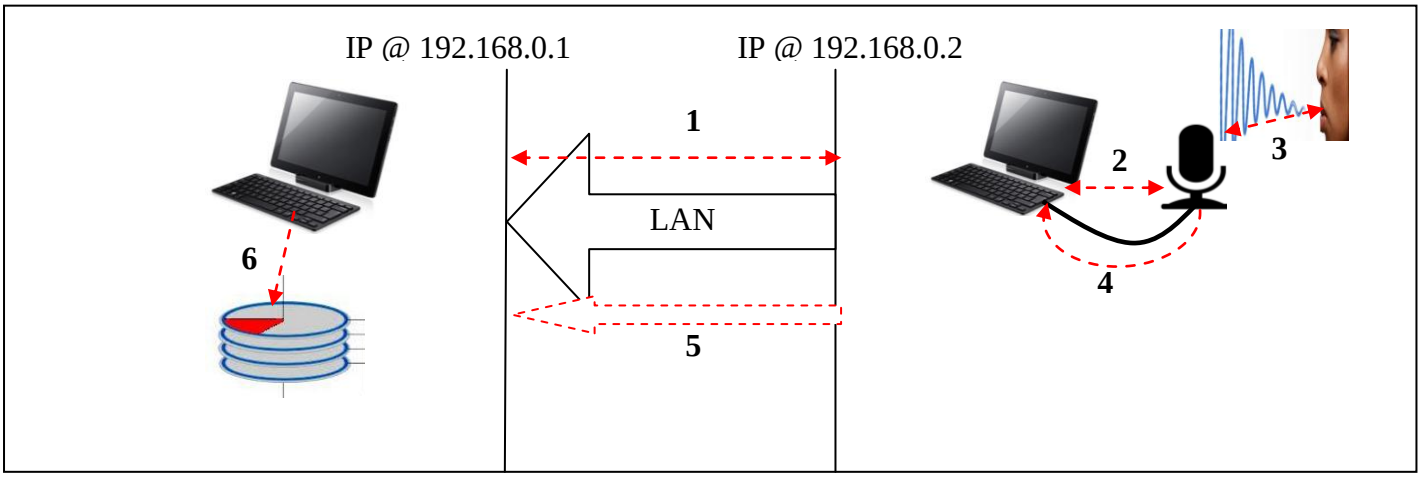


Figure V. 4 schéma démonstratif de l'opération appel distant (client, serveur)

5.4.2. PHASE 2- Application cliente :

Elle doit être installée dans les deux points de communication, et son rôle est de :

5.4.2.1. Cas d'appel

1. Joindre les deux points pour permettre l'appel vocal.
2. Détecter le périphérique d'acquisition
3. Acquérir la voix et le mettre dans un buffer.
4. compresser le buffer, et l'envoyer au deuxième point[27].
5. Si l'appel n'est pas terminé aller à 3.

5.4.2.2. Cas de recevoir d'appel

1. Mettre les données reçues dans un buffer temporaire
2. Décompresser le buffer et le mettre dans un autre buffer.
3. Créer un FILE STREAM qui accumule les buffers décompressés dans un fichier temporaire de type Wave.

[27] La compression se fait via *G711 Encoder* une norme de compression audio de l'UIT-T

4. Une fois l'appel est terminé on ferme le FILE STREAM et sauvegarde le fichier Wave pour le préparer à la deuxième phase (phase de reconnaissance).

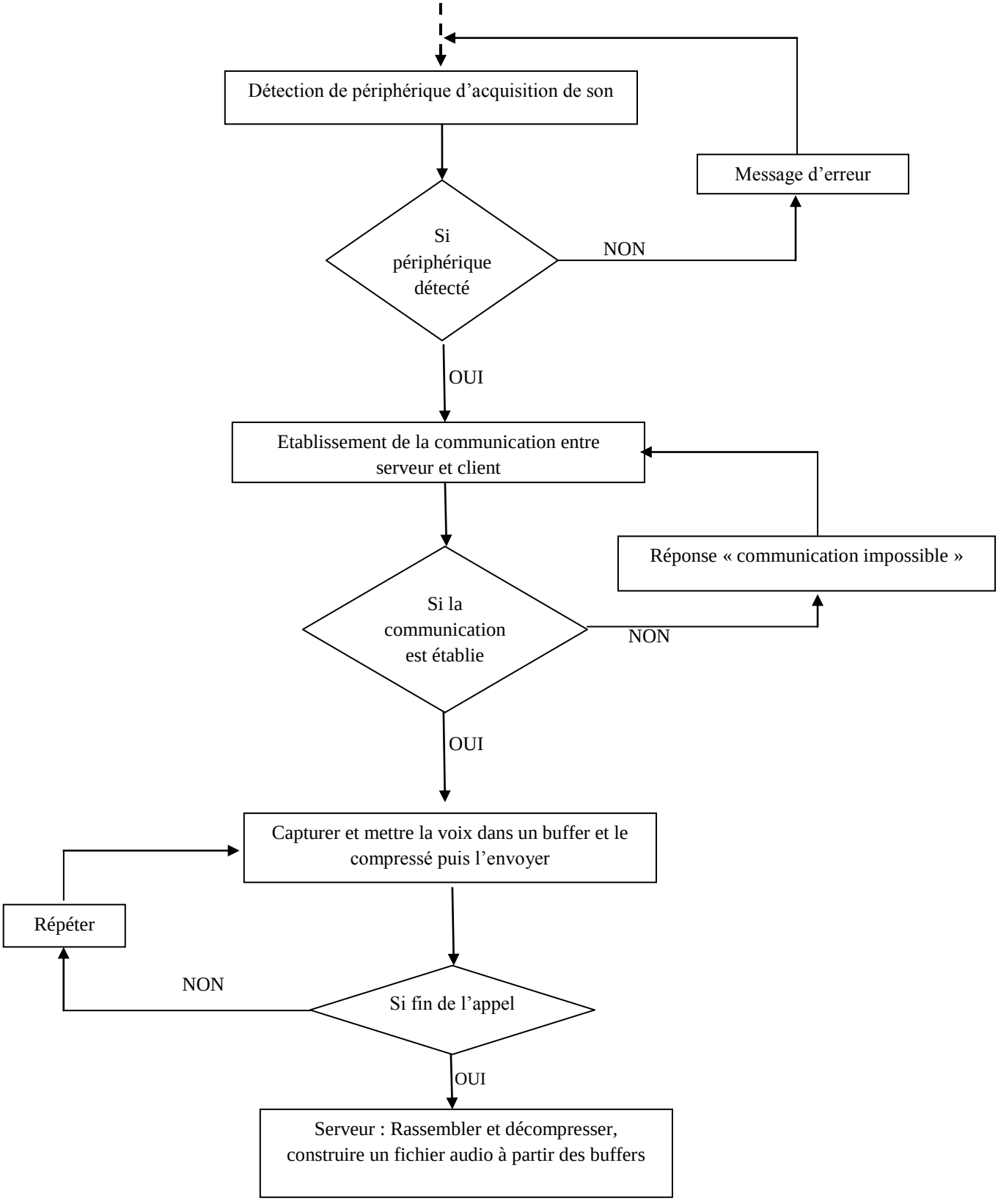


Figure V.5 Organigramme de l'opération d'appel

Cas d'utilisation :

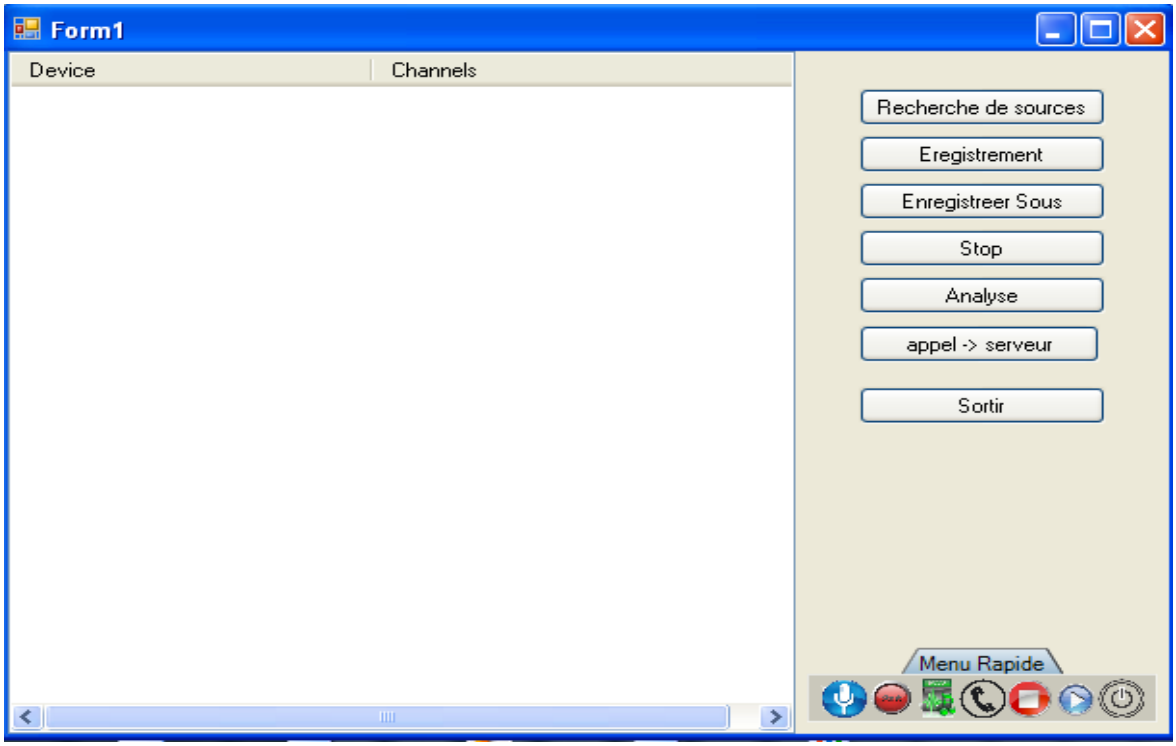


Figure V. 6 interface de l'application client (appelante).

Donc et d'après la (Figure V.5) si on veut faire un appel au serveur[28], on doit en premier lieu chercher la source d'acquisition en cliquant sur le bouton recherche de ressources et la liste des microphones connectés au PC s'affichera dans le canevas blanc, on choisit une de ces ressources puis on clique sur le bouton « appel->serveur » pour établir un appel, un message s'affichera pour nous indiquer que notre voix s'enregistre sur le serveur.

5.5. Partie II : La reconnaissance automatique de la parole

5.5.1. Reconnaissance Automatique des chiffres

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 des voyelles et 28 sont des consonnes, les phonèmes arabe se distinguent par la présence de deux classes

[28] L'adresse IP de serveur doit être 192.168.1.1

qui sont appelées pharyngales et emphatiques. Ces deux classes sont caractéristiques des langues sémitiques comme l'hébreu [H. Satori et al].

Dans son travail similaire au notre [H. Satori et al.] a utilisé le CMU Sphinx comme un utilitaire de développement et de conduite des applications de recherches dans la reconnaissance de la parole. Et d'après ses résultats qu'indiquent un taux de reconnaissance entre 80% et 83% avec 6 locuteurs (3 hommes et 3 femmes). On a décidé de réaliser un système de RAP pour les 10 chiffres arabe sans l'utilisation de CMU Sphinx, dont la différence est la base de connaissance (ou la base de donnée) qui est extensible, en se basant sur la programmation pure avec l'aide de la bibliothèque Accord.net.

L'idée est inspirée d'une application réalisée par [C. de Souza][29], qui consiste en la reconnaissance de l'écriture manuscrite avec la possibilité d'ajouter des nouveaux caractères écrits après la mise en œuvre de l'application. Donc une base de données avec le minimum des caractères reconnus qui s'étende pendant l'utilisation de l'application.

L'avantage de notre travail par rapport à celui de [H. Satori et al.], est que le résultat serait une application prête à l'utilisation et à la mise en œuvre, avec une base de données qui se peut s'étendre pendant l'utilisation, cela signifie une reconnaissance multi-langues ou même multi-dialectales, par exemple on peut ajouter les sons suivants « deux, اثنان, SIN (Amazigh) » pour être reconnu comme le chiffre « 2 » ; aussi préparer une base de reconnaissance éligible d'être utilisée par les deux autres modèles de la RAP (le modèle phonétique et le modèle de langage).

- **Critique de l'utilisation du CMU sphinx**

Le CMU Sphinx a besoin pour son installation de :

- Java 2 SDK, Standard Edition 5.0.[30]
- Java Runtime Environement (JRE)
- Les différentes librairies qui composent Sphinx-4.
- Ant : L'outil pour faciliter la compilation en automatisant les taches répétitives.[31]

[29] http://crsouza.blogspot.com/2010/03/hidden-markov-sequence-classifiers-in-c.html
[30] Sun Microsystems. Available: http://java.sun.com.
[31] http ://ant.apache.org

Qui rend la tâche d'installation un peu critique, surtout dans le cas de la mauvaise configuration de l'un de ces composants.

5.5.2. Le prétraitement et la classification du signal audio

C'est la partie la plus difficile parce qu'elle se réalise par quatre étapes, qui se déroulent l'une à suite de l'autre :

1. Segmentation de fichier audio.
2. Transformer le fichier en un autre fichier binaire qui représente la transformée rapide de Fourier
3. Extraire les coefficients de Mel à partir de ce dernier fichier.
4. Classifier le fichier (selon le modèle de Markov caché) et rendre compte à l'application serveur.[32]

5.5.2.1. Le prétraitement du signal audio et la segmentation

5.5.2.1.1. Le spectrogramme et la FFT

Figure V. 5 interface démonstratif de la phase prétraitement

La figure V.6 représente une interface supplémentaire pour démontrer le prétraitement élaboré sur le fichier reçu, où le premier graphe c'est la représentation de la voix prononcée

[32] Si le chiffre reconnu le message envoyé au serveur sera le chiffre si non le message « chiffre non reconnu »

par le locuteur dans sa forme spectrale. Le deuxième c'est pour un graphe qui représente la transformée rapide de Fourier (FFT) appliquée sur le fichier audio.

En premier lieu on doit segmenter le fichier en un nombre défini de segments (48 pour notre cas).Puis, on applique la transformée de Fourier sur chaque segment pour y avoir le graphe suivant.

Figure V. 6 interface démonstratif de la phase prétraitement (graphe représentatif de la FFT du mot سبعة).

5.5.2.1.2. La génération du MFCC

Une fois la transformée de Fourier est terminée, on passe à l'extraction des coefficients de Mel, utilisant la bibliothèque MFCC, qui va faire sortir les coefficients de Mel en se basant sur les résultats du calcul de la FFT.

Et pour plus de démonstration on a ajouté une zone qui nous affichera les résultats des coefficients du signal audio.

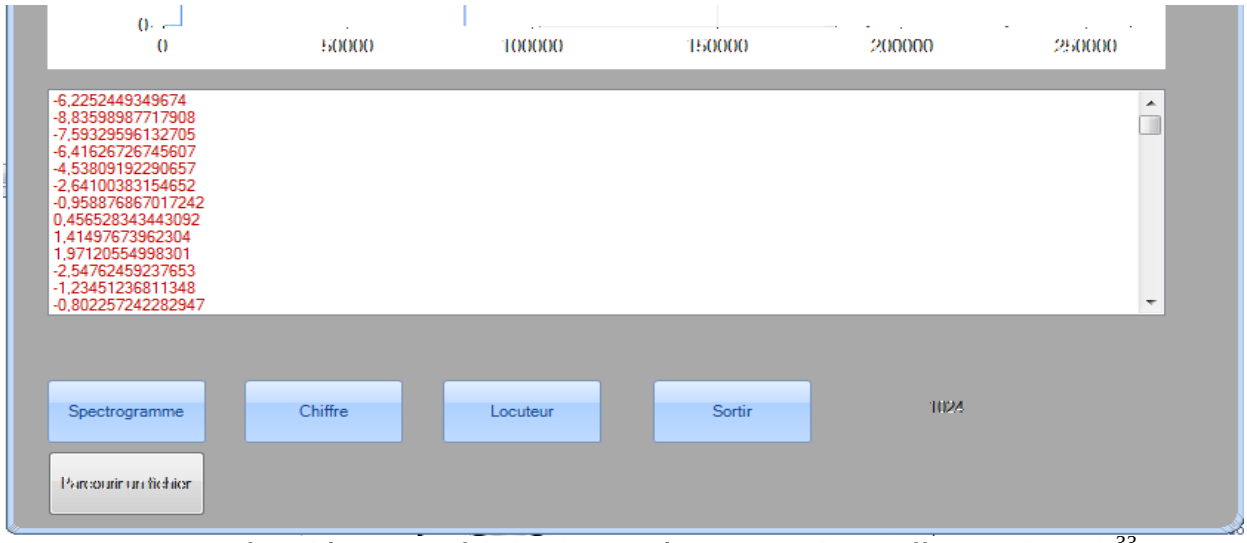


Figure V. 7 *interface démonstratif de la phase prétraitement (les coefficients de Mel)*[33] .

5.5.2.2. Classification de séquences :

5.5.2.2.1. La création du modèle de Markov caché

Jusqu'à présent, nous avons vu c'est quoi un modèle de Markov caché, sur le plan théorique mais dans la pratique et avec la précision de la problématique se résume comme suit :

On considère que nous voulons créer des séquences cachées où l'utilisateur de l'application a le droit de générer ces séquences cachées selon son besoin par exemple des séquences de chiffres (c'est-à-dire une séquence pour un chiffre) ou des séquences pour les nombres inferieurs à 100 où des séquences pour les nombres inferieurs à 1000 …etc.

Pour la mise en œuvre effective des modèles et des formules énoncées dans le chapitreII ; on a utilisé la classe IHiddenMarkovModel de Accord.NET Framework, qui est représentée par la figure suivante :

[33] La génération des mfcc se fait sur des vecteur de 512 pour chaque mfcc

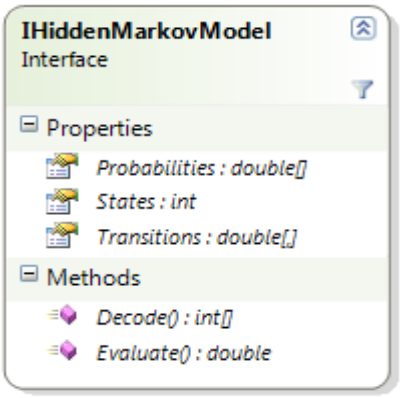


Figure V. 8 ***représentation de la classe IHiddenMarkovModel.***

Où Probabilities c'est le vecteur de probabilités d'état initiale ; States est le nombre d'états ; et Transitions c'est la matrice de transition.

Et cette classe a deux sortes d'implémentation, HiddenMarkovModel pour le cas d'une chaine de markov Cachée et une implémentation générique HiddenMarkovModel <TDistribution>:

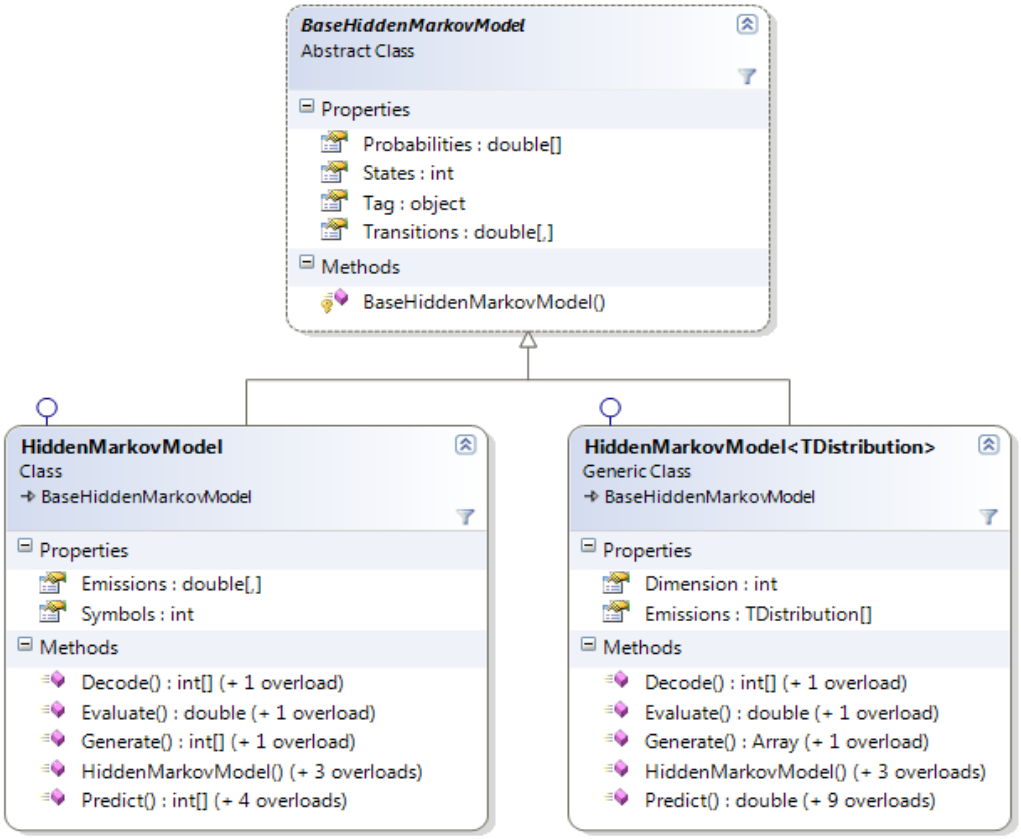


Figure V. 9 ***la classe IHiddenMarkovModel et ses héritées.***

Comment on peut l'utiliser ?

La Création d'un modèle de Markov caché et le calcul de la probabilité pour une séquence de nombre aléatoire se fait de la manière suivante :

```
// Création d'un modèle de Markov caché avec paramètre aléatoire probabilités
HiddenMarkovModel hmm = new HiddenMarkovModel(states: 3, symbols: 2);

// Créer une séquence d'observation allant jusqu'à 2 symboles (0 ou 1)
int[] observationSequence = new[] { 0, 1, 1, 0, 0, 1, 1, 1 };

// Evaluer son log-vraisemblance. Le résultat est -5,5451774444795623
double logLikelihood = hmm.Evaluate(observationSequence);
```

Un HMM a comme rôle soit de décoder ou d'apprendre ou de classifier et comme notre cas est un cas de classification on passe directement aux classes utilisée pour la classification.

5.5.2.2.2. La classification des séquences

De classifier une séquence selon le modèle de Markov caché, c'est de trouver la probable séquence cachée générée par la séquence observée c'est-à-dire de calculer ou d'évaluer les probabilités de la séquence observée sachant les séquences cachées, et le choix se fait selon deux cas soit on choisit le maximum entre eux ou le minimum.

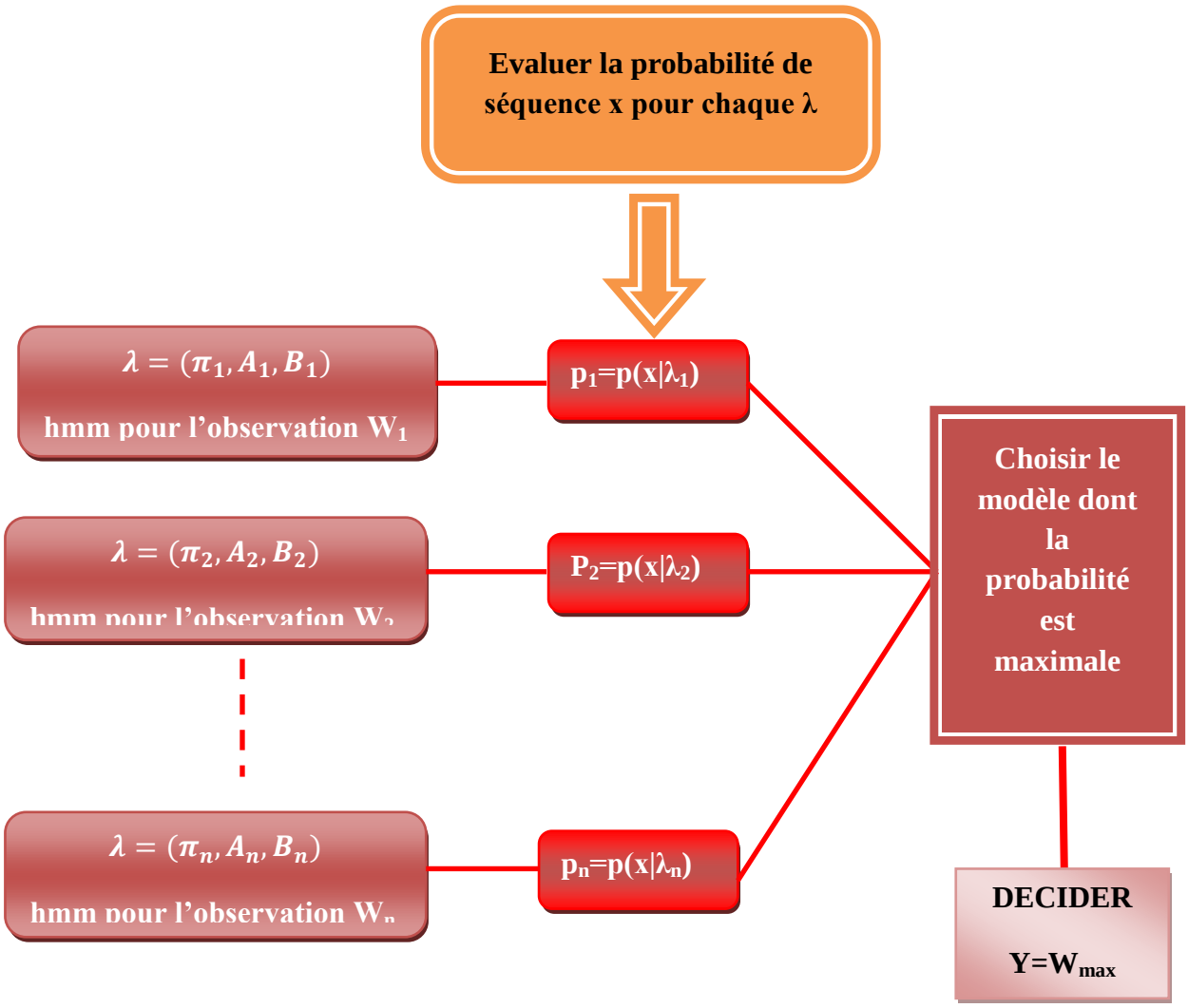


Figure V. 10 mécanisme de classification selon HMM.

Accord.Net nous offre pour la classification une classe qui nous permet la classification de séquence, dont le nom est IHiddenMarkovClassifier.

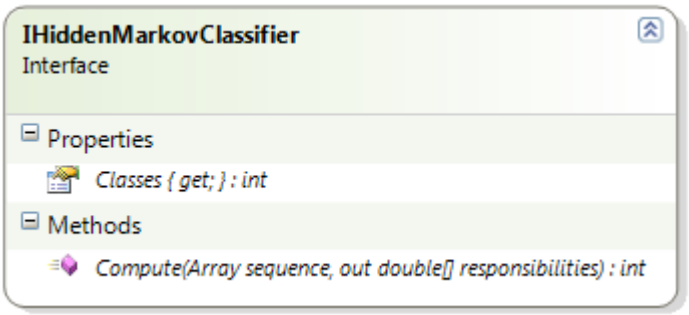


Figure V. 11 ***la classe IHiddenMarkovModelClassifier.***

Et de même que la création de la chaine de Markov cachée on a deux type d'implémentation de classificateurs soit pour les séquences déscrètes ou génériques.

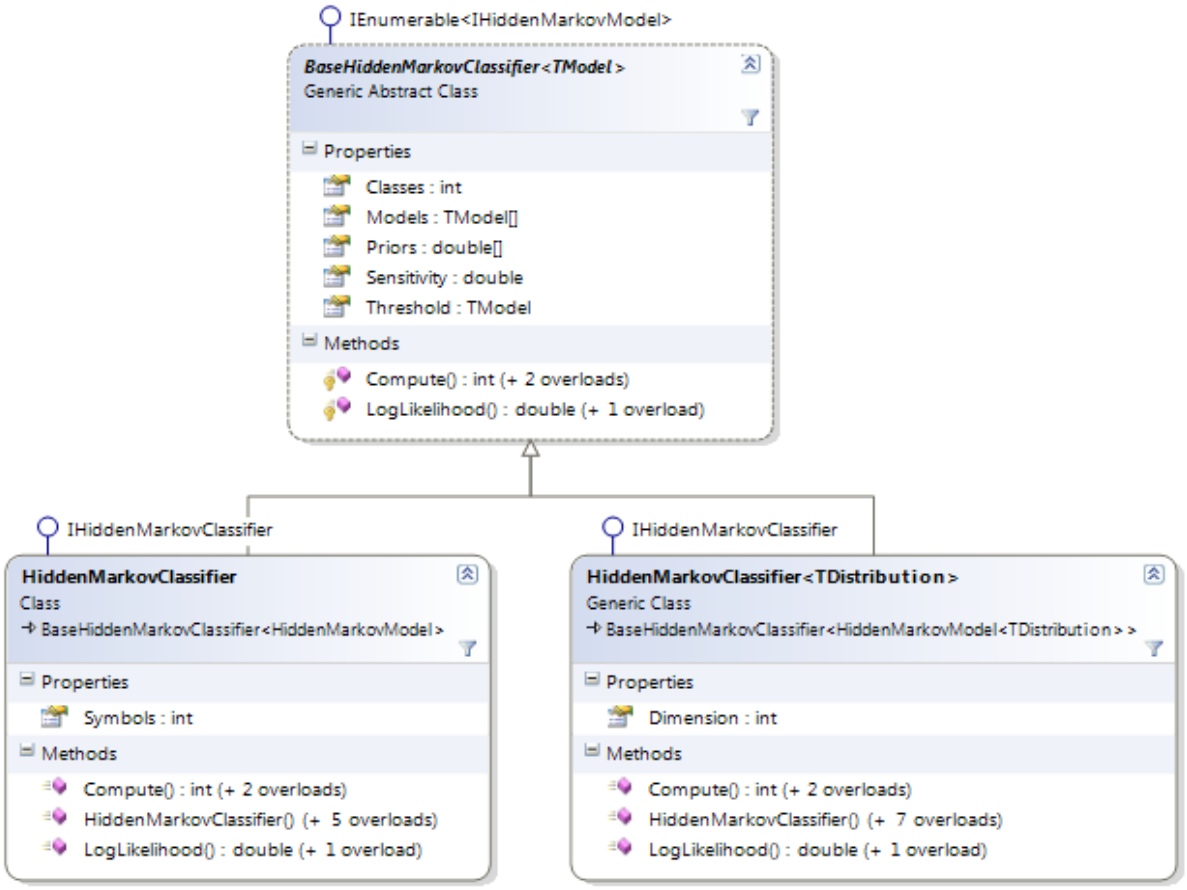


Figure V. 12 ***la classe IHiddenMarkovModelClassifier et ses héritées.***

L'utilisation de ses classes se fait de la manière suivante :

Exemple démonstratif sur l'utilisation de la classe HiddenMarkovModelClassifier dans le langage de programmation C#

```
//déclaration des états cachés pour les classe à reconnaitre supposant 3 classes définies comme suit
int[][] inputSequences =
{
   // supposant une classe 1 commence par zéro et termine par zéro
   new[] { 0, 1, 1, 1, 0 },
   new[] { 0, 0, 1, 1, 0, 0 },
   new[] { 0, 1, 1, 1, 1, 0 },
```

```
    //supposant une  classe 2 commence par 2 et termine par1
    new[] { 2, 2, 2, 2, 1, 1, 1, 1, 1 },
    new[] { 2, 2, 1, 2, 1, 1, 1, 1, 1 },
    new[] { 2, 2, 2, 2, 2, 1, 1, 1, 1 },

    // supposant la  classe 3 commence  termine par1
    new[] { 0, 0, 1, 1, 3, 3, 3, 3 },
    new[] { 0, 0, 0, 3, 3, 3, 3 },
    new[] { 1, 0, 1, 2, 2, 2, 3, 3 },
    new[] { 1, 1, 2, 3, 3, 3, 3 },
    new[] { 0, 0, 1, 1, 3, 3, 3, 3 },
    new[] { 2, 2, 0, 3, 3, 3, 3 },
    new[] { 1, 0, 1, 2, 3, 3, 3, 3 },
    new[] { 1, 1, 2, 3, 3, 3, 3 },
};
// symboles pour les classes précédentes
int[] outputLabels =
{
    /* classe 1: */ 0, 0, 0,
    /* classe 2: */ 1, 1, 1,
    /* classe 3: */ 2, 2, 2, 2, 2, 2, 2, 2
};

// supposant qu'il y a une seule typologie pour les trois classes

ITopology forward = new Forward(states: 3);

// créer hiddenMarkovclassifier avec la typologie définie
HiddenMarkovClassifier classifier = new HiddenMarkovClassifier(classes: 3,
    topology: forward, symbols: 4);
// créer un algorithme de reconnaissance pour chaque modèle
var teacher = new HiddenMarkovClassifierLearning(classifier,
    modelIndex => new BaumWelchLearning(classifier.Models[modelIndex])
    {
        Tolerance = 0.001, // l'iteration se fait jusqu'à log-likelihood
                           // Serait inferieur à 0.001
        Iterations = 0      });
// lancer la méthode de reconnaissance
double error = teacher.Run(inputSequences, outputLabels);
// avoir les résultat
int y1 = classifier.Compute(new[] { 0, 1, 1, 1, 0 });    // output is y1 = 0 c-à-d que la séquence { 0, 1, 1, 1, 0 }∈ classe 1
int y2 = classifier.Compute(new[] { 0, 0, 1, 1, 0, 0 }); // output is y2 = 0 c-à-d que la séquence{ 0, 0, 1, 1, 0, 0 }∈ classe 1
int y3 = classifier.Compute(new[] { 2, 2, 2, 2, 1, 1 }); // output is y3 = 1 c-à-d que la séquence{ 2, 2, 2, 2, 1, 1 }∈ classe 2
int y4 = classifier.Compute(new[] {2,2,1,1 });// output is y4= 1 c-à-d que la séquence{2,2,1,1 }∈ classe 2
```

```
int y5 = classifier.Compute(new[] { 0, 0, 1, 3, 3, 3 }); // output is y5 = 2 c-à-d que la séquence { 0, 0, 1, 3, 3, 3 }∈ classe 3
int y6 = classifier.Compute(new[] { 2, 0, 2, 2, 3, 3 }); // output is y6 = 2 c-à-d que la séquence{ 2, 0, 2, 2, 3, 3 }∈ classe 3
```

Dans notre cas la base de données de chaque ensemble de séquences cachées est extensible, c'est-à-dire que l'utilisateur peut ajouter d'autres séquences pour certaines valeurs pour enrichir la base de données et couvrir un grand nombre des cas de prononciation d'un nombre quelconque.

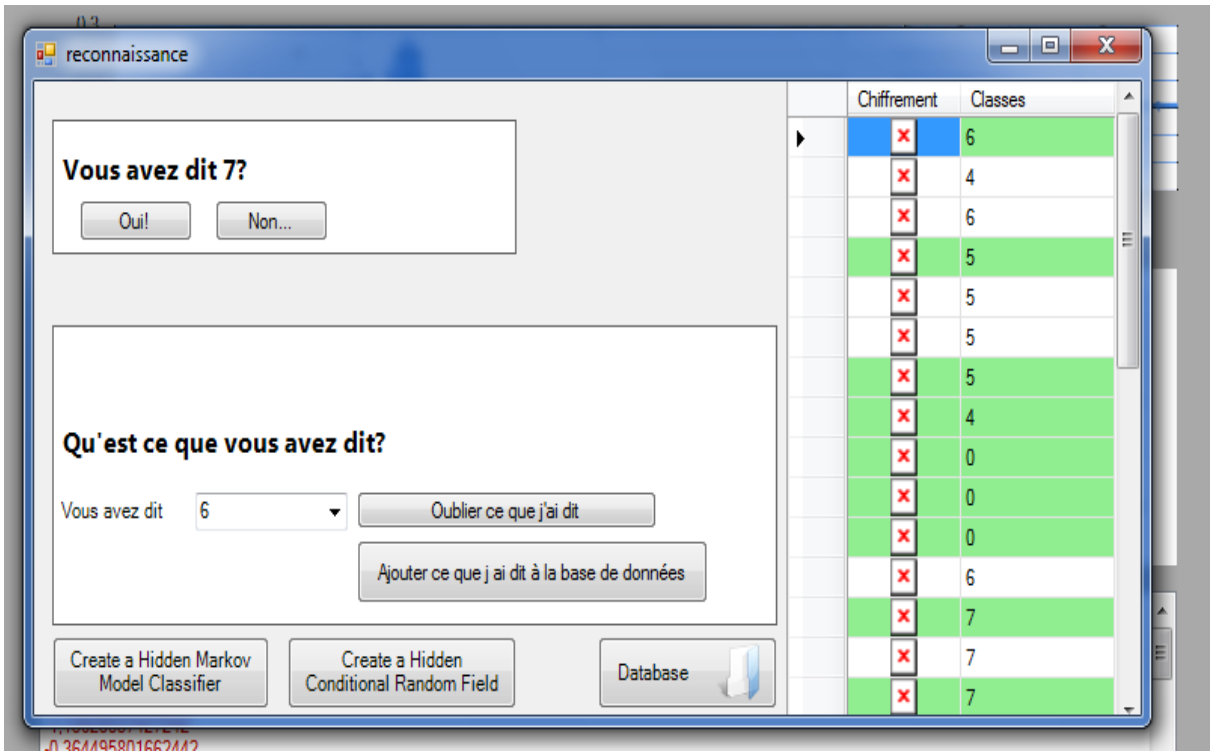


Figure V. 13 ***partie de l'applicatif de reconnaissance (classification).***

5.6. Conclusion :

De construire une Application de VOIP est simple si on a les outils et les composants exacts ; surtout avec l'apparition des Smartphones et leurs systèmes d'exploitation tels que (Androïde, AmigaOS, AtheOS etc.…). Ces systèmes qui offrent la possibilité de développer des applications, émulant le développement sous les systèmes d'exploitation ordinaires. Mais cette simplicité n'est pas possible si on veut démarrer du zéro et c'est à cause de cela, qu'on a trouvé des difficultés avec la programmation VOIP ; et notre conseil pour ceux qui veulent travailler avec VOIP d'essayer l'utilisation de la technologie SIP au lieu du H323, parce que

c'est l'avenir du VOIP, en plus tous les laboratoires de développement s'orientent vers cette technologie qui signifie l'augmentation des outils désignés à cette technologie

Pour la reconnaissance automatique de la parole, et comme on n'a pas utilisé ni l'HTK ni le Sphinx (connus comme des moteurs de reconnaissance) ; mais d'après les documents lus le plus conseillé est le Sphinx pour la raison qu'il est plus précis que HTK (à partir de ses version 3.0 est 4.0 Sphinx est plus précis que HTK).

Mais pour ceux qui ils veulent travailler avec DOTNET, en utilisant les HMMs on les conseille de télécharger Accord.NET parce qu'elle offre des bibliothèques contenant tous types des modèles (reconnaissance, apprentissage, évaluation) ; son avantage est que l'utilisateur peut de bien comprendre ce qu'il fait ; et son intervention est de manière directe et approfondi, et le résultat serait une application indépendante (application installable).

Conclusion générale et Perspectives

Conclusion générale et Perspectives

La recherche et le développement industriel en reconnaissance automatique de la parole ont été influencés par les progrès technologiques. Débutent avec les systèmes analogiques basés sur l'électronique, et avec le développement rapide de la micro-électronique et de l'informatique qui ont permis l'ouverture de nouveaux horizons pour ce domaine tant au niveau des techniques qu'au niveau des secteurs d'application.

Malgré ses performances croissantes, la reconnaissance de la parole, n'est pas un problème résolu, en particulier vis à vis de la modélisation du langage.

Notre mémoire contribue à l'amélioration des techniques de modélisation acoustique de la langue arabe, utilisant les chiffres arabes comme un corpus, avec la possibilité de l'enrichissement de la base de reconnaissance.

Au cours d'une application de type "*reconnaissance de mots isolés, multi-locuteur*", nous avons eu recours à un prétraitement segmental pour réaliser un système de reconnaissance.

La plate forme utilisée (l'utilisation de la bibliothèque Accord.Net) nous libère de la dépendance aux moteurs de reconnaissance (comme HTK et Sphinx) ; ce degré de liberté permet aussi à l'innovation dans nos prochains travaux et même de passer aux autres domaines de reconnaissance et de classification.

Dans le Chapitre 3, nous avons présenté une vue d'ensemble de la voix sur IP : les mécanismes de base du codage et du transport de la voix sur l'IP, les caractéristiques des codecs audio standards (G.711, G.726, G.729, G.723.1), les protocoles utilisés pour le transport des flux de vois sur l'IP (principalement RTP/RTCP). Les questions que nous avons dressées dans ce chapitre sont :

- La latence de la réponse qui est relative à l'efficacité d'utilisation de la bande passante et la qualité de la transmission, aussi affectée par le débit de codage ; peut avoir de délai à grand échelle qui serait pénible pour la réalisation d'un tel projet
- L'effet des caractéristiques du réseau (bande passante, hétérogénéité de l'infrastructure), qui peut influencer négativement sur la mise à niveau du projet.

Ce mémoire traite La reconnaissance automatique des chiffre arabes comme une initiation à la reconnaissance de grand nombre de vocabulaires de la langue arabe et de préparer un modèle acoustique efficace à la RAP continue arabe. Ainsi elle démontre la possibilité de construire un système de reconnaissance de la langue arabe, qui ne réfère pas à la transcription des mots arabe en des lettres romaines (cas de Sphinx).

Pour arriver à ce point il nous faut de grands efforts et une collaboration fortement couplée entre les experts en Syllabification arabe et les experts en RAP ; aussi d'utiliser d'autres modèles de reconnaissance ou l'hybridation entre les modèles pour avoir le modèle le plus convenable pour la langue arabe.

Dans ce travail on a rencontré certaines difficultés dues à la non disponibilité du matériel qui permet de tester la VOIP (GETKEEPER,) ; ce qui nous oblige à'utiliser un simple réseau entre deux PCs et des fois l'utilisation des machines virtuelles ; on a espéré aussi d'utiliser le SIP comme un protocole d'envoie audio, mais ce qu'on a trouvé sur le net concernant ce protocole et sa plate forme nous oblige à l'acheter.

Les résultats de la reconnaissance n'ont pas été très efficaces à cause de la mauvaise capture due au local d'expérimentation aussi la non exactitude des résultats fournis par l'étape de prétraitement et l'effet de perte de paquets dans l'envoie audio (qui est très fréquent lors de l'utilisation de la machine virtuelle).

Suite à ce qui précède, nous espérons d'avoir un grand nombre des projets locaux sur le SIP, ainsi que sur les phases de prétraitement du signal audio (FFT, MFCC,…), avec des états de sorties permettant l'innovation et l'amélioration de la phase de reconnaissance. Notre travail est une contribution à l'amélioration de la reconnaissance de la langue arabe pour l'aligner avec les autres langues (anglaise, japonaise, française,…). nous espérons avoir apporté un plus à cette langue.

Bibliographie

[Anne 2007] - Anne Bonneau et Yves Laprie. *Selective acoustic cues for French voiceless stop consonants*. Laboratoire Lorrain en Informatique et ses Applications. France. 2007 .

[M. Aubry 2000] - M. Aubry et F.Cellier . *Reconnaissance vocale sur plate-forme télécom Hewlett Packard*. Rapport de stage de 3ème année de l'ENS d'Électronique et de Radioélectricité de Grenoble. juin 2000.

[J. Baker 1975] – J. Baker. *Stochastic Modeling for Automatic Speech understanding*. PP 521-542. Academic Press.New York 1975.

[M. Bellanger 1980] – M. Bellanger .*Traitement numérique du signal Théorie et pratique*. édition N° 1. éditions Masson. 1980.

[J. Billa 2002]- J. Billa et al., *Audio indexing of arabic broadcast news*. IEEE international conference on acoustics, speech, and signal processing (ICASSP). 2002.

[M. Bisani 2003]- M. Bisani et H.Ney. *Multigram-based grapheme-to-phoneme conversion for lvcsr*. University of Technology Aachen. Germany. 2003.

[J. Bruno 1995]- J. Bruno. *Un outil informatique de gestion de Modèles de Markov Cachés, expérimentations en Reconnaissance Automatique de la Parole*. l'Institut de Recherche en Informatique de Toulouse. l'Université Paul Sabatier de Toulouse III. 1995.

[Calliope 1989] - Calliope et groupe d'auteurs. *La parole et son traitement automatique* . Collection Technique et scientifique des télécommunication, CENT/ ENST. édition Masson. I Paris, 1989.

[M .Chetouani 2004] M. Chetouani , *A new nonlinear speaker parameterization algorithm for speaker identification,* in: Proceedings of the ISCA Tutorial and Research Workshop on Speaker and Language Recognition. Laboratoire des Instruments et système d'Ile de France, Université Paris IV, 2004.

[Cooley et tukey 1965] - Cooley and Tukey J.W. An algorithm for the machine calculation of complex fourier series. 1965.

[Cornu 2007] - Céedric CORNU. *Extraction de signaux et Caractérisation des lois de phase instantanée Application aux modulations non-linéaires.* Université de Bretagne Occidentale, 2007.

[G.Pujolle 1998] - G.Pujolle. *Les réseaux*. édition, Eyrolles, 1998.

[H.Satori et al. 2010] - H. Satori, M. Harti, et N. Chenfour. *Système de Reconnaissance Automatiquede l'arabe basé sur CMUSphinx*. Université Sidi Mohamed Ben Abdellah Dhar El Mehraz Fès, 2010.

[J. Haton et al. 1991] - J. Haton, J. Pierrel, G. Pérennou,J. Caelen & J. Gauvain. *Reconnaissance automatique de la parole*. Dunod, 1991.

[H. Hermansky 1991] – H. Hermansky & Cox Jr, L. *Perceptual Linear Predictive* . IEEE ASSP Workshop, 1991.

[X. Huang et al. 2001] - X. Huang,A. Acero, & H. Hon. *Spoken language processing* . NJ, USA: Prentice Hall PTR Upper Saddle River, 2001.

[F. Jelinek 1976] – F. Jelinek. *Continous speech recognition bystatistical methods*. . Proceedings of the IEEE, 1976.

[G. Laboulais 2007] – G. Laboulais. *La compréhension de la parole chez le sujet presbyacousique porteur d'une prothèse auditive*. Elaboration d'un matériel de rééducation auditive. Nancy: Mémoire , 2007.

[LÊ Việt 2006]- LÊ Việt, Bắc. *Reconnaissance automatique de la parole*. UNIVERSITÉ JOSEPH FOURIER - GRENOBLE 1, 2006.

[J. Markel 1982] – J.Markel & A.Gray. *Linear prediction of speech*. Springer-Verlag, New York, USA 1982.

[A. Markov 1913]- A. Markov, *An example of statistical investigation in the text of eugene onyegin*, illustrating coupling of tests in chains, Proceedings of the Academy of Sciences of St. Petersburg, 1913.

[Rabiner 1989] - Rabiner Lawrence , *A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition*, In Proceedings of the IEEE, 1989.

[S. Seng 2010] - S.Seng. *Vers une modélisation statistique multi-niveau du langage*, L'Université de Grenoble. 2010.

[J. Taboada et al. 1994] – J. Taboada,S. Feijoo , R. Balsa, C. Hernandez . *Explicit estimation of speech boundaries*. Santiago de Compostela Univ. Spain. 1994.

[S. Umesh 1999]- S. Umesh, L. Cohen, D. Nelson. *Fitting the mel scale*. Phoenix Arizona (USA): ICASSP'99, vol. 1. 1999.

[Vaufreydaz2002] - Vaufreydaz Dominique. *Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue.* GRENOBLE I, 2002.

[Y.laprie 2002] - Y.laprie. *Analyse spectrale de la parole*. Laboratoire Lorrain en Informatique et ses Applications. France. 2002.