République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Mémoire de Magister

Présenté à l'Université de Guelma
Faculté des sciences et sciences de l'ingénierie

Département de : **Informatique**
Ecole **D**octorale **I**nformatique
Spécialité : **Informatique**
Option : Informatique

Présenté par : **Mr**: **Souadkia Abdelhak**

Reconnaissance automatique de la parole arabe: Approche évolutionniste

JURY

Président	**Pr**	**Haddadi Salim**	**Université de Guelma**
Rapporteur	**Pr**	**Mohamed Benmohammed**	**Université de Constantine**
Examinateur	**Pr**	**Seridi Hamid**	**Université de Guelma**
Examinateur	**Dr**	**Allaoua Chaoui**	**Université de Constantine**
Examinateur	**Dr**	**Azzedinne Bilami**	**Université de Batna**

2010

Résume

Le traitement automatique de la parole est un domaine d'étude active depuis les années 50 se qui explique sa richesse mais aussi sa difficulté. Il fait collaborer plusieurs disciplines et techniques. La complexité du signal de la parole qui résulte de l'interaction entre la production des sons et leur perception par l'oreille ce qui contribue dans la difficulté de la reconnaissance automatique de la parole, qui est devenue un sujet de recherche très intéressant.

La reconnaissance de la parole est un domaine de recherche qui inclut plusieurs approches et techniques, tels que les modèles de markov cachés MMC qui sont les plus répandus actuellement, les réseaux de neurones (RN), les modèles de markov cachés MCC associés au réseau de neurones (MMC-RN), la méthode DTW qui est la plus utilisée durant les années 70-80, et d'autres qui reposés sur l'hybridation des différentes techniques, bien que les approches évolutionnistes aussi s'impose comme un outil intéressant pour la reconnaissance de la parole.

Les coefficients MFCC restent ainsi un choix raisonnable pour représenter le signal de parole dans une tâche de reconnaissance automatique de la parole.

Ce travail vise à réaliser un système de reconnaissance automatique de la parole arabe par un algorithme génétique. Ce dernier est muni d'un algorithme de recherche tabou avec des entrées fournies par l'algorithme de k_plus proche voisin KPPV.

Notre première contribution consiste à utiliser un algorithme de K-plus proche voisin KPPV pour le filtrage de la population initiale et réduire sa taille. Les performances de la technique KPPV dans la classification justifie leur utilisation dans le filtrage de la population initiale. L'algorithme génétique évaluer la population des individus présentés par la l'algorithme KPPV et extraire toutes les solutions possibles.

Une seconde contribution consiste à utiliser un algorithme de recherche tabou qui assure la survie des individus voisins de la solution présentée par l'algorithme génétique, évite les optimaux locales et permettre à l'algorithme génétique d'être un système de reconnaissance de la parole arabe des performances considérables.

Les mots clés: reconnaissance automatique de la parole, les algorithmes génétiques, la langue Arabe, l'algorithme KPPV, recherche tabou, les coefficients MFCC

Abstract

Automatic speech processing is an area of active study since the 50, this explaining his wealth but also its difficulty. He includes several disciplines and technical. The complexity of the speech signal resulting from the interaction between sound production and perception by ear which contributes to the difficulty of automatic speech recognition, which has become a research topic very interesting.

The speech recognition is a research area that includes several approaches and techniques such as hidden Markov models HMM that are most popular now, neural networks (NN), the hidden markov models HMM associated with the neural networks (HMM-NN), the DTW method is most used during the years 70-80, and others who relied on the hybridization of different techniques. The evolutionary approach also stands out as an interesting tool for the speech recognition.

The MFFC coefficients are a reasonable choice to represent the speech signal in a task of automatic recognition of speech.

This work aims to develop a system for automatic recognition of Arabic speech by a genetic algorithm. This has a tabu search algorithm with inputs provided by the algorithm KNN.

Our first contribution is to use an algorithm K-nearest neighbor KNN for filtering the initial population and reduce its size. The performance of the technique KNN in classification justifies their use in filtering of initial population. The genetic algorithm estimating the population of individuals submitted by the KNN algorithm and extracts all possible solutions.

A second contribution is to use a tabu search algorithm which ensures the survival of neighboring individuals of the solution presented by the genetic algorithm. Tabu search algorithm avoids the local optimum and allows the genetic algorithm to be a system of speech recognition Arabic of interesting performance.

Keywords: automatic recognition of speech, genetic algorithms, the Arabic language, the KNN algorithm, tabu search, the MFCC coefficients.

Remerciements

Je voudrais exprimer ma profonde gratitude envers dieu tout puissant qui grâce à son aide, j'ai pu finir ce modeste travail.
Je voudrais remercier Pr. *Mohamed Benmohammed*, mon directeur de Thèse pour m'avoir encadré dans ce travail de thèse. Pour le sujet qu'il m'a proposé, pour la confiance qu'il m'a témoigné, pour ses précieux critiques, conseils, encouragements et son soutien constant qu'il veuille trouver dans ce mémoire ma profonde gratitude et mon grand respect.

Je remercie Pr S.*Haddadi* du département d'informatique à l'université de Guelma pour m'avoir fait l'honneur de présider mon jury de soutenance.
Je remercie Pr H.Seridi de l'université de Guelma pour avoir accepté de participer à mon jury de soutenance.

Mes remerciements les plus intenses sont adressés à monsieur A. Chaoui, Docteur à l'université de Constantine, à monsieur A. Bilami, Docteur à l'université de Batna, pour l'honneur qu'ils mon fait en acceptant, sans aucune hésitation de faire partie de ce jury et d'examiner ce travail.

Je tiens également à exprimer mes plus vifs remerciements à mes parents, pour leur soutien moral indéfectible, leurs encouragements et pour leur précieuse collaboration depuis de longues années et sans lesquels ce travail n'aurait pas été possible, qu'ils trouvent dans cette thèse ma profonde gratitude.

Sommaire

Chapitre 2 : Les algorithmes évolutionnaires

Chapitre 3 : La langue arabe

Chapitre 4 : Conception et réalisation

Liste de figures

Liste des tableaux

Introduction générale

Introduction générale

La parole est la manière naturelle et, en conséquence, la forme la plus intéressante et la plus commune de communication humaine. À la différence d'autres moyens électroniques de communication, les systèmes utilisant la parole offrent à l'utilisateur non entraîné un accès simple et naturel. Elle permet d'avoir un accès immédiat à une information sans avoir à parcourir toute une arborescence hiérarchique de menus. L'utilisation des raccourcis clavier ou des langages de commande existe toujours, mais la prononciation d'un seul mot peut remplacer jusqu'à une dizaine de commandes élémentaires effectuées à l'aide de touches fonctions ou de souris et représente ainsi un effort mnémotechnique moindre.

La simplicité et la souplesse de la parole comme un moyen de communication justifient les recherches effectuées dans ce domaine et la progression régulière, et le nombre de systèmes commercialisés et mis à la disposition du public. Et ces trente dernières années ont été témoin d'une évolution impressionnante dans le développement du traitement automatique de la parole.

Pour réaliser des systèmes de reconnaissance de la parole on trouve plusieurs obstacles tel que les techniques de reconnaissance utilisés dans la réalisations du système de reconnaissance de la parole et les difficultés de compréhension du langage naturel en parole spontanée qui limitent la mise en place d'un plus grand nombre de systèmes oraux et les résultats ne sont pas satisfaisantes à cent pour cent sans restriction de l'environnement. Une multitude d'algorithmes ont été élaborés pour améliorer la performance et la robustesse des systèmes RAP. Les algorithmes évolutionnaires et plus précisément les algorithmes génétiques constituent une des approche les plus intéressante dans ce domaine, ils fournissent des solutions proches de la solution optimale à l'aide des mécanismes de sélection, d'hybridation et de mutation, il est constaté que les solutions fournies par ces algorithmes sont généralement meilleures que celles obtenues par les méthodes plus classiques, pour un même temps de calcul.

Malgré le grand nombre des arabophones dans le monde, et malgré l'arrivée de produits commerciaux des systèmes RAP au grand public avec une bonne qualité et financièrement accessibles la reconnaissance automatique de la parole arabe est à ces débuts par rapport à d'autres langues, le manque de corpus oraux ou écrits ainsi que le manque de collaboration entre les laboratoires de recherches dans le monde arabe, rend la richesse de cette langue inaccessible à l'ensemble de la communauté scientifique, ainsi qu' au grand public. Pour enrichir cette langue et rendre accessible à l'ensemble de la communauté scientifique et au grand public dans le monde nous prendrons la responsabilité de réaliser

cette étude qui s'intègre dans le cadre du développement d'un système de reconnaissance automatique de la parole arabe avec une approche évolutionnaire, nous espérons apporter une contribution à notre chère langue.

Le but principal du travail décrit dans ce projet, est l'amélioration des performances des systèmes de reconnaissance automatique de la parole arabe en prendre comme objectif l'augmentation du taux de reconnaissance par apport les approches classiques. Nous utilisons un algorithme génétique mené par un algorithme de recherche tabou avec des entrées fournies par l'approche KPPV.

La thèse est structurée selon le plan suivant :

Le premier chapitre consacre à la reconnaissance automatique de la parole dans le quel nous présentons les techniques de paramétrisation du signal de la parole et les différentes modèles utilisés pour la réalisation des systèmes de RAP. Nous étudions au niveau du deuxième chapitre les concepts de base des algorithmes génétiques et ses applications dans le développement des systèmes de reconnaissance automatique de la parole arabe. Dans le chapitre trois nous décrirons la langue arabe. La conception et la réalisation du système de reconnaissance de la parole arabe sera illustrer au chapitre quatre. On clôture cette étude par une conclusion générale et un ensemble des perspectives.

.

Chapitre 1

Traitement automatique de la parole

1. Introduction

Le principe de reconnaissance automatique de la parole s'inscrit dans le domaine le plus général de la reconnaissance des formes. L'idée de base est d'apprendre des formes ou bien des statistiques sur les formes pour pouvoir le reconnaitre, la forme reconnue étant celle parmi toutes les formes apprises qui ressemble le plus à la forme inconnue.
Le traitement automatique de la parole est un domaine de recherche actif en croisement du traitement du signal et du traitement symbolique du langage. Cette discipline scientifique a connu depuis les années 60 une expansion fulgurante, liée au développement des moyens et des techniques de télécommunications.

Les fondements de la technologie récente de la reconnaissance de la parole ont été élaborés par Jelinek. F et son équipe à IBM dans les années 70. [48] Les premiers travaux dans les années 80 sont intéressés aux mots, pour des applications de vocabulaire réduit. Au début des années 90, l'élaboration des systèmes de reconnaissance continus et indépendants du locuteur. La technologie s'est développée rapidement, dans nous jours, la communauté de chercheurs travaille sur des systèmes de reconnaissance de parole continue, indépendants du locuteur ou de l'environnement acoustique le tout en temps réel. [42]

Aujourd'hui les systèmes de reconnaissance automatique de la parole sont de plus en plus répandus et utilisés dans des conditions acoustiques très variées, par des locuteurs différents. De ce fait, les systèmes de reconnaissance de la parole RAP, généralement conçus en laboratoire, doivent être robustes afin de garder des performances optimales en situation réelle.

Dans ce chapitre, nous nous intéresserons aux bases de la reconnaissance automatique de la parole (RAP) et nous verrons quels sont les fondements théoriques des différents algorithmes utilisés. La présentation suivra la progression du signal de parole, partant de la production par l'être humain pour finir sous forme d'une chaîne de mots reconnue. Pour ce faire, nous détaillerons la façon dont l'ordinateur traite le signal de parole par le biais de sa paramétrisation. Nous verrons quelles sont les méthodes les plus employées actuellement pour la reconnaissance de la parole.

2. Signal de la parole (le son)

Avant de passer au processus de codage du son dans l'ordinateur ou sa synthèse, il faut d'abord bien comprendre le son lui-même. Nous commencerons donc par la définition du son.

Le son est une vibration de l'air. A l'origine de tout son, il y a un mouvement (par exemple, une corde qui vibre, une membrane de haut-parleur...). Il s'agit de phénomènes oscillatoires crées 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.

Un son est d'abord défini par son volume sonore et sa hauteur tonale. Le volume dépend de la pression acoustique crée par la source sonore (le nombre de particules d'air déplacées). Plus elle est importante 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.

*2.*1 Les caractéristiques du son

Le son est défini par trois paramètres :

- L'amplitude d'un son qui correspond à la variation de pression maximale de l'air engendrée par les oscillations, (volume sonore).
- La dynamique qui permet la mesure de l'écart entre le volume maximal d'un son et le bruit de fond. La dynamique ce mesure en décibels (dB). En fait elle est le rapport entre le niveau maximal, à la limite de la distorsion, et le niveau minimal acceptable, à la limite du niveau de bruit de fond.
- Le timbre qui est un paramètre beaucoup plus subjectif, il s'agit de ce qui différencie deux sons de même hauteur et de même amplitude. C'est une notion qualitative, qui fera dire, par exemple, qu'un son est brillant ou profond…

2.2 Les avantages de la parole

Si on prend comme référence le modèle humain, les avantages de la parole semblent au premier abord déterminant :

- *Naturel:* la parole constitue le mode le plus naturel de communication entre personnes humaines, du fait que son apprentissage s'effectue dès l'enfance, ce qui est loin d'être le cas pour la maîtrise de l'écriture.
- *Rapidité/efficacité:* plusieurs études d'ergonomie montrent que le débit en parole spontanée est de l'ordre de 200 mots/minute à comparer aux 60 mots/minute d'un expert pour la frappe au clavier. L'efficacité de la parole ne provient pas seulement de ce qu'elle permet un débit d'informations plus élevé que d'autres modes de communication, mais également de ce qu'elle peut être aisément utilisée en superposition avec ceux-ci. La parole laisse l'utilisateur libre de ses mouvements, elle est donc particulièrement adaptée aux applications dans lesquelles il s'agit pour l'utilisateur de conduire plusieurs tâches simultanément, ou de contrôler des processus complexes qui monopolisent gestes et/ou vision.[21]

Par ailleurs, d'un point de vue cognitif, la parole associée à des informations visuelles permet d'en améliorer la mémorisation ou de n'en souligner que les points saillants, ce qui est couramment utilisé dans l'enseignement interactif. Cependant, si la parole présente un débit plus rapide pour l'émission d'un message par une machine ou une personne humaine, cet avantage ne se vérifie pas pour la perception humaine, la parole étant un phénomène séquentiel et monodimensionnel, l'écoute d'un message vocal (avec un débit de l'ordre de 200 mots/minute), nécessite un certain effort d'attention (du fait du caractère éphémère) et demande à l'usager plus de temps que la lecture d'une page écran (pour laquelle le débit peut atteindre jusqu'à 700 mots/minute). [56]

Ces résultats d'expérimentation proscrivent les longs messages de synthèse, il est en effet préférable d'afficher à l'écran un message dès qu'il dépasse une certaine longueur ou alors d'en produire une version condensée avec la synthèse vocale. Ceci démontre qu'un mode ne saurait se substituer directement à un autre sans une adaptation préalable de la présentation du message, même si le contenu reste inchangé.

- *Extension du champ d'action***:** la parole permet d'avoir un accès immédiat à une information sans avoir à parcourir toute une arborescence hiérarchique de menus. Il est toujours possible bien sûr d'utiliser des raccourcis clavier ou des langages de commande, mais la prononciation d'un seul mot peut remplacer jusqu'à une dizaine de commandes élémentaires effectuées à

l’aide de touches fonctions ou de souris et représente ainsi un effort mnémotechnique moindre. [14]

La parole permet également d’avoir accès à un objet non visible (sur l’écran, par exemple). À son pouvoir de désigner ou nommer des objets, s’ajoute la possibilité d’établir des échanges de niveau sémantique complexe et de manipuler des notions abstraites, ce qui peut contribuer à modifier, en l’enrichissant, le dialogue avec la machine.

2.3 Complexité du signal de parole

Le signal de la parole n’est pas un signal ordinaire, il est le vecteur d’un phénomène extrêmement complexe, la reconnaissance automatique de la parole pose de nombreux problèmes. D’un point de vue mathématique il est difficile de modéliser le signal de la parole car ses propriétés statistiques varient au cours du temps.

La complexité du signal de parole provient de la combinaison de plusieurs facteurs, principalement la redondance du signal acoustique, la grande variabilité intra-locuteurs et inter-locuteurs, et les effets de la coarticulation en parole continue, qui doivent être pris en compte lors de la conception d'un système de RAP. [11]

2.3.1. Redondance

Le signal acoustique dans le domaine temporel, présente une redondance qui rend indispensable un traitement préalable à toute tentative de reconnaissance. Il existe en effet une grande disproportion entre le débit du signal enregistré et la quantité d'information cherchée pour une tâche de reconnaissance. Un signal échantillonné à 16 kHz sur 16 bits représente un débit de 256 kbit/s, alors qu'une tâche de reconnaissance phonétique recherche typiquement une dizaine de phonèmes à la seconde, soit une compression de près de 10^4 du débit initial. Il existe un grand nombre de paramètres possibles pour la représentation du signal, et le choix de paramètres adaptés à un type de problème dépend des conditions d'enregistrement du signal et surtout de l'usage ultérieur qui doit en être fait, puisque ce qui n'est pas pertinent vis-à-vis du contenu lexical peut le devenir pour l'identification du locuteur. Le choix de cette représentation ne résout pas les difficultés provenant de la continuité de la production et plus généralement de la variabilité de la parole. [63]

2.3.2 Continuité et coarticulation

Tout discours peut être retranscrit par des mots, qui peuvent à leur tour être décrits comme une suite de symboles élémentaires appelés phonèmes par les linguistes. Cela laisse supposer que la parole est un processus séquentiel, au cours duquel des unités indépendantes se succèdent. Malheureusement, les spécialistes de phonétique eux-mêmes ont parfois des difficultés à identifier individuellement ces unités discrètes dans le signal, même si quelques

événements acoustiques particuliers peuvent être détectés. La parole est en réalité un flux continu, et il n'existe pas de pause entre les mots qui pourrait faciliter leur localisation automatique par les systèmes de reconnaissance.

De plus, les contraintes introduites par les mécanismes de production créent des phénomènes de coarticulation. La production d'un son est fortement influencée par les sons qui le précèdent mais aussi qui le suivent en raison de l'anticipation du geste articulatoire. Ces effets s'étendent sur la durée d'une syllabe, voire même au-delà, et sont amplifiés par une élocution rapide. L'identification correcte d'un segment de parole isolé de son contexte est parfois impossible. La prise en compte des phénomènes de coarticulation ne suffit pourtant pas à prédire la réalisation acoustique d'une phrase en raison de la grande variabilité de la parole.[11]

2.3.3. Variabilité

On distingue généralement deux sources de variabilité qui peuvent rendre deux prononciations d'un même énoncé très différentes, la variabilité inter-locuteurs et la variabilité intra-locuteur. La variabilité inter-locuteurs est a priori la plus importante. Les différences physiologiques entre locuteurs, qu'il s'agisse de la longueur du conduit vocal ou du volume des cavités résonantes, modifient la production acoustique; à cet égard, les voix d'hommes, de femmes et d'enfants sont les plus caractéristiques. A cela s'ajoutent les habitudes acquises en fonction du milieu social et géographique, comme les accents régionaux. La variabilité intra-locuteur est plus réduite, mais n'est pas négligeable. L'état physique, par exemple la fatigue ou le rhume, les conditions psychologiques, comme le stress, et même le bruit de fond pendant l'élocution, influent sur la production et sur la prosodie, entraînant des variations complexes du rythme d'élocution, de la mélodie et de l'intensité du discours. Ces phénomènes sont encore mal modélisés et compliquent la réalisation des systèmes de reconnaissance. J.S Liénard défend l'idée d'une prise en compte de toutes les caractéristiques du locuteur pour la représentation de la parole. [63]

3. Tache de reconnaissance de la parole

Le traitement de la parole est situé au croisement du traitement du signal numérique et du traitement du langage (c'est-à-dire du traitement de données symboliques). L'importance particulière du traitement de la parole s'explique par la position privilégiée de la parole comme vecteur d'information dans notre société humaine. L'extraordinaire singularité de cette science, qui la différencie fondamentalement des autres composantes du traitement de l'information, tient sans aucun doute au rôle fascinant que joue le cerveau humain à la fois

dans la production et dans la compréhension de la parole et à l'étendue des fonctions qu'il met, inconsciemment, en œuvre pour y parvenir de façon pratiquement instantanée.

Les techniques modernes de traitement de la parole tendent cependant à produire des systèmes automatiques qui se substituent à l'une ou l'autre de ces fonctions :

3.1 Les analyseurs de parole: les analyseurs de parole cherchent à mettre en évidence les caractéristiques du signal vocal tel qu'il est produit, ou parfois tel qu'il est perçu (on parle alors d'analyseur perceptuel), mais jamais tel qu'il est compris, ce rôle étant réservé aux reconnaisseurs. Les analyseurs sont utilisés soit comme composant de base de systèmes de codage, de reconnaissance ou de synthèse, soit en tant que tels pour des applications spécialisées, comme l'aide au diagnostic médical (pour les pathologies du larynx, par analyse du signal vocal) ou l'étude des langues.

3.2 Les reconnaisseurs de la parole: les reconnaisseurs de la parole ont pour mission de décoder l'information portée par le signal vocal à partir des données fournies par l'analyse.

On distingue des systèmes de reconnaissance de parole *monolocuteur*, *multilocuteur*, ou *indépendant du locuteur*, selon qu'il a été entraîné à reconnaître la voix d'une personne, d'un groupe fini de personnes, ou qu'il est en principe capable de reconnaître n'importe qui. On distingue enfin reconnaisseur de *mots isolés*, reconnaisseur de *mots connectés*, et reconnaisseur de *parole continue*, selon que le locuteur sépare chaque mot par un silence, qu'il prononce de façon continue une suite de mots prédéfinis, ou qu'il prononce n'importe quelle suite de mots de façon continue.

3.3 Les synthétiseurs: les synthétiseurs ont quant à eux la fonction inverse de celle des analyseurs et des reconnaisseurs de parole, ils produisent de la parole artificielle. On distingue fondamentalement deux types de synthétiseurs :

- Les synthétiseurs de parole à partir d'une représentation numérique, inverses des analyseurs, dont la mission est de produire de la parole à partir des caractéristiques numériques d'un signal vocal telles qu'obtenues par analyse.

- Les synthétiseurs de parole à partir d'une représentation symbolique, inverse des reconnaisseurs de parole et capables en principe de prononcer n'importe quelle phrase sans qu'il soit nécessaire de la faire prononcer par un locuteur humain au préalable. Dans cette seconde catégorie, on classe également les synthétiseurs en fonction de leur mode opératoire. Les *synthétiseurs à partir du texte* reçoivent en entrée un texte orthographique et doivent en donner lecture. *Les synthétiseurs à partir de concepts*, appelés à être insérés dans des systèmes de dialogue homme-machine, reçoivent le texte à prononcer et sa structure linguistique, telle que produite par le système de dialogue.

3.4 Les codeurs : leurs rôle est de permettre la transmission ou le stockage de parole avec un débit réduit, ce qui passe tout naturellement par une prise en compte judicieuse des propriétés de production et de perception de la parole.

On comprend aisément que, pour obtenir de bons résultats dans chacune de ces tâches, il faut tenir compte des caractéristiques du signal étudié. Et, vu la complexité de ce signal, due en grande partie au couplage étroit entre production, perception, et compréhension.

4. Les Méthodes de reconnaissance automatique de la parole

On distingue usuellement en reconnaissance de la parole l'approche analytique et l'approche globale. La première approche cherche à traiter la parole continue en décomposant le problème, le plus souvent en procédant à un décodage acoustique phonétique exploité par des modules de niveau linguistique. La seconde consiste à identifier globalement un mot ou une phrase en le comparant avec des références enregistrées. [11]

4.1. Approche analytique

L'approche analytique cherche à résoudre le problème de la parole continue en isolant des unités acoustiques courtes (de petites tailles) comme les phonèmes, les diphonèmes ou les syllabes. Un exemple classique de cette approche est l'analyse par traits des indices acoustiques sont calculés à partir du signal de parole, ils permettent de faire des hypothèses locales sur certains traits phonétiques, comme le voisement, la nasalisation, le lieu d'articulation ou le degré d'ouverture du conduit vocal. En fonction de ces traits, le signal acoustique est segmenté et une identification phonétique des segments est réalisée.

Cette approche est mieux adaptée aux systèmes à grand vocabulaire et pour la parole continue. Le décodage acoustico-phonétique utilisé par cette approche est une tâche difficile, peu de systèmes s'approchent du taux de reconnaissance souhaitable. [11]

4.2 Approche globale

Les méthodes globales identifient un mot ou une phrase en le considérant comme des entités élémentaires et en le comparant avec des références enregistrées. Elle consiste à faire prononcer un ou plusieurs exemples de chacun des mots susceptibles d'être reconnus, et à les enregistrer sous forme de vecteurs acoustiques.

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

une grande capacité de reconnaissance et une indépendance vis-à-vis des particularités de la langue à reconnaître, mais elle utilisé vocabulaires très limités. [11]

5. Principaux modules d'un système de RAP

Les systèmes classiques de RAP sont composés essentiellement de cinq modules :

- la paramétrisation du signal, qui doit permettre de ne garder que les informations pertinentes de ce dernier.
- Les modèles acoustiques, qui doivent représenter au mieux les unités acoustiques choisies (phonèmes, diphones, mots. . .).
- Les modèles linguistiques, qui doivent être une représentation la plus vraisemblable possible du langage.
- Le dictionnaire, qui doit contenir l'ensemble des mots que l'on souhaite pouvoir reconnaître (dans certains cas le dictionnaire peut être spécifique à une application).
- le système de reconnaissance lui même.

Ces différentes composantes d'un système de RAP, sont relativement indépendantes les unes des autres. Bien qu'ils sont toutes nécessaires pour la reconnaissance de parole. [23]

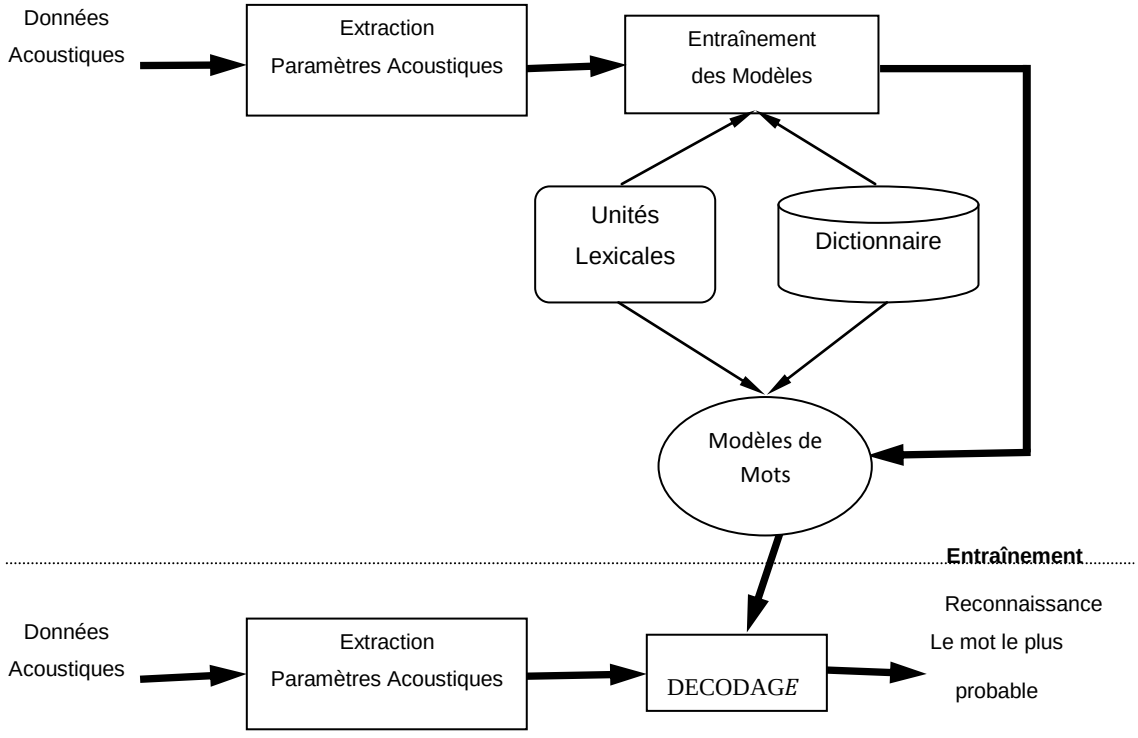


Figure1.1 Schéma général d'un système de reconnaissance de la parole. [63]

5.1 Paramétrisation du signal

Il est possible de calculer des coefficients représentatifs du signal traité, pour résoudre les problèmes liés à la complexité de la parole, ces coefficients sont calculés à intervalles temporels réguliers, le signal de la parole est transformé en une série de vecteurs de coefficients.

Un tel système prend un signal en entrée et retourne un vecteur de paramètres (coefficients) (appelé indifféremment vecteur acoustique ou encore vecteur d'observations). Les vecteurs de paramètres doivent être pertinents (précis, de taille restreinte et sans redondance), discriminants (pour faciliter la reconnaissance) et robustes (aux différents bruits et/ou locuteurs). Les coefficients doivent représenter au mieux le signal qu'ils sont censées modéliser, et extraire le maximum d'information utiles pour la reconnaissance.

Il existe un certain nombre d'approches pour la paramétrisation du signal :

- paramétrisation basée sur un modèle de production de la parole.
- paramétrisation basée sur une analyse dans le domaine cepstral.
- paramétrisation basée sur la représentation par formes d'ondes.
- paramétrisation basée sur la représentation en ondelettes de Morlet.

Les méthodes les plus courantes pour le traitement du signal de la parole sont les analyses spectrales réalisées soit par transformée de Fourier à court terme, soit par prédiction linéaire ou soit par évaluation des coefficients cepstraux. [26]

5.1.1 Analyse temps fréquence du signal de la parole

Le signal acoustique de la parole est variable dans le temps. Aussi, les descriptions temps fréquences sont des formes de représentation couramment utilisées en analyse de la parole. Les termes description, temps et fréquence doivent être pris dans un sens suffisamment large pour inclure diverses formes de représentation et plusieurs notions pour le temps ou la fréquence.

La notion de fréquence évoque la répétition dans le temps d'un même motif (par exemple la sinusoïde). On peut distinguer La fréquence μ au sens de Fourier. Elle permet de représenter un signal d'énergie finie par une somme d'exponentielles complexes.

$$x(\mu) = \int_{-\infty}^{+\infty} x(t).e^{-2i\Pi\mu t} dt \qquad 1.1$$

Il existe d'autres répétitions avec d'autres formes que des sinusoïdes, ou une invariance de formes à des échelles de temps et de fréquences différentes.

La notion de temps peut donner lieu à deux interprétations permettant de distinguer: les méthodes adaptatives pour lesquelles le temps est un ensemble de dates, avec hypothèse de stationnarité locale. Les méthodes évolutives où le temps est une variable de la représentation, sans notion de stationnarité locale.

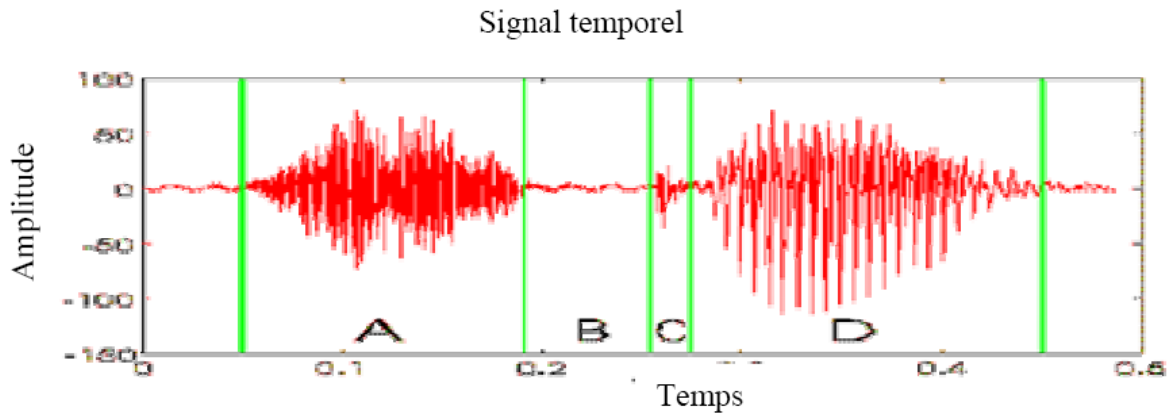


Figure 1.2 Représentation temporelle du signal acoustique de la parole

5.1.2 Le spectrogramme

Le spectrogramme présente l'énergie a court terme pour différentes bandes des fréquences en fonction du temps (très proche par le concept de la transformée de Fourier à court terme). Mais le problème était que la largeur de bande devait être choisie. Une bande étroite permet une bonne résolution harmonique mais génère des imprécisions importantes pour la résolution temporelle. A l'inverse, une bande large permet une bonne résolution temporelle mais une faible résolution harmonique. [13]

Avec l'utilisation des calculateurs, et donc des méthodes numériques, il faut échantillonner et numériser le signal. La fréquence d'échantillonnage est généralement comprise entre 8 et 16 kHz tandis que la quantification se fait sur 8 à 16 bits.

L'échantillonnage transforme le signal à temps continu x(t) en signal à temps discret x (nTe) défini aux instants d'échantillonnage, multiples entiers de la période d'échantillonnage Te, celle-ci est elle-même l'inverse de la fréquence d'échantillonnage fe. Ce qui concerne le signal vocal, le choix de *fe* résulte d'un compromis. Son spectre peut s'étendre jusque 12 kHz.

Il faut donc en principe choisir une fréquence *fe* égale à 24 kHz au moins pour satisfaire raisonnablement au théorème de Shannon. [13]

Avant l'échantillonnage, un filtre passe-bas de fréquence de coupure égale à la moitié de la fréquence d'échantillonnage est inséré pour éviter l'effet dénommé *repliement* ou *aliasing* postulé par le théorème de Shannon, Ce filtre est donc appelé filtre (*anti-repliement* ou *anti-aliasing*). Quelques fois, le filtre n'est pas choisi complètement plat. On peut le faire d'une autre manière avec un filtre numérique du premier ordre après l'échantillonnage de la façon suivante :

$H(z)= 1 - \alpha z^{-1}$ Avec α proche de 1(α= 0,97). 1.2

Pour obtenir un spectrographe numérique, on effectue sur le signal une TFR (transformée de fourier rapide) à fenêtre glissante. C'est à dire qu'on analyse une portion limitée du signal, prélevée à l'aide d'une fenêtre de pondération. Pour ne pas perdre d'information et assurer un meilleur suivi des non-stationnarités, les fenêtres se recouvrent. Elles ont généralement une longueur de 256 ou 512 points et le recouvrement est de 50%, soit 128 ou 256 points.

$$W(n) = 0.54 + 0.46 \cos\left(2\pi \frac{n}{N-1}\right) \quad 1.3$$

La fenêtre de hamming est couramment utilisée en reconnaissance de la parole mais il existe plusieurs d'autres types de fenêtre telle que rectangulaire, hanning, blackman…..la fenêtre de hamming est un cas particulier de la fenêtre de hanning.
Ce traitement implique une hypothèse importante du fait des limitations postérieures qu'elle occasionne le signal vocal est supposé stationnaire sur une courte période. [63]

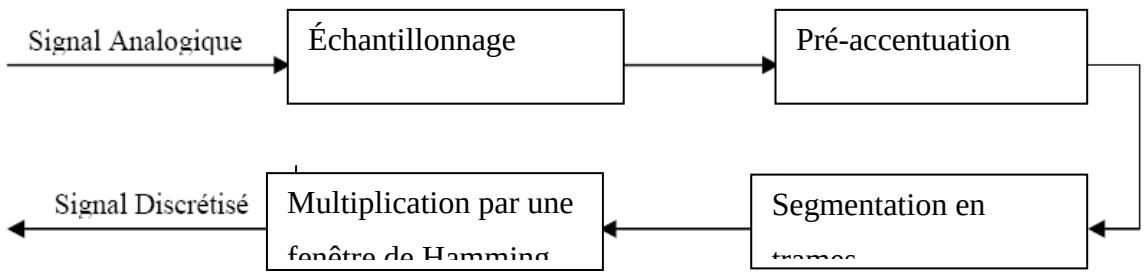


Figure1.3 Prétraitement du signal vocal. [63]

5.1.3 Transformée de Fourier

Le signal de la parole, échantillonné et préaccentué dans les hautes fréquences, est prélevé par une fenêtre temporelle glissante de type Hamming. Puisque c'est la largeur de cette dernière qui détermine la résolution spectrale de l'analyse, il apparaît donc un conflit entre la résolution temporelle et la résolution fréquentielle. On peut conclure qu'une analyse en bande étroite, d'une résolution fréquentielle de 50Hz environ, permet une bonne représentation de la structure harmonique du signal. Mais cette dernière se fait au détriment de la résolution temporelle qui se traduit par une intégration des évolutions temporelles rapides. Une transformée de Fourier est ensuite calculée pour chaque valeur de décalage de la fenêtre.

$$X(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(K) e^{-jk2\pi(\frac{n}{N})} \qquad 1.4$$

Avec K Є [0, N-1] et N le nombre de points prélevés. En prenant le carré du module de la TFCT. On obtient alors un spectrogramme représentant la distribution énergétique dans le plan temps fréquence, puisque pour chaque instant n, on dispose alors de l'énergie associée aux fréquences k=0, ..., N-1. Il suffit alors d'appliquer un filtrage suivant une échelle de Mel ou de Bark, fréquemment utilisée en reconnaissance vocale, pour obtenir un superbe sonagraphe numérique. [44] La TFCT présente l'avantage de vecteurs de paramétrisation constitués d'une vingtaine de composantes obtenus avec un faible volume de calcul donnant une image proche de celle du sonagraphe. [22]

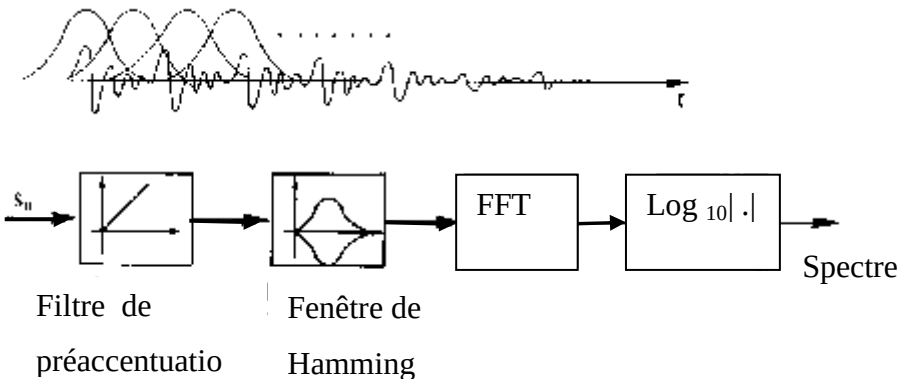


Figure 1.4 Transformée de Fourier. [19]

5.2 Paramétrisation basée sur un modèle de production de la parole LPC (*Linear Predictive Coefficients*). [75]

La théorie du codage par prédiction linéaire appliquée à la parole figure parmi les techniques les plus utilisées en traitement automatique de la parole. La prédiction linéaire est une technique qui s'applique directement après l'échantillonnage et la quantification du signal de la parole. Le modèle d'analyse par prédiction linéaire LPC est basé sur les connaissances expertes en production de la parole. Le conduit vocal est modélisé comme un filtre autorégressif (AR). Ceci permet d'approximer l'échantillon de l'instant n (s(n)) par une combinaison linéaire des p échantillons précédents (P étant l'ordre du modèle).

$$\tilde{s}(n)=\sum_{i=1}^{p} a.i * s(n-i) \qquad 1.5$$

L'erreur de prédiction du modèle peut être estimée par :

$$e(n)=s(n)-\tilde{s}(n) \qquad 1.6$$

e(n) représente un bruit blanc dû à toutes les sources d'erreurs possibles (précision des termes, arrondis de calcul, ...).

On peut donc estimer l'erreur quadratique moyenne par :

$$E_n=\sum_m e(m)^2=\sum_m [s(m)-\sum_{i=1}^{p} ai.s(m-i)]^2 \qquad 1.7$$

Minimiser cette erreur quadratique revient à annuler les dérivées partielles de E_n par rapport aux α_i. Pour cela, plusieurs approches sont présentées dans la littérature (méthode de covariance, méthode d'auto-corrélation). [76]

L'analyse par prédiction linéaire permet de passer d'un spectre échantillonné, donc bruité à une représentation spectrale continue et lissée. La détection des formants en est alors plus aisée. Cette méthode présente l'inconvénient du choix du nombre de coefficients à prendre en fonction de la fidélité par rapport au signal analysé.

LPCC (Linear Predictive Cepstral Coefficients)

La méthode présentée par Miet G [60] permet de calculer les coefficients LPCC directement depuis les coefficients LPC. Selon la fonction (1.8).

$$LPCC_i=-LPC_i+\sum_{k=1}^{i-1}\left(1-\frac{k}{i}\right)LPC_k\, LPCC_{i-1} \qquad 1.8$$

Cette approche a pour but de modéliser davantage l'enveloppe du signal. [23]

5.3 Paramétrisation basée sur une analyse dans le domaine cepstral

5.3.1 Le cepstre

Le cepstre est basé sur une connaissance du mécanisme de production de la parole. On part de l'hypothèse que la suite s_n constituant le signal vocal est le résultat de la convolution du signal de la source par le filtre correspondant au conduit:

$$s_n = g_n * h_n \quad 1.9$$

Avec s_n le signal temporel, g_n le signal excitateur, h_n la contribution du conduit.

Le but du cepstre est de séparer ces deux contributions par déconvolution. Il est fait l'hypothèse que g_n est soit une séquence d'impulsions (périodiques, de période T_0, pour les sons voisés), soit un bruit blanc, conformément au modèle de production.

Dans le domaine spectral, la convolution devient un produit qui est transformé en addition par le logarithme:

$$\log S_k = \log G_k + \log H_k \quad 1.10$$

où $\{S_k\}$, $\{G_k\}$ et $\{H_k\}$ sont les spectres respectifs de $\{s_n\}$, $\{g_n\}$ et $\{h_n\}$.

Par transformation inverse, on obtient le cepstre. L'expression du cepstre est donc:

$$ç(n) = FFT^{-1}(Log(FFT(s(n))). \quad 1.11$$

L'espace de représentation du cepstre (espace quéfrentiel) est homogène au temps et il est possible, par un filtrage temporel (liftrage), de séparer dans le signal, la contribution de la source de celle du conduit. Les premiers coefficients cepstraux contiennent l'information relative au conduit. Cette contribution devient négligeable à partir d'un échantillon n_0. Les pics périodiques visibles au-delà de n_0, reflètent les impulsions de la source.

A partir du cepstre, il est possible de définir la fréquence fondamentale de la source g_n en détectant les pics périodiques au-delà de n_0. Le spectre du cepstre pour les indices inférieurs à n_0 permettra d'obtenir un spectre lissé, débarrassé des lobes dus à la contribution de la source.[11]

5.3.2 Analyse MFCC

L'analyse MFCC consiste en l'évaluation de coefficients cepstraux à partir d'une répartition fréquentielle selon l'échelle des Mels. Mais il faut noter, dans l'analyse MFCC décrite ci-dessous, que les coefficients cepstraux obtenus ne correspondent pas exactement à la définition du cepstre défini précédemment. L'utilisation d'une FFT et d'une FFT inverse permettent de calculer les coefficients MFCC, LFCC et PLP. [51]

MFCC (Mel Frequency Cepstral Coefficient

Afin de rapprocher l'analyse en banc de filtres de la perception humaine, les filtres ne sont généralement pas répartis de manière linéaire mais en fonction d'une échelle Mel. La correspondance entre une fréquence en Hz et en Mel se calcule de la manière suivante :

$$F_{mel}=2595*\log(1+\frac{F}{700}Hz) \quad 1.12$$

Intuitivement, cela revient à utiliser une échelle linéaire en basse fréquence, puis logarithmique en haute fréquence.

La chaîne complète de calculs des coefficients MFCC est définie par la figure (1.6). Le calcul se déroule de la manière suivante :

- La FFT est calculée sur le fragment (frame).
- Cette dernière est filtrée par un banc de filtres triangulaires répartis le long de l'échelle de Mel.

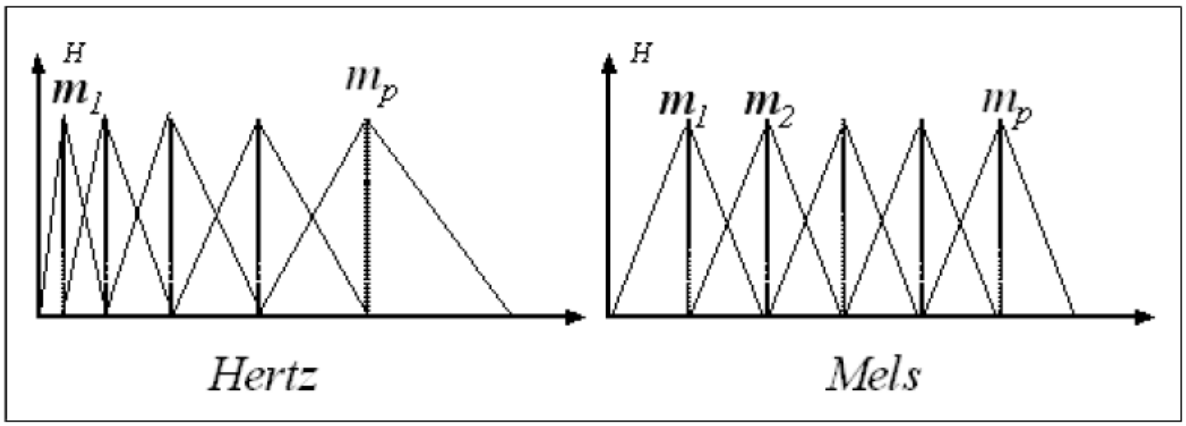


Figure1.5 Répartition des filtres triangulaires sur les échelles fréquentielle et Mel. [32]

Sur cette illustration, m_p correspond au nombre de filtres que l'on souhaite.

- Le logarithme module de l'énergie de sortie du banc de filtres est calculé.
- Une Transformée en cosinus discrète inverse, (équivalente à la FFT inverse pour un signal réel) est appliquée.

$$X(n) = \frac{1}{N}\sum_{k=0}^{N-1} x(k)e^{-jk2\pi(\frac{n}{N})} \quad 1.13$$

- Seuls les premiers coefficients sont conservés.

Généralement, seuls les 12 premiers coefficients cepstraux sont conservés et une vingtaine de filtres sont utilisés pour l'analyse en banc de filtres.

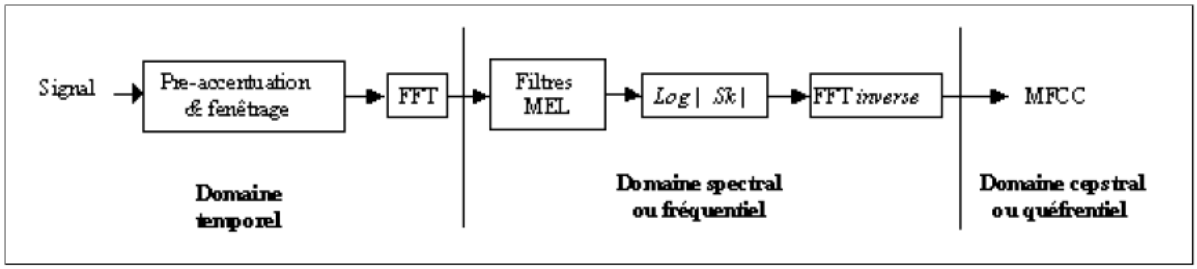


Figure1.6 schéma de calcul des MFCC. [32]

Les avantages de cette méthode sont les suivants :

- Le nombre de données par vecteur est réduit. En pratique, pour 24 filtres (Nf=24), il a été montré que 12 coefficients cepstraux (Nc=12) suffisent pour représenter l'information.
- Les valeurs des vecteurs sont relativement décorellées entre elles, ce qui est idéal pour la reconnaissance vocale.

LFCC (linéaire fréquence cepstral coefficient)

Il s'agit d'une variante des MFCC. La différence vient de l'utilisation d'un banc de filtres linéaire contrairement à l'échelle Mel des MFCC.

Energie : Généralement, l'énergie du signal est utilisée en complément des coefficients issus d'une paramétrisation basée sur une analyse dans le domaine cepstral. L'énergie correspond à la puissance du signal.

$$En=\sum_{n=0}^{N-1} s_n^2 \qquad 1.14$$

Le calcul de l'énergie se fait généralement sur des fenêtres glissantes de 25ms avec un décalage de 10ms (soit une valeur toutes les 10ms de signal).

5.3.3 Autres paramétrisation

Taux de passage par zéro

Le ZCR (Zero Crossing Rate) est un bon complément de l''energie. Un taux de passage par zéro faible et une énergie forte sont un bon indice d'un son voisé alors qu'un taux de passage par zéro élevé et une énergie plus faible caractérisent plutôt une zone non voisée.

$$Zcr = 0{,}5.\sum_{n=0}^{n-1} \left| sign(x_n) - sign(x_{n-1}) \right| \qquad 1.15$$

Une évolution du ZCR est proposée en 1994, il propose une bande d'amplitude autour de zéro pour limiter un certain nombre de phénomènes parasites qui provoquent de faibles oscillations aux alentours de 0. [23]

6. Décodage acoustique phonétique

Le décodage acoustique phonétique consiste à décrire le signal de parole en termes d'unités linguistiques discrètes. On trouve plusieurs unités de décodage seuls les plus

utilisées sont les phonèmes, les syllabes, les mots… Un phonème est un élément sonore d'un langage donné, déterminé par les rapports qu'il entretient avec les autres sons de ce langage. Cette notion est assez importante en reconnaissance vocale.

Le décodage a pour but de segmenter le signal en segments élémentaires et d'étiqueter ces segments. Le principal problème est de choisir les unités sur lesquelles portera le décodage. Si des unités longues telles que les syllabes ou les mots sont choisies, la reconnaissance en elle-même sera facilitée mais leur identification est difficile. Si des unités courtes sont choisies, comme les phones (sons élémentaires), la localisation sera plus facile mais leur exploitation nécessitera de les assembler en unités plus larges. Les phonèmes constituent un bon compromis, leur nombre est limité, ils sont donc souvent utilisés. Mais le choix dépend également du type de reconnaissance effectuée: mots isolés ou parole continue. Une fois la segmentation effectuée, l'identification des différents segments se fait en fonction de contraintes phonétiques, linguistiques… Il faut que le système ait intégré un certain nombre de connaissances: données articulatoires, données phonétiques, prosodiques, syntaxiques, sémantiques …

7. Principe de décodage

Plusieurs méthodes en reconnaissance automatique de la parole à partir des années 1950 sont élaborées parmi ces méthodes on trouve une méthode appelée Dynamic Time Warping (DTW) est apparue. Elle repose sur les travaux de Bellman. R [15] et qui reste aujourd'hui une approche importante en reconnaissance de mots isolés. Une deuxième méthode est développée dans les années 70. Elle s'appuie sur l'utilisation des modèles de Markov cachés (Hidden Markov Models - HMM). Elle a permis de nombreuses avancées dans les domaines de la reconnaissance de la parole continue et de la reconnaissance multi-locuteurs, domaines dans les quels la DTW était peu probante. A la fin des années 80, les réseaux de neurones sont venus offrir une nouvelle voie pour le traitement automatique de la parole [17]. Plusieurs types de réseaux de neurones coexistent mais, pour la reconnaissance de mots isolés, les approches les plus classiques sont le perceptron multi-couches et les TDNN (Time Delay Neural Network). [18]

7.1 Systèmes experts

Les systèmes experts sont des systèmes cherchant à reproduire l'analyse faite par des experts humains (notamment des phonéticiens dans le cadre de la reconnaissance de la parole). Ces experts procèdent généralement en deux phases. Une analyse visuelle du

spectrogramme suivie d'un raisonnement contextuel avec les indices notés lors de la première phase.

De tels systèmes sont généralement composés de deux entités distinctes :

- la base de connaissance qui contenant les règles et les faits répertoriés par un expert humain.

- le moteur d'inférence: réalisant les déductions logiques à partir de la base de connaissances.

L'objectif principal de ces systèmes est de réaliser des raisonnements logiques comparables à ceux que feraient des experts humains de ce domaine. Des exemples de règles et de faits sont présentés dans [20]. L'analyse visuelle d'un certain nombre de segments a permis d'extraire des faits, Ils présentent aussi des exemples de règle.

Des systèmes experts pour la reconnaissance du français peuvent être trouvés dans [12,46,73], ou encore un système en anglais dans [79]. Ces systèmes ont progressivement été abandonnés avec la généralisation des systèmes Markoviens.

7.2 Reconnaissance par alignement temporel dynamique DTW (Dynamic Time Warping)

Un locuteur ne peut pas éviter les variations du rythme de prononciation ou de la vitesse d'élocution même s'il est entraîné, à répète plusieurs fois une phrase ou un mot. Ces variations entraînent des transformations non linéaires dans le temps du signal acoustique. La non-linéarité vient du fait que les transformations affectent plus les parties stables du signal que les phases de transitions.

La méthode (DTW: *Dynamic Time Warping*) introduit en reconnaissance de la parole en 1968. Elle se base sur une technique de comparaison dynamique, ou alignement temporel dynamique. Cette méthode utilisée pour affranchir les transformations et de réaliser une normalisation temporelle en même temps que la comparaison des deux mots. Cette méthode utilise la matrice des distances locales entre ces vecteurs et permet de calculer récursivement les distances globales entre les zones du signal de test et du signal de référence en partant des instants de début de chacun des deux signaux pour aboutir aux instants de fin de ces signaux. La formule de récurrence nous permet de connaître à chaque instant t le chemin local optimal et donc de retrouver le chemin optimal à partir de la fin en passant par les chemins locaux optimaux. L'efficacité de cette méthode dépend d'un certain nombre de choix et de paramètres: choix des distances, choix des chemins locaux admissibles et de la zone de recherche du chemin optimal. [34, 59]

7.2 .1 Distance euclidienne

L'algorithme de comparaison dynamique, basé sur le principe de programmation dynamique, il commence par le calculer de la *distance euclidienne d(I,J)* entre deux signaux en tenant compte des variations de durée dans la prononciation des mêmes mots. La comparaison du signal d'un mot du dictionnaire (référence) et du signal pour un mot de test revient alors à rechercher un *chemin optimal W* dans une grille de *I*J* points

Soit « i », i ∈ [1, I], un vecteur issu de la paramétrisation et appartenant au mot test, Soit « j », j ∈ [1, J], un vecteur appartenant au mot du dictionnaire de référence, et d(i,j) la distance euclidienne :

Si $i = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}$ et $j = \begin{bmatrix} y_1 \\ y_2 \\ \dots \\ y_n \end{bmatrix}$ alors $d(i,j) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$ 1.16

La distance d (i, j) représente la distance entre le spectre de la référence et le spectre du test aux instants i et j.

7.2 .2 La distance cumulée

La distance cumulée en un point est la somme des distances locales depuis l'origine en suivant le chemin optimal, c'est à dire de moindre coût. Pour préserver une certaine cohérence dans le calcul du chemin optimal, les transitions autorisées entre les points du graphe de coïncidence sont limitées à quelques uns des points les plus proches. Ces limitations peuvent être définies par les contraintes locales suivantes que nous avons choisies:

- La distance cumulée **g (i,j)** est calculée en respectant les propriétés de monotonie et d'évolution lente du signal étudié. C'est à dire, les seuls chemins valides arrivant au point (i,j) viennent des points (i-1,j), (i-1,j-1) ou (i, j-1).

$$g(i,j) = \min \begin{bmatrix} g(i-1,j) + d(i,j) \\ g(i-1,j-1) + 2.d(i,j) \\ g(i,j-1) + d(i,j) \end{bmatrix} \quad 1.17$$

7.2 .3 La distance normalisée

La distance normalisée G entre les deux prononciations du mot et est défini par :

$$G = \frac{g(i,j)}{i+j} \quad 1.18$$

La méthode de la comparaison dynamique consiste à choisir, parmi tous les chemins physiquement possibles, la référence pour laquelle la distance totale " G " est la plus faible et

qui représente le chemin le plus court. L'étiquette du mot reconnu peut alors être fournie comme un résultat. Si la distance est trop élevée, en fonction d'un seuil prédéfini, la décision de non reconnaissance du mot est alors prise, cela permet de rejeter les mots qui n'appartiennent pas au dictionnaire de référence. La ressemblance idéale se traduit donc par une diagonale.

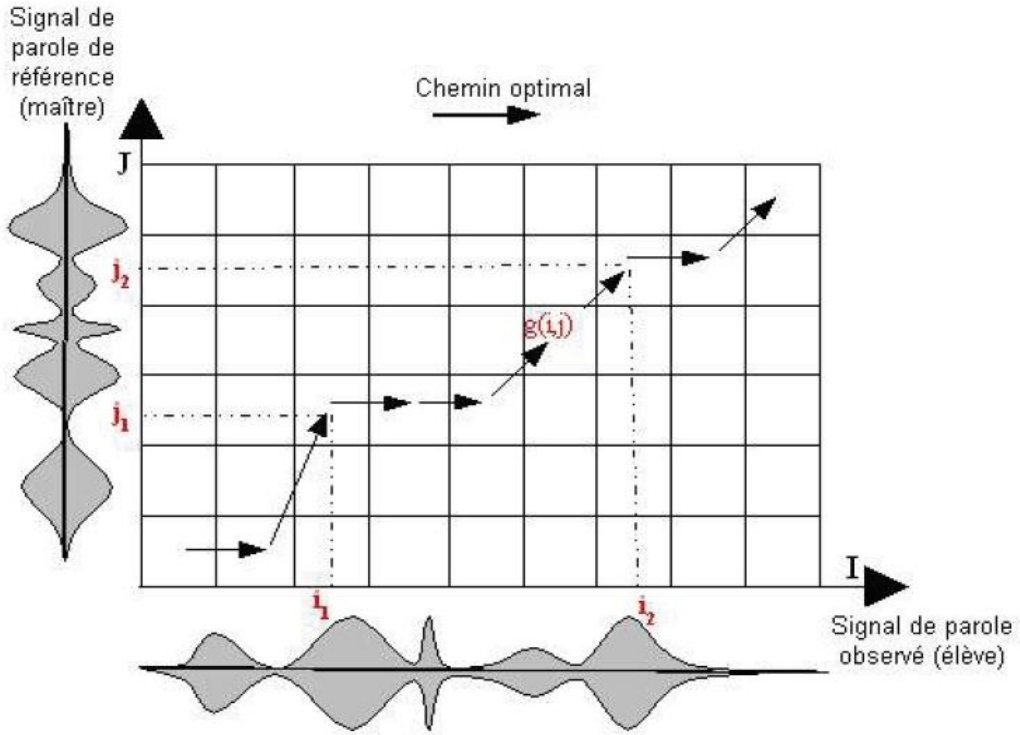


Figure1.7 distances cumulées et chemin optimal

L'algorithme DTW est un très bon outil capable de comparer deux spectres ayant des durées différentes, un débit, une intensité de la voix différente et cela de façon optimale en recherchant le meilleur chemin pour passer d'un spectre à l'autre.

7.3 La reconnaissance statistique de la parole

Les systèmes de RAP sont généralement reposent sur une modélisation statistique du processus de génération de la parole. Mettre des mots sur un signal équivaut à trouver la séquence de mots W=w1, w2 …..wn, la plus probable étant donné le signal observé. Chaque mot de cette chaine appartient à l'ensemble fixe et fini qui constitue le vocabulaire V. Et en fait de signal c'est une suite de vecteurs d'observations acoustiques X, ce ci est traduit par l'équation (1.19):

$$W^* = \arg\max_w P(W / X)$$

1.19

Où P(W\X) est la probabilité de la séquence de mots W étant donné la suite d'observations acoustiques X. l'application de la formule de beyes (1.19) permet d'inverser les dépendances et la modélisation (source \canal) permet de décomposer le processus de production de la parole comme le montre la figure1.8. Le message W est généré par un modèle linguistique P(W) ce message est transformé par le modèle de prononciation P(H\W) en une séquence de phones H.

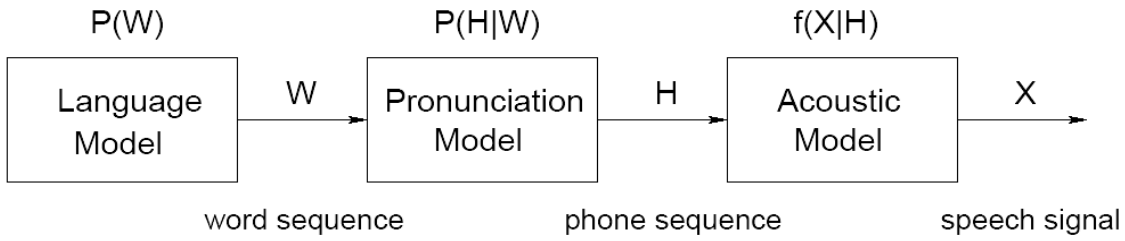


Figure1.8 modèle source –canal du processus de production de la parole

En fin le canal acoustique F(X\H) encode H dans le signal X. La reconnaissance de la parole consiste alors à trouver la séquence de mots W^* qui maximise la probabilité a posteriori de W décomposée de la manière suivante :

$$W^* = \arg\max_w \sum_H P(W)P(H/W)f(X/H)P(W/X) \qquad 1.20$$

Cette équation met en évidence les différentes composantes et l'architecture d'un système de RAP. Un décodeur cherche la séquence de mots la plus probable en utilisant les modèles acoustiques pour estimer F(X\H) un modèle de prononciation pour P(H\W) et un modèle de langage pour la probabilité a priori P(W). Donc un mot pour qu'il existe du point de vue du décodeur il est indispensable qu'il fasse partie du vocabulaire que sa transcription phonétique soit disponible et que le modèle de langage soit apte à lui attribuer une probabilité. [3]

7.3.1 Modèles de Markov cachés (HMM). [70]

Les modèles acoustiques utilisés pour la reconnaissance de la parole sont depuis des années principalement basés sur les HMMs (Hidden Markov Models ou Modèles de Markovs Cachés). Les modèles de markov cachés (HMM Hidden Markov Model) ont été décrits pour la première fois 1970, mais ce n'est qu'en 1975 qu'ils ont été proposés dans le cadre de la reconnaissance automatique de la parole et se sont imposés depuis comme modèles de référence dans ce domaine nous proposons dans les paragraphes suivantes la définition d'un HMM, et de décrire sa mise en œuvre dans le cadre de la reconnaissance automatique de la parole.

7.3.2 Définition

Un HMM est un cas particulier des modèles stochastiques graphiques, ils sont des automates probabilistes à états finis qui permettent de calculer la probabilité d'émettre une séquence d'observations. Pour un système de RAP, les émissions sont donc les vecteurs de caractéristiques du signal de parole composés généralement de coefficients MFCC. Il est caractérisé par un quadruplet (S,Π,A,B) :

- $S=\{S_O,S_1,\ldots,S_i,\ldots,S_k\}$ est l'ensemble des états de l'automate .
- $\Pi=\{\Pi_0, \ldots, \Pi_i,\ldots, \Pi_k\}$, avec Π_i étant la probabilité que S_i soit l'état initial.
- A est l'ensemble des probabilités de transition d'un état vers un autre, A est caractérisé par un matrice k*k d'éléments a_{ij} avec i et j Є [0,K] et K est le nombre d'états tout élément a_{ij} de cette matrice est la probabilité d'atteindre l'état S_j au temps t sachant que nous étions dans l'état S_i au temps t-1 .
- B est un ensemble de lois de probabilité $b_i(0)$ donnant la probabilité $p(o\backslash S_i)$ que l'état S_i ait généré l'observation o, cette probabilité est la vraisemblance de l'observation au regard de S_i . [32]

Un HMM doit respecter les contraintes suivantes :

1. La somme des probabilités des états initiaux doit être égale à 1.

$$\sum_i \Pi_i = 1$$

2. La somme des probabilités des transitions sortant d'un état doit être égale à 1. $\forall i \sum_j a_{ij} = 1$

3. La somme des probabilités des émissions d'un état doit être égale à 1 :

- Dans le cas d'observations discrètes $\forall i \sum_o bi(o) = 1$

- Dans le cas d'observation continues $\forall i \quad b \int_o bi\,(o)\ do = 1$

Un HMM représente un objet par deux suites de variables aléatoires :

- Une suite observable qui correspond à la suite de observations $o_1,o_2\ldots\ldots,o_t$ où les o_i sont des vecteurs d'observations du signal à reconnaitre.
- Une suite cachée qui correspond à une suite d'états $q_1,q_2,\ldots\ldots,q_t$,où les q_i prisent leurs valeurs parmi l'ensemble des N états du modèle $\{s_1,s_2,\ldots,s_n\}$. la suite observable est définie comme une réalisation particulière de la suite cachée. L'objectif est de déterminer la meilleure séquence d'états $Q^*= (q_1^*, q_2^*,\ldots..,q_t^*)$ à partir de la séquence

d'observation O=(o_1,o_2…….,o_t). Le meilleur chemin Q^* est celui qui maximise la probabilité *a posteriori* P(Q\O) (critère de maximum a posteriori), est donné par l'équation suivant la regle de byése $P(X\backslash C_i).P(C_i) > P(X\backslash C_j).P(C_j)\ \forall\ j \neq i$ 1.21

En effet en dérivant cette probabilite a posteiori par la règele de bayes il vient

$$Q^* = \arg\max_Q P(Q\backslash O) = arg\max_Q \frac{P(O\backslash Q)P(Q)}{P(O)} \quad 1.22$$

P(O) étant constant pour tout Q : $Q^*=\arg\max_Q P(O\backslash Q)P(Q)$ 1.23

Un HMM présente plusieurs avantages. Il s'inscrit dans un formalisme mathématique bien établi, il bénéficie de méthodes d'apprentissage automatique des ses paramètres et il particulièrement bien adapté à la modélisation de processus à évolution temporelle. [70]

7.3.3 Mise en œuvre

La mise en œuvre d'un système de reconnaissance de la parole à partir de HMM nécessite de formuler quelques hypothèses simplificatrices dans le but d'adapter le cadre théorique des HMM à la RAP mais aussi d'en simplifier le formalisme mathématique et ainsi proposer des algorithmes d'apprentissage et de classification optimaux sous ces hypothèses. Une fois ces hypothèses posées trois points importants sont à considérer pour la reconnaissance de la parole à partir de HMM. Si on a une suite d'observation O et Q une séquence d'états alignée avec la suite d'observations, au temps t le HMM est dans l'état q_t engendrant l'observation O_t. Les hypothèses simplificatrices sont :

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

$$P(o_t\backslash q_t, q_{t-1}, .., q_1, o_{t-1}, o_{t-2}, .., o_1) = P(o_t\backslash q_t, q_{t-1}, ., q_1) \quad 1.24$$

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

$$P(o_t\backslash q_t, q_{t-1}, \ldots, q_1) = P(o_t\backslash q_t) \quad 1.25$$

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

$$P(q_t\backslash q_{t-1}, q_{t-2}, \ldots, q_1) = P(q_t\backslash q_{t-1}) \quad 1.26$$

Un modèle de Markov d'ordre 1 est un modèle qui respecte l'hypothèse 3. Un modèle d'ordre N est un modèle pour lequel la probabilité de se trouver dans un état q_t est conditionnée par la suite d'états q_{t-1},q_{t-2},……,q_{t-n}.

7.3.4 La topologie du modèle

Pour déterminer la topologie du modèle il faut que on répondre ou questions suivants : Comment définir le nombre d'états du modèle ? Quelles transitions entre les états sont permises ?quelles lois de probabilités utiliser pour modéliser la distribution des paramètres de chaque état ?

La topologie d'un modèle regroupe le choix du nombre d'états ainsi que la définition d'une matrice de transition initiale. Ce choix est fortement dépendant de l'unité de modélisation choisie. L'entité de modélisation pour les HMM la plus utilisé est le phonème mais il existe d'autre unité plus grande comme la syllabe ou le mot. La réalisation des systèmes pour chaque mot d'un langage est couteuse est n'est pas envisageable pour des raisons de temps et d'espace de calcule, la taille de la base d'apprentissage doit contenir suffisamment d'exemple pour chaque mot pour la fiabilité du système.

Les états des HMM peuvent être interprétés comme des zones stationnaires du signal. Le début et fin des réalisations phonétiques présentant des caractéristiques assez différentes de leur centre, plusieurs états sont nécessaires. La topologie gauche-droite à 3 états s'est révélée bien adaptée à la RAP continue. Les techniques d'inférence automatique peuvent être mises en œuvre pour améliorer l'adéquation entre les données et la topologie des modèles. [33]

7.3 .4.1 Modèle gauche-droit

La topologie gauche droite est adapte dans la grande majorité des cas pour le but de restituer l'évolution temporelle du signal de la parole ce qui impliquer qu'aucun retour on arrière n'est pas possible. Donc Le modèle gauche-droit traduit la causalité du processus de production de la parole, il n'existe pas de cycle dans le graphe orienté engendré par les transitions entre états. Les modèles gauche-droit particuliers, autorisant le bouclage à l'état courant, le passage à l'état suivant ou le saut d'un état sont le plus couramment utilisés pour représenter les unités phonétiques. Trois états sont généralement utilisés. [33]

Des structures plus complexes ont été proposées. La topologie à 7 états de l'équipe d'IBM [31], qui est utilisée dans le système SPHINX du CMU. [57]

7.3.4.2 Apprentissage

L'un des principaux problèmes de l'utilisation des HMMs réside dans la phase d'apprentissage, qui consiste à estimer les paramètres des chaînes de Markov (probabilités de transitions) et des densités d'observation associées aux états. Il s'agit avec un corpus d'apprentissage, contenant un étiquetage par sous-unités acoustiques du signal temporel, de maximiser la vraisemblance que le modèle HMM ait produit la suite d'observations. Il existe plusieurs algorithmes pour faire cela : Apprentissage heuristique basé sur l'algorithme de

Viterbi ou bien L'algorithme *Baum- Welch* qui est un cas particulier d'Expectation-Maximisation (EM).

7.3.4.3 Décodage

Le décodage de la parole par un HMM consiste à trouver la séquence de modèles qui maximise la probabilité de généré une séquence d'observations donnée O. Donc le décodage par HMM revient à détermine la meilleur séquence d'états.

$Q^*=$ $(q_1^*, q_2^*, \ldots, q_T^*)$ pouvant engendrer la séquence d'observation O=(O1,O2,….,O_T) :

$$Q^* = \arg\max_Q P(O \backslash Q) = arg\,\max \Pi 0_Q \prod_{t=1}^{T} a_{qt-1qt} \cdot b_{qt}\,(ot) \qquad 1.27$$

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

L'algorithme de viterbi est une variante stochastique de la programmation dynamique propose de simplifier l'arbre au fur et à meure de sa construction. En effet lors de sons déroulement on se trouve rapidement avec des branches proposant les mêmes substitutions mais avec des probabilités différentes. Plusieurs hypothèses peuvent se retrouver dans le même état au même instant. L'algorithme de viterbi stipule qu'il n'est pas nécessaire de dérouler les hypothèses de plus faible probabilité car elles ne peuvent plus être candidates pour décrire le message le plus probable.

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

Initialisation : $\delta_i(O1)=\pi_i$ 1.28

Récursion : pour se trouver dans l'état i à l'instant t, le processus markovien se trouvait forcément dans un état j à l'instant t-1 pour lequel une transition vers l'état i est possible : $\alpha_{ji} > 0$ d'après le principe d'optimisation de bellman

$\delta_i(Ot) = \max_i(\delta_i(Ot\text{-}1).\ \alpha_{ji}).b_i(Ot).$ 1.29

Terminaison : la vraisemblance des observations correspondant à la meilleure hypothèse est obtenue en recherchant l'état i qui maximise la valeur $\delta_i(OT)$ à la dernière observation OT.

$$P(O\backslash Q^*)=\max_i(\delta_i(OT)) \quad 1.30$$

7.3.5 Limitation des HMM

L'utilisation des HMM en RAP repose sur plusieurs hypothèses simplificatrices celles-ci sont sertes mais elles constituent également des points faibles des HMM.

La modélisation de la durée des phonèmes n'est qu'implicitement contenue au travers des probabilités de transitions entre les états.

L'hypothèse d'indépendance conditionnelle des observations est irréaliste une solution efficace et largement répandue consiste à prendre en compte les dérivées premières Δ et secondes ΔΔ des paramètres. Une deuxième solution est de modéliser explicitement la corrélation entre les vecteurs d'observations successifs.

7.4 Les approches connexionnistes pour la reconnaissance vocale

La reconnaissance de la parole est parmi les domaines auxquels ont été appliquées des techniques connexionnistes. La description des algorithmes connexionnistes s'appuie presque toujours sur une modélisation plus ou moins fidèle des neurones et de leurs interconnexions. Les réseaux de neurones constituent un outil de classification flexible. Ils ont été appliqués à de nombreux problèmes où la modélisation explicite des relations cause-effet a résisté à toute analyse conventionnelle, en particulier dans le cadre de la reconnaissance de parole. Il nous paraît donc utile de souligner leurs caractéristiques principales ayant une implication directe dans la reconnaissance de parole.

Les réseaux de neurones sont apparus dans les années 40 lorsque McCulloch et Pitts proposèrent un modèle de calcul (automate à seuil) basé sur de simples éléments logiques mimant les relations neuronales. En 1949, D. Hebb énonça sa règle d'apprentissage, aussi connue sous le nom de loi delta, et permit ainsi le développement d'intérêt que la communauté scientifique lui porta jusque dans les années 60 où elle tomba en léthargie. Ce n'est que dans les années 80 avec les travaux de Hopfield, que l'approche neuronale connait un regain d'intérêt. En 1986, généralisation de la loi delta et développement d'une méthode efficace d'entraînement des réseaux multicouches.

7.4.1 Système neuronal pour la reconnaissance vocale

Les réseaux de neurones (RN) constituent la seconde technique pour la reconnaissance de la parole. Différents types de réseaux sont utilisés et parmi eux, les perceptrons multi-couches ou les réseaux récurrents. Ces méthodes appliquées à la reconnaissance de la parole sont largement détaillées dans [45].

Les perceptron multi-couches ont de grandes capacités de classification et ont montré leurs aptitudes en parole, notamment pour les mots isolés. De plus, par rapport à l'ensemble des systèmes connexionnistes, ils ont l'avantage d'être basés sur des principes simples et relativement maîtrisables. Contrairement aux réseaux récurrents par exemple, leur temps de convergence peut être relativement rapide. Il est également possible de détecter le moment où l'algorithme d'apprentissage n'est plus capable d'améliorer les performances, ce qui permet d'optimiser le temps d'apprentissage. [72]

7.4.2 Principe de réseau de neurone

Les éléments de base d'un neurone sont décrits dans la figure ci-dessous. Il est caractérise principalement par un ensemble de poids associés aux connexions du neurone, et La fonction f(s) appelée *fonction d'activation* qui peut avoir de multiples formes (fonction à seuil, fonction à saturation, fonction sigmoïdale….). Les plus courantes sont les fonctions de forme sigmoïdale. Où les x_i correspondent à l'entrée du neurone et y à sa sortie. Les valeurs d'entrée sont multipliées par leur poids correspondant et additionnées pour obtenir la somme S. [63]

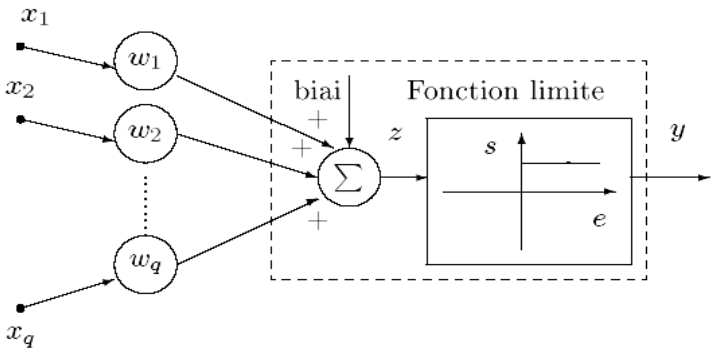


Figure1.9 Un neurone formel

Les réseaux les plus courants en reconnaissance de parole restent les réseaux de type perceptron multicouches (MLP). Lippman, montre en 1987, que 3 couches sont suffisantes pour générer n'importe quelle séparation de l'espace d'entrée. Donc l'utilisation de réseaux ayant une seule couche cachée, tel celui décrit dans la figure (1.10) est suffisante pour la reconnaissance de la parole.

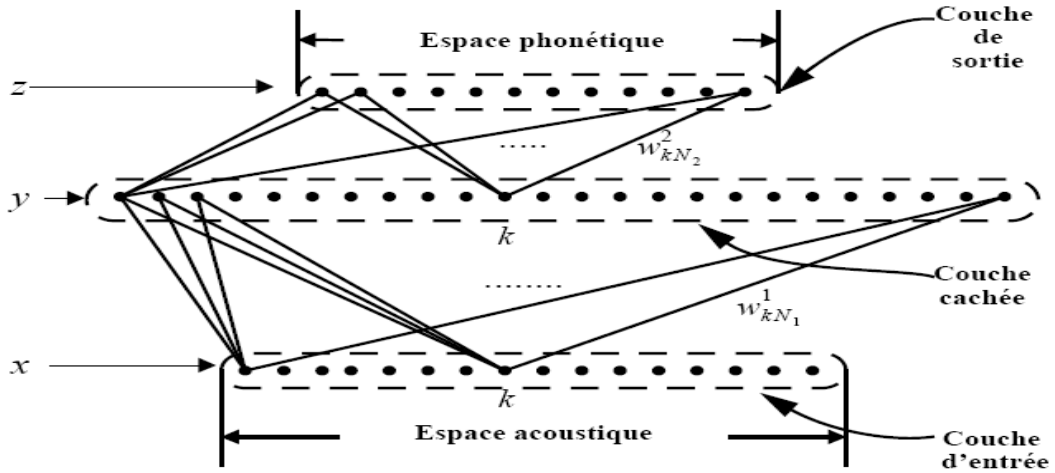


Figure1.10 Réseau de neurone à une seule couche cachée

Une autre classe de perceptrons multi-couches, les *réseaux récurrents*, permet également de réaliser des systèmes dynamiques pour la reconnaissance de la parole. [63]

On trouve aussi les réseaux à délais (Time Delay Neural Networks, TDNN) qui ont été largement utilisés pour la reconnaissance de phonèmes pour plusieurs langues, dont le français, l'anglais, et le japonais. Ils ont également été utilisés dans le cadre de la reconnaissance de mots isolés. Un réseau à délais est en fait un cas particulier de perceptron multi-couches à connexions contraintes. [55]

7.4.3 Apprentissage

L'apprentissage d'un réseau de neurones signifie qu'il change son comportement de façon à lui permettre de se rapprocher d'un but défini. Ce but est normalement l'approximation d'un ensemble d'exemples ou l'optimisation de l'état du réseau en fonction de ses poids pour atteindre l'optimum d'une fonction économique fixée a priori. L'apprentissage du réseau de neurones consiste à minimiser un critère d'erreur. Ce critère peut être quadratique:

$$E= 0.5 \sum_{n=1}^{Nx} \sum_{l=1}^{Nq} (z_l(n, \theta) - d_l(n))^2 \qquad 1.31$$

Où θ représente l'ensemble des paramètres du réseau (principalement les poids $w_{ij}^{1,2}$) $z_l(n, \theta)$ représente la valeur obtenue à la l[ième] sortie lorsque l'on applique le n[ième] vecteur acoustique à l'entrée, et $d_l(n)$ correspond à la sortie désirée.

Il existe d'autres types de minimisation comme celle basée sur l'entropie relative :

$$E=0.5 \sum_{n=1}^{Nx} \sum_{l=1}^{Nq} \left[(1 + d_l(n)) \ln \left[\frac{1+dl(n)}{1+zl(n,\theta)} \right] + (1 - d_l(n)) \ln \left[\frac{1-d_l(n)}{1-z_l(n,\theta)} \right] \right] \qquad 1.32$$

La minimisation du critère d'erreur choisi est effectuée en dérivant les équations précédentes par rapport aux différents poids. On en déduit ainsi la loi delta généralisée qui modifie les poids du réseau.

Pour la couche de sortie, nous avons :

$$\Delta w_{ij}^2 = \tau(d_i - f(s_i^2))f'(s_i^2)y_j \qquad 1.33$$

Et pour la couche cachée :

$$\Delta w_{ij}^1 = \left[\sum_j (w_{ij}^2 \Delta w_{ij}^2)\right] f'(s_i^2)x_j \qquad 1.34$$

Où τ représente le taux d'apprentissage. [64]

7.4.4 Surentraînement (sur-apprentissage)

Le surentraînement est l'un des phénomènes que l'on peut rencontrer lors de l'apprentissage de réseau de neurones. Si le réseau possède trop de degrés de liberté par rapport à la complexité du problème, il aura tendance à apprendre les exemples du problème qu'on lui soumet à l'entraînement, et cela au détriment du caractère de généralisation que l'on attend du système.

La méthode la plus fréquemment utilisée pour palier ce défaut consiste à constamment tester l'efficacité du réseau sur une partie de données non utilisées pour l'entraînement. Cette méthode porte le nom de validation croisée (cross validation). Nous montrons dans la figure(1.11) un exemple typique de comportement d'entraînement. [64]

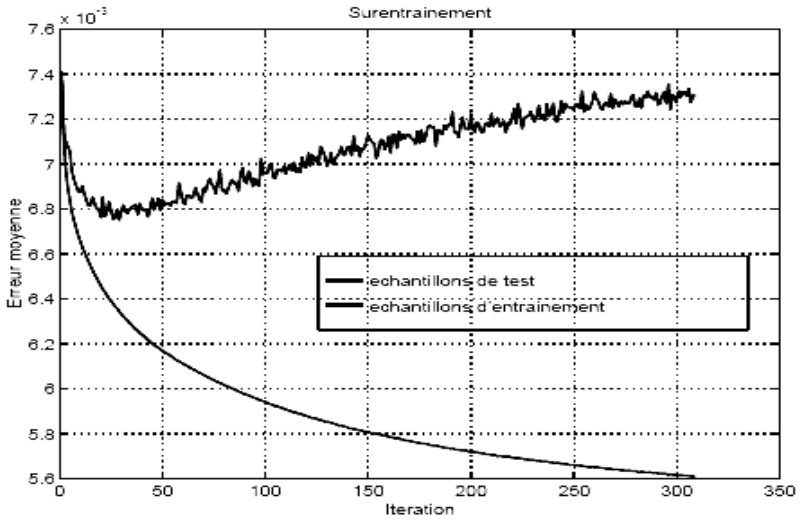


Figure1.11 Courbes d'erreurs sur les échantillons de test et d'entraînement.

7.4.5 Utilisation en classification

L'utilisation de réseaux de neurones en vue de la classification de données se présente comme suit: chaque sortie du réseau correspond à une des classes. Ceci implique directement que le nombre de classes soit préalablement fixé. En reconnaissance de parole, nous choisirons évidemment les classes phonétiques. L'entrée du réseau correspond généralement

aux vecteurs acoustiques que l'on désire classer. Cependant, l'analyse statistique montre que les probabilités dépendent des vecteurs voisins.

Une approche analytique prendre en compte les vecteurs voisins dans la définition des probabilités. Une autre façon de tenir compte de cet effet est d'adjoindre les vecteurs acoustiques temporellement proches (entrée contextuelle) ou une représentation de l'état précédent à l'entrée du réseau de neurones. [64]

Les méthodes connexionnistes donnent des résultats comparables aux HMM, ils présentent un intérêt certain pour le traitement automatique de la parole essentiellement pour leur capacité d'apprentissage discriminant, ils ont l'avantage d'être robustes au bruit, d'être adaptatives (apprendre et généraliser des informations), d'être matériellement réalisables, mais il faut mentionner que leur développement nécessite beaucoup d'expérience même avec les outils logiciels de développement maintenant disponibles, la façon dont ils prennent en compte la structure temporelle de la parole n'est pas encore totalement satisfaisante, malgré les diverses solutions proposées.

7.5 Modèles hybrides

L'hybridation de plusieurs méthodes afin d'exploiter les avantages de chacune, semble constituer une approche intéressante pour la robustesse des systèmes de RAP. Ainsi, on voit apparaître des systèmes combinant plusieurs systèmes de reconnaissance comme, par exemple, un HMM et un réseau de neurones (ou *Neural Network* (NN)).

La combinaison la plus courante de nos jours est l'hybridation HMM et réseaux connexionnistes (ou *Neural Networks* (NN)) [45]. La principale raison est que ces deux systèmes présentent des avantages complémentaires. Les HMM ont une grande capacité à traiter les évènements dans le temps, les réseaux de neurones sont experts dans la classification de formes statiques.

On trouve aussi des systèmes hybrides MMC\RNA pour des basées de données allant de petits lexiques aux moyens lexiques, ayant comme objectif la reconnaissance de la parole indépendante du locuteur [54]. La reconnaissance de la parole par système hybride MMC/RNA se pratique selon le même principe que la reconnaissance MMC. Le réseau de neurone est simplement utilisé comme estimateur de probabilités locales pour les MMC. Après division par les estimateurs de probabilités a priori, le réseau de neurones (PMC dans ce cas) fournit donc les vraisemblances normalisées $p(x_n \backslash q_k)/p(x_n)$ qui sont utilisées, soit dans un algorithme Viterbi, soit dans une récurrence "avant" , afin d'estimer $P(M_j \backslash X)$ pour tous les

modèles M_j possibles et d'assigner la séquence X au modèle M_k conduisant au maximum de probabilités a posteriori. [54]

On trouve aussi, l'hybridation des NN avec des EA a été largement étudiée depuis une dizaine d'années (voir Le système hybride RNA\AG: à partir d'une population d'individus entrainés par algorithme génétique, une sélection du meilleur classifieur neuronal de type PMC). [4]

8. Conclusion

Dans ce chapitre, nous avons décrit les premiers modules du traitement de la parole en vue de sa reconnaissance. Nous avons abordé la technique de paramétrisation du signal de la parole, les outils et les algorithmes les plus répandus de nos jours. Malgré le nombre important des techniques utilisés pour la réalisation des systèmes de reconnaissance les voies qui restent à explorer sont nombreuses. Cependant on peut distinguer quelques modèles inspirés de la nature qui semblent faire l'unanimité des chercheurs comme les réseaux de neurones, les colonies de fourmis, les algorithmes génétiques ... , ces modèles sont de bons candidats pour la reconnaissance de la parole et ont contribués à l'amélioration des résultats des systèmes de RAP.

Dans le chapitre suivant nous présentons les algorithmes évolutionnaires particulièrement les algorithmes génétiques, qui inspirer de la biologie, et comment utiliser cette technique dans la réalisation des systèmes de reconnaissance vocale.

Chapitre 2

Les algorithmes Evolutionnaires

1. Introduction

L'évolution biologique a engendré des systèmes vivants extrêmement complexes. Elle est le fruit d'une altération progressive et continue des êtres vivants au cours des générations et s'opère en deux étapes la sélection et la reproduction.

La sélection naturelle est le mécanisme central qui opère au niveau des populations, en sélectionnant les individus les mieux adaptés à leur environnement. La reproduction implique une mémoire l'hérédité, sous la forme de gènes. Ce matériel héréditaire subit, au niveau moléculaire, des modifications constantes par mutations et recombinaisons, aboutissant ainsi à une grande diversité.

Ces principes, présentés pour la première fois par Darwin, ont inspiré bien plus tard les chercheurs en informatique. Ils ont donné naissance à une classe d'algorithmes regroupés sous le nom générique d'*Algorithmes Evolutionnaires* (ou *Evolutionary Algorithms* (EA)). Les sections suivantes en présentent les fondements.

Les algorithmes évolutionnaires sont classés parmi les algorithmes d'optimisation par recherche probabiliste basés sur le modèle de l'évolution naturelle, qui s'appliquent à des espaces de recherche non-standards. Ces algorithmes sont utilisés pour résoudre des problèmes difficiles où ils permettent souvent de proposer de meilleures solutions que les algorithmes classiques d'optimisation lorsque ces derniers s'appliquent. Ils modélisent une population d'individus par des points dans un espace. Un individu est codé dans un génotype composé de gènes correspondant aux valeurs des paramètres du problème à traiter. Le génotype de l'individu correspond à une solution potentielle au problème posé, le but des EA est d'en trouver la solution optimale.

Les algorithmes évolutionnaires (AE) suivent un principe général quelque soit le type de problème à résoudre, ils commencent par initialisation de la population de façon dépendante du problème à résoudre (l'environnement), puis évolue de génération en génération à l'aide d'opérateurs de sélection, de recombinaison et de mutation. L'environnement a pour charge d'évaluer les individus en leur attribuant une performance ou fitness. Cette valeur favorisera la sélection des meilleurs individus, en vue, après reproduction (opérée par la mutation et/ou recombinaison), d'améliorer les performances globales de la population. [72]

2. Principales familles des algorithmes évolutionnaires

Historiquement, trois grandes familles d'algorithmes ont été développées indépendamment, entre la moitié des années 1960 et 1970. Les premières méthodes furent les stratégies d'évolution . Proposées par I. Rechenberg en 1965, pour résoudre des problèmes d'optimisations continus. L'année suivante, Fogel, Owens et Walsh conçoivent la programmation évolutionnaire comme une méthode d'intelligence artificielle pour la conception d'automates à états finis. Enfin, en 1975, J. H. Holland propose les premiers algorithmes génétiques , pour l'optimisation combinatoire. Le travail de D. E. Goldberg [41] rendre les algorithmes génétiques populaires.

Par la suite, ces différentes approches ont beaucoup évoluées et se sont rapprochées, et regroupées sous le terme générique d'algorithmes évolutionnaires. Aujourd'hui, la littérature sur le sujet est extrêmement abondante, et ces algorithmes sont considérés comme un domaine de recherche très prolifique.

2.1 Stratégies d'évolution

Dans sa version de base, l'algorithme manipule itérativement un ensemble de vecteurs de variables réelles, à l'aide d'opérateurs de mutation et de sélection. La sélection s'effectue par un choix déterministe des meilleurs individus, selon l'échelle de valeur de la fonction objectif. L'étape de mutation est classiquement effectuée par l'ajout d'une valeur aléatoire, tirée au sein d'une distribution normale.

Un algorithme représentatif des stratégies d'évolution est l'évolution différentielle. Dans cette classe de méthode, on utilise la différence pondérée entre sous-populations pour biaiser un opérateur de mutation différentiel.

2.2 Programmation évolutives

Historiquement, ces algorithmes étaient conçus pour des problèmes d'apprentissage à partir d'automates à états finis et n'utilisaient que des opérateurs de mutation et de remplacement. Cependant, aujourd'hui ils ne se limitent plus à une représentation, mais n'utilisent toujours pas d'opérateur de croisement. Ils diffèrent des stratégies d'évolution en ce qu'ils privilégient des opérateurs de remplacement stochastiques.

2.3 Algorithmes génétiques

L'algorithme génétique est un exemple de procédure pseudo aléatoire qui utilise un choix aléatoire pour guider une exploration dans un espace de recherche. [49] En effet, l'AG est une technique de solution aux problèmes d'optimisation combinatoire fondée sur les principes d'évolution naturelle et d'héritage. A partir d'un ensemble de solutions (nommées population ou ensemble de chromosomes parents), l'algorithme se sert des opérateurs génétiques pour obtenir de nouvelles solutions (ensemble d'enfants), souvent meilleures selon un critère d'évaluation donné. Le processus prend après comme population la nouvelle génération et ainsi de suite. La condition d'arrêt du processus et la grandeur de la population initiale sont définies selon des tests expérimentaux, en fonction des connaissances du problème traité. [50]

2.4 Programmation génétiques

Ces algorithmes utilisent une représentation en arbres d'expressions logiques, du fait qu'ils sont historiquement appliqués à l'apprentissage statistique et la modélisation. La programmation génétique ce sont les concepts d'évolution et d'algorithme génétique appliqué à la programmation d'ordinateurs. Elle s'intéresse spécifiquement à la construction automatique de programmes.

De ces méthodes classiques ont dérivé différentes techniques mélangeant les méthodes d'évolution des unes et des autres. Impossible à classer dans l'une des quatre familles citées ci-dessus, elles sont néanmoins considérées comme des EA. [72]

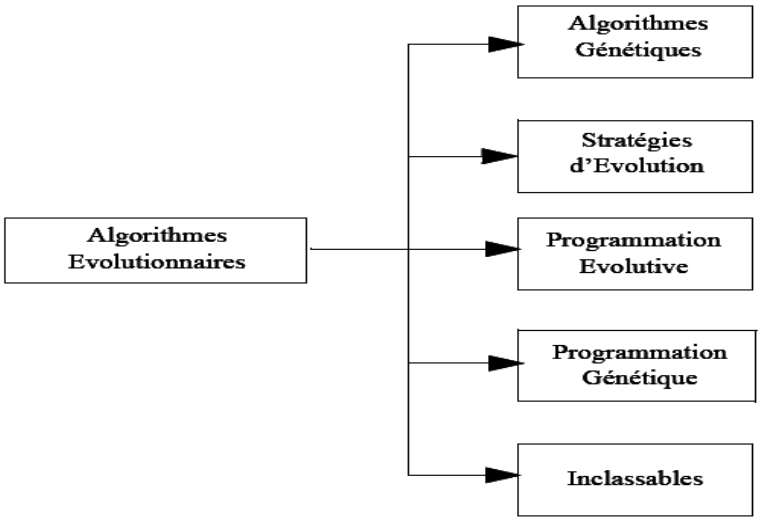


Figure 2.1 : Différentes branches des algorithmes évolutionnaires.

3. Les Algorithmes génétiques

Au début des années 1960 John Holland a commencé à s'intéresser à ce qui allait devenir les algorithmes génétiques. Ses travaux ont trouvé un premier aboutissement en 1975. [47] Holland poursuivait un double objectif. Améliorer la compréhension des processus naturels d'adaptation, et concevoir des systèmes artificiels possédant des propriétés similaires aux systèmes naturels. Il expliqua comment ajouter de l'intelligence dans un programme informatique avec les croisements (échangeant le matériel génétique) et la mutation (source de la diversité génétique).

Darwin proposa une théorie qui clarifie l'évolution des espèces en mettant en avant quatre lois:

- La loi de croissance et de reproduction.
- La loi d'hérédité qu'implique quasiment la loi de reproduction.
- La loi de variabilité, résultant des conditions d'existence.
- La loi de multiplication des espèces qui amène la lutte pour l'existence et qui a pour conséquence la sélection naturelle.

En 1989 Goldberg [41] ajouta à la théorie des algorithmes génétiques les idées suivantes:

- un individu est lié à un environnement par son code d'ADN.
- une solution est liée à un problème par son indice de qualité.

L'idée fondamentale est la suivante: le pool génétique d'une population donnée contient potentiellement la solution, ou plutôt une meilleure solution, à un problème adaptatif donné. Cette solution n'est pas exprimée car la combinaison génétique sur laquelle elle repose est dispersée chez plusieurs individus. Ce n'est que par l'association de ces combinaisons génétiques au cours de la reproduction que la solution pourra s'exprimer.

L'originalité des travaux de Holland repose en particulier sur le fait qu'il n'a pas considéré les seules mutations comme source d'évolution mais aussi et surtout les phénomènes de croisement (crossover) c'est en croisant les solutions potentielles existant au sein du pool génétique que l'on peut se rapprocher de l'optimum. [web1]

Les algorithmes génétiques, initiés dans les années 1970 par John Holland, sont des algorithmes d'optimisation s'appuyant sur des techniques dérivées de la génétique et des mécanismes d'évolution de la nature: croisement, mutation, sélection.

Nos travaux consisteront à définir les algorithmes génétiques, et à les comparer aux méthodes déterministes afin de pouvoir cerner leur utilité en fonction du problème de la reconnaissance automatique de la parole arabe.

3.1 Les éléments des algorithmes génétiques

Les algorithmes génétiques fournissent des solutions aux problèmes n'ayant pas de solutions calculables en temps raisonnable de façon analytique ou algorithmique. Avant de donner le principe de fonctionnement des algorithmes génétiques il est utile d'introduire le vocabulaire que nous allons utiliser. [66]

- gène : Un gène est une suite de bases azotées (adénine(A), cytosine(C), guanine(G) et le la thymine(T)) qui contient le code d'une protéine donnée. On appellera gène la suite de symboles qui codent la valeur d'un variable. Dans le cas général, un gène correspond à un seul symbole (0 ou 1 dans le cas binaire). Une mutation changera donc systématiquement l'expression du gène muté.

- chromosome: Un chromosome est constitué d'une séquence finie de gènes qui peuvent prendre des valeurs appelées allèles qui sont prises dans un alphabet qui doit être judicieusement choisi pour convenir du problème étudié. Chaque gène à une position dans le chromosome. Cette position est appelée locus.

- individu, organisme: un organisme biologique est plus qu'un génome, c'est aussi une forme, un phénotype qui est le produit de l'activité des gènes. Dans le cadre d'un AG traditionnel, l'individu est réduit à son génome, à un ensemble de caractéristiques élémentaires.

- locus : signifie lieu en latin. Le locus est la position du gène dans le chromosome. Ce terme vaut pour le GA aussi bien que pour les systèmes biologiques.

- allèle : symbole attaché à un gène. Alors que l'alphabet du code génétique naturel est composé des quatre lettres ATCG nous pouvons choisir n'importe quel alphabet pour un GA (un alphabet binaire est couramment utilisé).

- population : un groupe d'organismes artificiels ou naturels.

- Génération : une génération est une population à un instant donné *t*, les algorithmes génétiques faisant évoluer les populations, cette évolution est effectuée par des opérateurs de sélection, de croisement et de mutation.

- La fitness (la fonction d'évaluation): La fitness est la pièce maitresse dans le processus d'optimisation. C'est l'élément qui permet aux algorithmes génétiques de prendre en compte un problème donné. Pour que le processus d'optimisation puisse donner des bons résultats, il faut concevoir une fonction de fitness permettant une évaluation pertinente des solutions d'un problème sous forme chiffrée. Cette fonction est déterminée en fonction du problème à résoudre et du codage choisi pour les chromosomes. Pour chaque chromosome, elle attribue

une valeur numérique, qui est supposée proportionnelle à la qualité de l'individu en tant que solution. Le résultat renvoyé par la fonction d'évaluation va permettre de sélectionner ou de refuser un individu selon une stratégie de sélection.

3.2 Principe générale des algorithmes génétiques

Les AG ont un principe de fonctionnement simple, qui suit les étapes suivantes :

1- Codage du problème sous forme d'une structure de données (Les codages binaires ont été très utilisés). La qualité du codage des données conditionne le succès des algorithmes génétiques.

2- Génération aléatoire d'une population. Un mécanisme de génération doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures.

Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

3- Calcul d'une valeur d'adaptation pour chaque individu. Elle sera fonction à optimiser. Celle-ci retourne une valeur appelée fitness ou fonction d'évaluation de l'individu.

4. Sélection des individus devant se reproduire en fonction de leurs parts respectives dans l'adaptation globale.

5- Reproduction des séquences en fonction de son adaptation.
Faire l'opération de croisement aléatoirement de quelque paire de séquences.
Faire l'opération de mutation d'un bit choisi aléatoirement dans une ou plusieurs séquences.

6. Sur la base de ce nouveau pool génétique, on repart à partir du point 3. [41]

On peut également exprimer le fonctionnement d'un algorithme génétique en se référant aux notions de génotypes (GTYPE) et phénotypes (PTYPE) :

On sélectionne des paires de GTYPE en fonction de l'adaptation de leurs PTYPE respectifs.

1. On applique les opérateurs génétiques (reproduction, croisement et mutation) pour créer de nouveaux GTYPE.
2. On développe les GTYPE pour obtenir les PTYPE de la nouvelle génération et on repart en 1.

En effet, quand on utilise les algorithmes génétiques, aucune connaissance de la manière dont résoudre le problème n'est requise, il est seulement nécessaire de fournir une fonction permettant de coder une solution sous forme de gènes (et donc de faire le travail inverse) ainsi

que de fournir une fonction permettant d'évaluer la pertinence d'une solution au problème donné. Cela en fait donc un modèle minimal et canonique pour n'importe quel système évolutionnaire et pour n'importe quel problème pouvant être abordé sous cet angle, sous ce paradigme. Cette représentation nous permet donc d'étudier des propriétés quasiment impossibles à étudier dans leur milieu naturel, ainsi que de résoudre des problèmes n'ayant pas de solutions calculables en temps raisonnables si on les aborde sous d'autres paradigmes, avec des performances quantifiables, facilement mesurables et qu'on peut confrontés aux autres stratégies de résolution. [web2]

4. La différances des algorithmes génétiques par rapport aux autres paradigmes

Les principales différences des algorithmes génétiques par rapport aux autres paradigmes sont les suivantes :

• On utilise un codage des informations, on représente toutes les caractéristiques d'une solution par un ensemble de gènes, c'est-à-dire un chromosome, sous un certain codage (binaire, réel, code de Gray ...), valeurs qu'on concatène pour obtenir une chaîne de caractères qui est spécifique à une solution bien particulière (il y a une bijection entre la solution et sa représentation codée)

• On traite une population "d'individus", de solutions, cela introduit donc du parallélisme.

• L'évaluation de l'optimalité du système n'est pas dépendante vis-à-vis du domaine.

• On utilise des règles probabilistes, il n'y a pas d'énumération de l'espace de recherche, on explore une certaine partie en étant guidé par un semi-hasard, en effet des opérateurs comme la fonction d'évaluation permet de choisir de s'intéresser à une solution qui semble représenter un optimum local, on fait donc un choix délibéré, puis de la croiser avec une autre solution optimale localement, en général la solution obtenue par croisement est meilleure ou du même niveau que ses parents, mais ce n'est pas assuré, cela dépend des aléas du hasard, et cela et d'autant plus vrai pour l'opérateur de mutation qui ne s'applique qu'avec une certaine probabilité et dans le cas où il s'applique choisit aléatoirement sur quel locus introduire des modifications.[71]

5. Stratégies d'évolution d'un algorithme génétique

5.1 Le Codage

Les individus intervenant dans un algorithme génétique étaient codés sous forme de chaînes de bits. Chaque paramètre d'une solution est assimilé à un gène, toutes les valeurs qu'il peut prendre sont les allèles de ce gène, on doit trouver une manière de coder chaque

allèle différent de façon unique (établir une bijection entre l'allèle réel et sa représentation codée).

Un chromosome est une suite de gène, on peut par exemple choisir de regrouper les paramètres similaires dans un même chromosome (chromosome à un seul brin) et chaque gène sera repérable par sa position. Chaque individu est représenté par un ensemble de chromosomes. Une population est un ensemble d'individus.

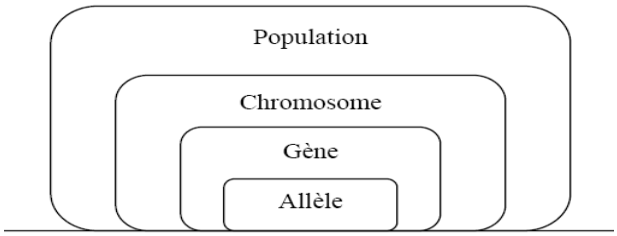


Figure 2.2 : Les niveaux d'organisation d'un algorithme génétique

Il existe trois types principaux de codage utilisables; qui on peut faire le passage de l'un à l'autre relativement :

5.1.1 Le codage binaire: C'est le codage plus utilisé. Chaque gène est codé par un alphabet binaire de (0, 1), les chromosomes qui sont des suites de gènes sont représentés par des tableaux de gènes et les individus d'un espace de recherche sont représentés par des tableaux de chromosomes.

5.1.2 Le codage réel: Cela peut-être utile notamment dans le cas où l'on recherche le maximum d'une fonction réelle.

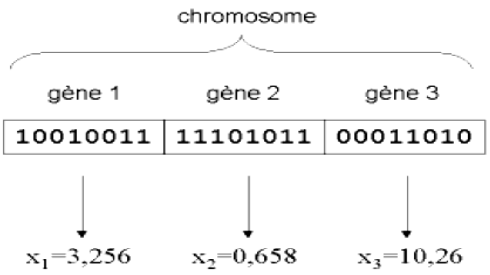


Figure 2.3 : illustration schématique du codage des variables réelles.

5.1.3 Le codage de Gray: Dans le cas d'un codage binaire on utilise souvent la "distance de Hamming" comme mesure de la dissimilarité entre deux éléments de population, cette mesure compte les différences de bits de même rang de ces deux séquences. C'est la que le codage binaire commence à montrer ses limites. En effet, deux éléments voisins en termes de distance de Hamming ne codent pas nécessairement deux éléments proches dans l'espace de recherche. Cet inconvénient peut être évité en utilisant un codage de Gray qui a comme propriété

qu'entre un élément n et un élément $n + 1$, donc voisin dans l'espace de recherche, un seul bit diffère.

5.2 Création de la population initiale

Dans n'importe quel problème d'optimisation, la connaissance des points de départ conditionne la rapidité de la convergence vers l'optimum. Si la position de l'optimum dans l'espace d'état est totalement inconnue, il est naturel de générer aléatoirement des individus en faisant des tirages uniformes dans chacun des domaines associés aux composantes de l'espace d'état, en veillant à ce que les individus produits respectent les contraintes. Si par contre, des informations a priori sur le problème sont disponibles, il parait bien évidemment naturel de générer les individus dans un sous-domaine particulier afin d'accélérer la convergence.

5.3 Gestion des contraintes

La gestion des contraintes liées au problème est une tâche difficile et sensible pour laquelle l'utilisateur aura à arbitrer entre différentes techniques suivant son appréhension du problème. Ces contraintes sont de diverses natures et peuvent intervenir sur l'espace d'état (contraintes de signe, restrictions à un sous-espace, etc..), ou de manière plus complexe dans le problème lui-même.

Dans le cas de contraintes sur l'espace dans lequel doit se faire la recherche, on pourra sélectionner les individus rapidement (sans avoir à les réévaluer) par différentes méthodes. Un individu hors champ pourra être :

- Rejeté brutalement et remplacé par un autre individu tiré aléatoirement sur l'espace admissible.
- Ramené à la frontière la plus proche (principe du mur).
- Reporté à la frontière diamétralement opposée à la frontière la plus proche (principe du tore).

Il est bon de noter qu'il peut être préférable de garder des individus hors champ mais qui conservent une direction originale, plutôt que de confiner la population à un sous-espace. Dans l'hypothèse où la gestion des contraintes ne peut se faire directement, les contraintes peuvent être incluses dans le critère à optimiser sous forme de pénalités. Ainsi, un individu qui viole une contrainte se verra attribuer une mauvaise fitness et sera donc éliminé, avec une forte probabilité, par le processus de sélection. Cette façon de gérer les contraintes est difficile. En effet, inclure les contraintes dans la fonction d'évaluation peut se faire de diverses

façons, et un dosage s'impose pour ne pas favoriser la recherche de solutions admissibles au détriment de la recherche de l'optimum ou inversement. On risque alors de fournir une solution admissible certes, mais éloignée de l'optimum.

Un algorithme génétique doit préserver les points suivants :

- Diversité génétique qui désigne la variété des génotypes présents dans la population. Si les individus sont identiques la diversité génétique devient nulle. Si on parle sur la diversité génétique on parle alors de convergence de l'algorithme. Lorsque la diversité génétique devient très faible, il y a très peu de chances pour qu'elle augmente à nouveau, si cela se produit trop tôt, la convergence à lieu vers un optimum local, on parle alors de convergence prématurée. Il faut donc préserver la diversité génétique, sans pour autant empêcher la convergence.
- A chaque étape de l'algorithme, il faut effectuer le compromis entre explorer l'espace de recherche, afin d'éviter de stagner dans un optimum local, et exploiter les meilleurs individus obtenus, afin d'atteindre de meilleures valeurs aux alentours. Trop d'exploitation entraîne une convergence vers un optimum local, alors que trop d'exploration entraîne la non-convergence de l'algorithme.

Les opérations de sélection et de croisement sont des étapes d'exploitation, alors que l'initialisation et la mutation sont des étapes d'exploration. On peut ainsi régler les parts respectives d'exploration et d'exploitation en jouant sur les divers paramètres de l'algorithme. Malheureusement, il n'existe pas de règles universelles de réglages et seuls des résultats expérimentaux donnent une idée du comportement des divers composantes des algorithmes.

5.4 Les opérateurs génétiques

Trois opérateurs jouent un rôle prépondérant dans la possible réussite d'un AG: L'opérateur de sélection, l'opérateur de croisement et l'opérateur de mutation. Si le principe de chacun de ces opérateurs est facilement compréhensible, il est toutefois difficile d'expliquer l'importance isolée de chacun de ces opérateurs dans la réussite de l'AG. Chacun de ces opérateurs agit selon divers critères qui sont propres à lui (valeur sélective des individus, probabilité d'activation de l'opérateur, ...).

5.4.1 L'opérateur de sélection

Cet opérateur est chargé de définir quels seront les individus de la population P qui vont être dupliqués dans la nouvelle population P' et vont servir de parents (application de l'opérateur de croisement). Il permet d'identifier statistiquement les meilleurs individus d'une population et d'éliminer les mauvais. On trouve dans la littérature un nombre important de principes de sélection plus ou moins adaptés aux problèmes qu'ils traitent. Les méthodes de sélection les plus utilisées sont les suivants :

5.4.1.1 La sélection par roulette de casino (loterie biaisée)

C'est la sélection naturelle la plus connue et la plus utilisés dans la littérature. [41] Chaque chromosome occupe un secteur de roulette dont l'angle est proportionnel à son indice de qualité. Un chromosome est considéré comme bon aura un indice de qualité élevé, un large secteur de roulette et alors il aura plus de chance d'être sélectionné. On fait tourner la roue et quand elle cesse de tourner, on sélectionne l'individu correspondant au secteur désigné par un curseur; qui pointe sur un secteur particulier de celle-ci après qu'elle se soit arrêtée de tourner. Cette méthode, bien que largement répandue, a pas mal d'inconvénients :

• En effet, elle a une forte variance. Il n'est pas impossible que sur n sélections successives destinées à désigner les parents de la nouvelle génération *P'*, la quasi-totalité, voire pire la totalité des n individus sélectionnés soient des individus ayant une fitness vraiment mauvaise et donc que pratiquement aucun individu voire aucun individu a forte fitness ne fasse partie des parents de la nouvelle génération. Ce phénomène est bien sûr très dommageable car cela va complètement à l'encontre du principe des algorithmes génétiques qui veut que les meilleurs individus soient sélectionnés de manière à converger vers une solution la plus optimale possible.

• A l'inverse, on peut arriver à une domination écrasante d'un individu localement supérieur.

Ceci entrainant une grave perte de diversité. Imaginons par exemple qu'on ait un individu ayant une fitness très élevée par rapport au reste de la population, disons dix fois supérieure, il n'est pas impossible qu'après quelques générations successives on se retrouve avec une population ne contenant que des copies de cet individu. Le problème est que cet individu avait une fitness très élevée, mais que cette fitness était toute relative, elle était très élevée mais seulement en comparaison des autres individus. On se retrouve donc face à problème connu sous le nom de convergence prématurée, l'évolution se met donc à stagner et on atteindra alors jamais l'optimum, on restera bloqué sur un optimum local.

5.4.1.2 La sélection élitiste

C'est le principe de sélection le plus simple, il consiste à attribuer à chaque individu son classement par ordre d'adaptation. Le meilleur (c'est à dire celui qui possède la meilleure fitness) sera numéro un, et ainsi de suite. On tire ensuite une nouvelle population dans cet ensemble d'individus ordonnés, en utilisant des probabilités indexées sur les rangs des individus. Cette procédure semble toutefois assez simpliste et exagère le rôle du meilleur élément au détriment d'autres élément potentiellement exploitables. Le second, par exemple, aura une probabilité d'être sélectionné nettement plus faible que celle du premier, bien qu'il puisse se situer dans une région d'intérêt. [16]

Il est inutile de préciser que cette méthode est encore pire que celle de la loterie biaisée dans le sens où elle amènera à une convergence prématurée encore plus rapidement et surtout de manière encore plus sûre que la méthode de sélection de la loterie biaisée, en effet la pression de la sélection est trop forte, la variance nulle et la diversité inexistante, du moins le peu de diversité qu'il pourrait y avoir ne résultera pas de la sélection mais plutôt du croisement et des mutations. Là aussi il faut opter pour une autre méthode de sélection. [71]

5.4.1.3 La sélection par tournoi

Choisir aléatoirement deux individus et on compare leur fonction d'adaptation (combattre) et on accepte la plus adapte pour accéder à la génération intermédiaire, et on répète cette opération jusqu' à remplir la génération intermédiaire (N/2 composants). Les individus qui gagnent à chaque fois on peut les copier plusieurs fois ce qui favorisera la pérennité de leurs gènes.

5.4.2 L'opérateur de croisement

Le phénomène de croisement est une propriété naturelle de l'ADN, c'est analogiquement qu'on fait les opérations de croisement dans les AG. Il a pour but d'enrichir la diversité de la population en manipulant les composantes des individus (chromosomes). Classiquement, les croisements sont envisagés avec deux parents et génèrent deux enfants. Initialement, le croisement associé au codage par chaînes de bits ou chromosomes, est le croisement à découpage de chromosomes (slicing crossover). Pour effectuer ce type de croisement sur des chromosomes constitués de M gènes, on tire aléatoirement une position de découpage. On échange ensuite les deux sous-chaînes terminales de chacun des deux chromosomes (les parents) P_1 et P_2, ce qui produit deux nouveaux chromosomes (les enfants) C_1 et C_2 [41]. On distingue les types suivants du croisement.

5.4.2.1 Le croisement binaire: On distingue deux types de croisement binaire:

- Croisement en un point: On choisit au hasard un point de croisement, pour chaque couple. Notons que le croisement s'effectue directement au niveau binaire, et non pas au niveau des gènes. Un chromosome peut donc être coupé au milieu d'un gène.

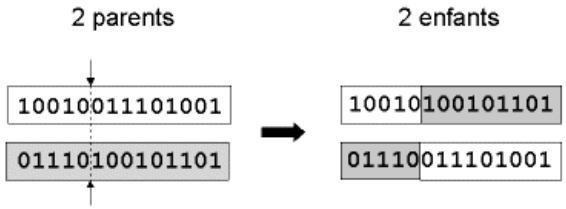


Figure 2.4 Représentation schématique du croisement en 1 point.

- Croisement en deux points: On choisit au hasard deux points de croisement. Par la suite, nous avons utilisé cet opérateur car il est généralement considéré comme plus efficace que le précédent. Néanmoins nous n'avons pas constaté de différence notable dans la convergence de l'algorithme.

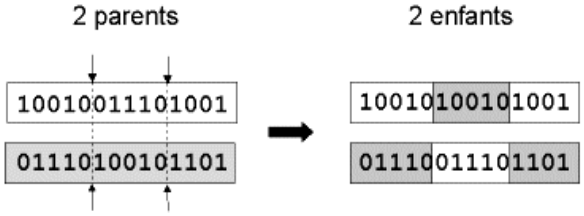


Figure 2.5 : Représentation schématique du croisement en 2 points .

Notons que d'autres formes de croisement existent, du croisement en k points jusqu'au cas limite du croisement uniforme.

5.4.2.2 Le croisement réel

Le croisement réel ne se différencie du croisement binaire que par la nature des éléments qu'il altère, ce ne sont plus des bits qui sont échangés à droite du point de croisement, mais des variables réelles.

5.4.2.3 Le croisement arithmétique

Le croisement arithmétique est propre à la représentation réelle. Il s'applique à une paire de chromosomes et se résume à une moyenne pondérée des variables des deux parents.

Soient [ai, bi, ci] et [aj, bj, cj] deux parents, et p un poids appartenant à l'intervalle [0, 1], alors les enfants sont [pai + (1-p)aj, pbi + (1-p)bj, pci + (1 - p)cj]. Si nous considérons que p est un pourcentage, et que i et j sont nos deux parents, alors l'enfant i est constitué à p% du parent i et à (100-p)% du parent j, et réciproquement pour l'enfant j.

5.4.2.4 Le croisement uniforme

Le croisement uniforme très utilisée dans le cas des problèmes modélisés par un codage binaire, La mise en œuvre de ce procédé consiste à définir de manière aléatoire un masque, c'est-à-dire une chaîne de bits de même longueur que les chromosomes des parents sur lesquels il sera appliqué. Ce masque est destiné à savoir, pour chaque locus, de quel parent le premier fils devra hériter du gène s'y trouvant, si face à un locus le masque présente un 0, le fils héritera le gène s'y trouvant du parent n° 1, s'il présente un 1 il en héritera du parent n° 2. La création du fils n° 2 se fait de manière symétrique; si pour un gène donné le masque indique que le fils n° 1 devra recevoir celui-ci du parent n° 1 alors le fils n° 2 le recevra du parent n°2, et si le fils n° 1 le reçoit du parent n° 2 alors le fils 2 le recevra du parent n° 1. [74]

5.4.3 L'opérateur de mutation

L'opérateur de mutation apporte aux algorithmes génétiques l'aléa nécessaire à une exploration efficace de l'espace. Cet opérateur nous garantit que l'algorithme génétique sera susceptible d'atteindre tous les points de l'espace d'état, sans pour autant les parcourir tous dans le processus de résolution. Ainsi, en toute rigueur, l'algorithme génétique peut converger sans croisement, et certaines implantations fonctionnent de cette manière et les résultats sont conformes aux résultats théoriques de R. Cerf. Les propriétés de convergence des algorithmes génétiques sont donc fortement dépendantes de cet opérateur. [25]

5.4.3.1 Définition

La mutation est définit comme étant l'inversion d'un bit dans un chromosome. Cela revient à modifier aléatoirement la valeur d'un paramètre du dispositif. Les mutations jouent le rôle de bruit et empêchent l'évolution de se figer. Elles permettent d'assurer une recherche aussi bien globale que locale, selon le poids et le nombre des bits mutés. De plus, elles garantissent mathématiquement que l'optimum global peut être atteint.

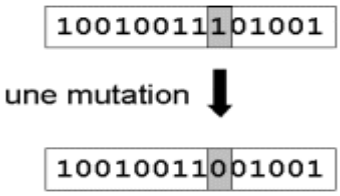


Figure 2.6 : Représentation schématique d'une mutation dans un chromosome.

D'autre part, une population trop petite peut s'homogénéiser à cause des erreurs stochastiques, les gènes favorisés par le hasard peuvent se répandre au détriment des autres. Cet autre mécanisme de l'évolution, qui existe même en l'absence de sélection, est connu sous le nom de dérive génétique. Du point de vue du dispositif, cela signifie que l'on risque alors d'aboutir à des dispositifs qui ne seront pas forcément optimaux. Les mutations permettent de contrebalancer cet effet en introduisant constamment de nouveaux gènes dans la population.

5.4.3.2 Types de mutation: De nombreuses méthodes existent. Souvent la probabilité de mutation p_m par bit et par génération est fixée entre 0,001 et 0,01. On peut prendre également $p_m=1/l$ où l est le nombre de bits composant un chromosome. Il est possible d'associer une probabilité différente de chaque gène. Et ces probabilités peuvent être fixes ou évoluer dans le temps.

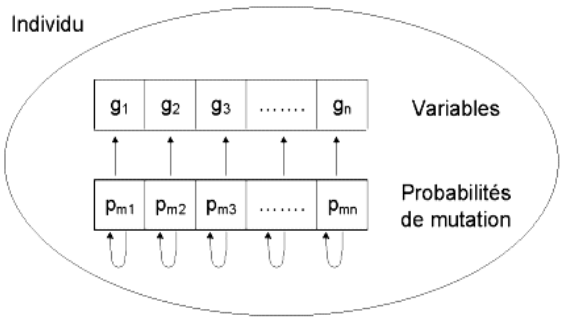


Figure 2.7 : principe de l'auto-adaptation.

Après divers essais, ils ont abouti à la méthode d'auto-adaptation des probabilités de mutation. [6] Si dans un environnement stable il est préférable d'avoir un taux de mutation faible, la survie d'une espèce dans un environnement subissant une évolution rapide nécessite un taux de mutation élevé permettant une adaptation rapide. Les taux de mutation d'une espèce dépendent donc de leur environnement. Pour prendre en compte cette formulation biologique et l'adapter avec les systèmes informatiques, ils ont introduit dans chaque individu un second chromosome (ensemble de paramètres) dont les gènes (paramètres) représentent les

probabilités de mutation de chaque gène du premier chromosome. Ce second chromosome est géré de façon identique au premier, c'est-à-dire qu'il est lui-même soumis aux opérateurs génétiques (croisement et mutation). Cela revient à fixer les probabilités assurant la modification des valeurs des paramètres du composant en fonction des valeurs d'un ensemble d'autres paramètres (les probabilités de mutation).

Lors de la genèse, les probabilités de mutation sont posées égales à 0,1 (valeur qui a paru la meilleure après plusieurs essais). Au cours du déroulement de l'algorithme, les gènes et les individus ayant des probabilités de mutation trop élevées ont tendance à disparaître. De même, les gènes ayant des probabilités de mutation trop faibles ne peuvent pas évoluer favorablement et tendent à être supplantés. Les probabilités de mutation dépendent donc du gène considéré et de la taille de la population. De plus, elles évoluent au cours du temps. Il y a donc auto-adaptation des probabilités de mutation.

5.4.3.2.1 La mutation binaire

La mutation binaire s'applique à un seul chromosome. Un bit du chromosome est tiré au hasard, sa valeur est alors inversée. Il existe une variante où plusieurs bits peuvent muter au sein d'un même chromosome. Un test sous le taux de mutation est effectué non plus pour le chromosome mais pour chacun de ses bits, en cas de succès, un nouveau bit tiré au hasard remplace l'ancien.

5.4.3.2.2 La mutation réelle

La mutation réelle ne se différencie de la mutation binaire que par la nature de l'élément qu'elle altère, ce n'est plus un bit qui est inversé, mais une variable réelle qui est de nouveau tirée au hasard sur son intervalle de définition.

5.4.3.2.3 La mutation non uniforme

La mutation non uniforme possède la particularité de retirer les éléments qu'elle altère dans un intervalle de définition variable et de plus en plus petit. Plus nous avançons dans les générations, moins la mutation n'écarte les éléments de la zone de convergence. Cette mutation adaptative offre un bon équilibre entre l'exploration du domaine de recherche et un affinement des individus. Le coefficient d'atténuation de l'intervalle est un paramètre de cet opérateur.

L’operateur de mutation garantit la diversité de la population, et permet d'éviter la dérive génétique. Il permet de limiter les risques d'une convergence prématurée. Il permet d'atteindre la propriété d'ergodicité qui garantissant que chaque point de l'espace de recherche puisse être atteint. Grâce à cette propriété on est donc sûr de pouvoir atteindre l'optimum global.

5.4.4 L’opérateur de remplacement

Cet opérateur est le plus simple, son travail consiste à réintroduire les descendants obtenus par application successive des opérateurs de sélection, de croisement et de mutation dans la population de leurs parents. On trouve essentiellement deux méthodes de remplacement différentes :

5.4.4.1 Le remplacement stationnaire

Dans ce type de remplacement, les enfants remplacent automatiquement les parents sans tenir compte de leurs performances respectives, et le nombre d'individus de la population ne varie pas tout au long du cycle d'évolution simulé, ce qui implique donc d'initialiser la population initiale avec un nombre suffisant d'individus. On distingue :

- Remplacement générationnel : Consiste de remplacer la totalité de la population P par la population P'
- un remplacement proportionnel : qui consiste à choisir une certaine proportion d'individus de P' qui remplaceront leurs parents dans P (proportion égale à 100 % dans le cas du emplacement générationnel). Ce type de remplacement engendre une population ayant une grande variation et de se fait favorise la dérive génétique.

5.4.4.2 Le remplacement élitiste

Dans le remplacement élitiste, on garde au moins l'individu possédant les meilleures performances d'une génération à la suivante. En général, on peut partir du principe qu’un nouvel individu (enfant) prend place au sein de la population que s’il remplit le critère d’être plus performant que le moins performant des individus de la population précédente. Donc les enfants d'une génération ne remplaceront pas nécessairement leurs parents comme dans le remplacement stationnaire et par la même, la taille de la population n'est pas figée au cours du temps.

Le remplacement élitiste améliore les performances des algorithmes évolutionnaire dans certains cas. Mais présente aussi un désavantage en augmentant le taux de convergence prématuré. Néanmoins, des implémentations plus fines procèdent de manière différente. Dans ce cas là, le taux de remplacement n'est pas de 100 %, la taille de la population augmente

donc au cours des générations successives, on dit qu'il y a *overcrowding*. Il faut donc trouver un moyen pour sélectionner les parents qui seront supprimés, qui vont mourir. De Jong a proposé la solution suivante : imaginons qu'on veuille remplacer 30 % des parents, soit np le nombre de parents correspondants à ce pourcentage, on remplacera les np parents les plus proches de leurs descendants de P' [29]. Cette méthode permet donc premièrement de maintenir la diversité et deuxièmement d'améliorer la fitness globale de la population.

5.5 Le critère d'arrêt

Différentes politiques sont possibles pour décider quand arrêter l'algorithme car l'algorithme génétique, comme tous les algorithmes évolutionnaires, sont dans l'absolu sans fin. La politique la plus courante et la plus simple est sans conteste d'effectuer un nombre prédéfini d'itérations mais d'autres sont possibles. On peut distinguer trois grandes familles de critères d'arrêt :

- Le temps ou le nombre d'itérations voulu est atteint: il représente la majeure partie des critères d'arrêt employés. En effet, ils sont très faciles à mettre en œuvre. Le temps est lié au nombre d'itérations suivant la taille des données.

- La fonction d'évaluation est constante depuis quelque temps. Cela permet de caractériser un algorithme qui n'arrive pas à trouver de meilleures solutions et qui est dans une configuration de minimums locaux.

- La population est dominée par quelques individus. Ce critère est en général asse complexe et peu faible. Il est très rarement employé. [65]

6. Les paramètres d'un algorithme génétique

Aujourd'hui il y a pas une théorie ou bien un standard qui décrire les paramètres des AG pour n'importe quel problème. Mais on peut trouve des résultats des tests très utilisées dans la communauté de AG.

Les opérateurs de l'algorithme génétique sont guidés par un certain nombre de paramètres fixés à l'avance. La valeur de ces paramètres influence la réussite ou non d'un algorithme génétique. Ces paramètres sont les suivants :

1. La fonction d'évaluation qui détermine la probabilité de sélection et de reproduction d'un individu.
2. La taille de la population: qui détermine le nombre de gènes dont l'AG dispose pour produire de bonnes solutions. Une population trop petite évoluera probablement vers un optimum local peu intéressant. Une population trop grande sera inutile car le temps de

convergence sera excessif. La taille de la population doit être choisie de façon à réaliser un bon compromis entre temps de calcul et qualité du résultat. La bonne taille de la population est approximativement 30-50. Quelques recherches ont montré que la taille de la population dépend du type de codage.
Les résultats expérimentaux de De Jong indiquent que la taille idéale d'une population est de 50 à 100 individus. [29]
3. La probabilité de mutation que subit chaque allèle lors de la reproduction, le taux de la mutation devrait être très bas. Les meilleurs taux rapportés sont approximativement 0.5%-1%.
4. La probabilité de crossover qui détermine la fréquence à laquelle les hybridations entre individus vont avoir lieu, généralement, elle doit être élevée entre 80%-95%. (Cependant quelques résultats montrent que le taux 60% est le meilleur pour quelques problèmes). Les résultats expérimentaux de De Jong indiquent le meilleur taux de croisement en un point est ~0.6 par paire de parents. [29]

Ces deux derniers paramètres sont corrélés. Alors que la mutation permet l'apparition de nouveaux gènes par la modification d'un allèle, le crossover diffuse les gènes existants lors du renouvellement de la population. Cette balance entre mutation et crossover doit être soigneusement respectée car un taux de mutation trop élevé entraîne la destruction de gènes avant qu'ils n'aient eu la chance d'être assemblés par crossover afin de former des structures valables. De l'autre côté, si le taux de crossover est excessif, la population sera uniformisée trop rapidement et la population convergera alors prématurément vers un type d'individu probablement sous-optimal.

Plusieurs chercheurs trouvent qu'on ne peut pas formuler des principes généraux sur les paramètres d'un AG à priori, et que l'approche la plus prometteuse est d'avoir des valeurs de paramètres qui s'adapte en temps réel.

Par sa difficulté, le réglage des paramètres d'un algorithme génétique, est souvent assimilé à de la magie noire (la littérature des AG utilise le terme de Black Art). Deux types d'approches sont possibles afin de tenter de résoudre ce problème:

- Mettre ces paramètres sous le contrôle d'un super-système adaptatif, un autre AG. Ce type de tentative pose un problème supplémentaire : le paramétrage du super-système.
- Coder dans le génome lui même des coefficients qui vont paramétrer l'activité des opérateurs génétiques. On peut introduire un facteur de lien (une sorte de colle) entre

les gènes d'un individu afin de produire, par sélection naturelle, de robustes associations de gènes. Il est aussi possible d'associer aux génomes une carte des points de crossover possibles et d'utiliser des opérateurs de crossover spéciaux. De la même manière, la mutation peut être contrôlée par l'association, à chaque allèle, d'un coefficient de mutation.

7. Améliorations classiques

Les processus de sélection présentés sont très sensibles aux écarts de fitness et dans certains cas, un très bon individu risque d'être reproduit trop souvent et peut même provoquer l'élimination complète de ses congénères. On obtient alors une population homogène contenant un seul type d'individu. Pour éviter ce comportement très néfaste et pénalisant, il a été développé d'autres méthodes de sélection (*scaling, sharing*) qui empêchent les individus forts d'éliminer totalement les faibles. Nous décrirons ici les mécanismes de *scaling* et *sharing*.

7.1 Le scaling

Le scaling ou mise à l'échelle, modifie les fitness afin de réduire ou d'amplifier artificiellement les écarts entre les individus. Le processus de sélection n'opère plus sur la fitness réelle mais sur son image après scaling. On souhaite ainsi aplatir ou dilater la fonction d'évaluation. La perte de précision qui s'ensuit est au prix d'une meilleure convergence. Parmi les fonctions de scaling, on peut envisager le scaling linéaire et le scaling exponentiel. Soit fr la fitness avant scaling et fs la fitness modifiée par le scaling.

7.1.1 Scaling linéaire

La fonction de *scaling* est définie de la façon suivante: $fs = a.fr + b$ 2.1

Dans ce type, la fonction de scaling est une fonction affine, dont le coefficient directeur est généralement inférieur à un, ce qui permet de réduire les écarts de fitness et donc de favoriser l'exploration de l'espace. Néanmoins, ce scaling est statique par rapport au numéro de génération et devient donc gênant en fin de convergence lorsqu'on désire favoriser les dominants pour accélérer le processus de recherche. Le scaling exponentiel permet d’éviter ce phénomène.

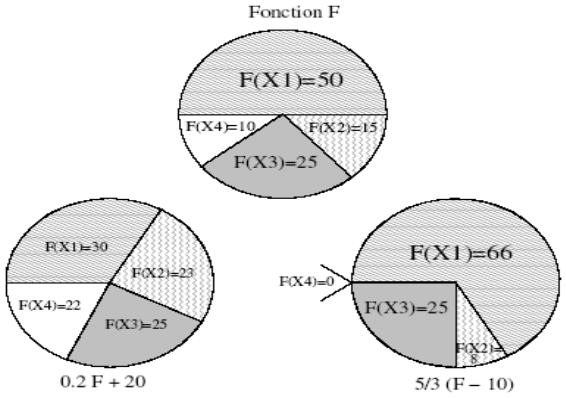


Figure. 2.8 Influence de la fonction de *scaling* sur la pression de sélection.

7.1.2 Scaling exponentiel

Le scaling exponentiel est défini de la façon suivante: $fs = fr^{k(n)}$ 2.2

Où n représente le numéro de la génération courante. Il est nécessaire d'introduire un coefficient exposant variable, car les phénomènes résultant du scaling varient avec sa valeur.

1. Pour k proche de zéro, on réduit fortement les écarts de fitness. Aucun individu n'est vraiment favorisé et l'algorithme génétique se comporte comme un algorithme de recherche aléatoire et permet d'explorer l'espace.
2. Pour k proche de 1: le *scaling* est presque inopérant. Les individus sont triés juste selon leur adaptabilité propre, quasiment non modifiée. Le scaling ne favorise aucun individu, mais n'en défavorise également aucun.
3. Pour k > 1 : les écarts sont exagérés et seuls les bons individus sont sélectionnés ce qui produit l'émergence des modes. [62]

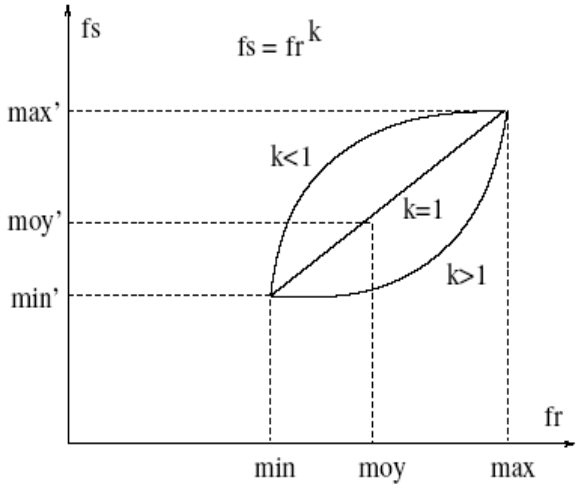


Figure. 2.9 Fonction de *scaling* exponentielle

Dans la pratique, on fait donc varier k des faibles valeurs vers des valeurs fortes au cours des générations, Pour cela on peut utiliser la formule suivante : $k = (\tan[(\frac{n}{N+1})\frac{\pi}{2}])^p$ 2.3.

n étant la génération courante, N le nombre total de générations prévues, p un paramètre à choisir. L'évolution de k en fonction de la génération n est donnée par la figure (2.9). Ce type de *scaling* donne de meilleurs résultats que le scaling linéaire. [62]

7.2 Le sharing (partage)

Le sharing consiste à modifier la fitness utilisée par le processus de sélection. Pour éviter le rassemblement des individus autour d'un mode dominant, il faut pénaliser la fitness en fonction du taux d'agrégation de la population dans le voisinage d'un individu plus les individus sont regroupés, plus leur fitness est faible, et des individus proches les uns des autres doivent partager leur fitness. Dans la pratique, pour estimer ce taux d'agrégation, on ouvre un domaine autour d'un individu, puis on calcule les distances entre les individus contenus dans ce voisinage et ce dernier. Il faut donc pouvoir définir une distance représentative entre deux individus.

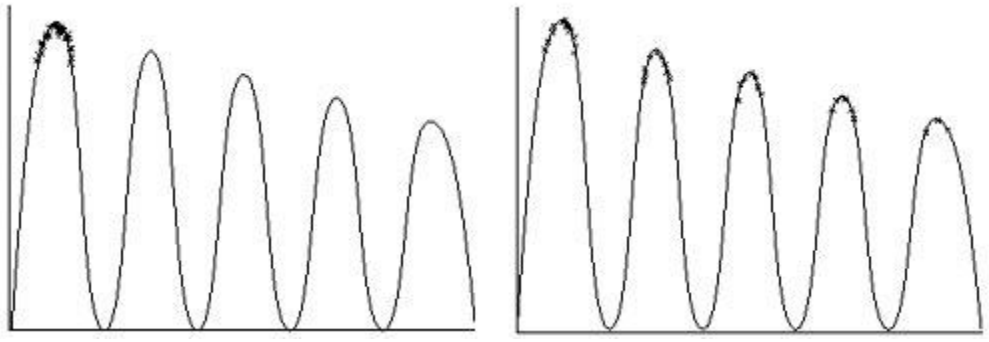


Figure 2.11 a. sans sharing : concentration des individus sur un seul mode b. avec sharing : répartition des individus sur l'ensemble des modes

Cette méthode donne de bons résultats, mais au prix de n^2 calculs supplémentaires par génération, où n représente la taille de la population. [62]

8. Association des algorithmes génétiques avec des méthodes locales

Il est bien connu que l'évolution de tout algorithme génétique est fortement influencée par la structure de l'espace de recherche. En fait, pour concevoir une heuristique efficace, il est essentiel d'exploiter (implicitement ou explicitement) les propriétés de l'espace de recherche. Le risque de tomber dans des optimums locaux reste toujours, mais pour de nombreux algorithmes, la situation la plus difficile n'est pas l'optimum local classique, mais le piège (trap).

Une fois piégée dans une telle structure, une recherche locale classique va boucler en permanence entre les optimums locaux à l'intérieur du puits bien qu'elle puisse avoir les capacités d'échapper à tout optimum local individuel.

8.1 La recherche locale typique et la vision globale

Particulièrement dans le contexte des heuristiques à base de recherche locale, un risque important est le fait que le processus de recherche pourrait être toujours attiré par les mêmes optima locaux (des forts attracteurs). Tandis qu'il y a de nombreuses méthodes bien étudiées pour aider une recherche à échapper à tout optimum local individuel (la recherche Tabou [39,40] a été conçue pour cela), il semble plus difficile de l'empêcher de boucler entre seulement quelques optimums locaux. Rappelons qu'une recherche locale typique se déplace d'une configuration à l'autre sans enregistrer beaucoup de données sur les régions visitées. Habituellement, au moment d'une itération donnée, il n'y a aucune information si la recherche est en train d'explorer une région complètement inconnue (jamais visitée) ou une région déjà explorée (avec des configurations visitées dans la proximité).
En effet, un algorithme typique de recherche locale n'a pas une vue d'ensemble sur son propre chemin d'exploration à travers l'espace, il ne prend pas en compte les relations entre des configurations visitées à des moments différents dans une longue recherche.

Étant donné un processus de recherche traversant l'espace des solutions, quelques questions importantes pourraient être posées :

- A quoi ressemble son chemin d'exploration ?
- Quelles régions seront le plus souvent explorées ?
- Le processus de recherche, explore-t-il beaucoup plus que quelques régions ?
- Quelle est la distribution spatiale des meilleures configurations visitées ?
- Ces meilleures configurations sont-elles aléatoirement dispersées ?
- Le processus de recherche, peut-il être guidé vers un optimum global ? [27]

8.2 Les algorithmes génétiques et les méthodes locales

La grande force des algorithmes génétiques est leur capacité à trouver la zone de l'espace des solutions contenant l'optimum de la fonction. En revanche, ils sont inefficaces lorsqu'il s'agit de trouver la valeur exacte de l'optimum dans cette zone. Or, c'est précisément ce que les algorithmes locaux d'optimisation réalisent le mieux. Il est donc naturel de penser à associer un algorithme local à l'algorithme génétique de façon à trouver la valeur exacte de l'optimum. On peut aisément le faire en appliquant à la fin de l'algorithme génétique un algorithme local sur le meilleur élément trouvé. Cette technique est d'autant plus efficace que l'on utilise simultanément du clustering, et que l'algorithme local est appliqué à chaque meilleur élément de chaque cluster. En effet, on constate souvent que le meilleur élément trouvé par l'algorithme génétique ne reste pas le meilleur élément après amélioration par l'algorithme local de tous les meilleurs éléments de clusters.

Une autre technique consiste à utiliser un algorithme local associé à l'algorithme génétique pour calculer la fitness d'un élément. On peut, par exemple dans un espace fortement combinatoire, rechercher avec l'algorithme génétique les zones intéressantes de l'espace en générant les éléments, l'adaptation de chaque élément étant calculée par un programme d'optimisation local. En fait, l'association algorithme génétique et méthodes locales est une quasi-nécessité. Les deux méthodes sont complémentaires et ont des champs d'application différents. L'algorithme génétique permet de faire disparaitre la combinatoire du problème, laissant alors le champ libre aux méthodes locales dans chacune des zones connexes qu'il pense susceptible de contenir l'optimum global. [62]

9. L’application des algorithmes génétiques dans le domaine de la RAP

La moyen privilégié de communication de l’homme reste toujours la parole avec la révolution des machines et les ordinateurs ce qui encouragés les chercheurs à l’usage de la parole comme un moyen de communication homme machine.

La complexité du signal de la parole poussé Les chercheurs à explorent de nouvelles voies de recherche pour une meilleure compréhension du problème de la RAPs. Une multitude d’algorithmes ont ainsi été élaborés afin d’améliorer la performance et la robustesse des SRAPs, Les algorithmes génétiques constituent une approche intéressante pour les SRAPs.

L'efficacité de AG dans l'optimisation des fonctions complexes et ses différentes domaines d'application donnent un motive pour l'utilisation des AG dans la RAP. Nous cherchons à savoir si ce type d'algorithmes peut améliorer les performances des SRAPs ou donnent des résultats comparables ou mieux que les approches classiques des systèmes de reconnaissance de la parole arabe.

Nous trouvons dans les travaux déjà réalisés : un modèle d'identification acoustique des voyelles de l'arabe standard qui utilisé les algorithmes génétiques. [1] Aussi les algorithmes génétiques appliqués à la reconnaissance automatique *of the Arabic Stop Sounds* [2]. Dans le domaine de vérification des locuteurs pour l'identification d'une personne à partir de sa voix nous trouvons une combinaison de codeurs par algorithme génétique [24]. Il existe d'autre type d'application des AG dans le domaine de la reconnaissance de la parole qui consiste à optimisé des variables pour un réseau de neurone (système hybride AG /RN) [4].

10. Conclusion

Les algorithmes évolutionnaires présentent des caractéristiques intéressantes les rendant populaires, notamment grâce à leur facilité d'emploi. Leurs points forts sont une certaines robustesse due à l'utilisation de l'aléatoire, une facilité de mise en œuvre. Le principe de ces derniers se base sur une description fidèle d'une évaluation naturelle, et assure une recherche efficace dans le monde des solutions d'un problème donné.

L'algorithme génétique n'exige aucune connaissance de la manière dont on résoudre le problème, il est seulement nécessaire de pouvoir estimer la qualité d'une solution potentielle. Cette approche présente l'immense avantage pratique de fournir des solutions pas trop éloignées de l'optimal, même si l'on ne connaît pas de résolution algorithmique. Donc l'algorithme génétique se caractérise par sa fonction d'évaluation (d'adaptation), qui donnera à chaque solution possible une valeur reflétant sa qualité pour résoudre le problème posé, plus la valeur de la fonction est élevée, meilleure est la solution.

Le chapitre suivant est consacré à la langue arabe avec les particularités phonologiques et les traits caractéristiques de cette langue et leurs problèmes en traitement automatique.

Chapitre 3

La Langue arabe

1. Introduction

L'arabe est une langue sémitique. Elle doit sa fortune à l'expansion de l'islam, qui s'est étendu en l'espace de quelques siècles, de l'Afrique du nord à l'Espagne, puis au Proche-Orient et en Asie.

Dans ce chapitre on s'intéresse à la langue arabe classique ou bien officielle. Il s'agit d'une forme linguistique ancienne dont la grammaire a été fixée entre le 8ème et le 10ème siècle. L'arabe classique dit aussi arabe « coranique » n'est plus que la langue du patrimoine culturel passé avec ses œuvres classiques et son livre sacré le Coran. L'arabe classique est appris dans les établissements d'enseignement à travers la littérature arabe classique et les cours de théologie.

La première grammaire arabe, rédigée par Sibawahi (8ème siècle) dans « Al-Kitab » constitue le premier travail de standardisation de la langue arabe. Il fut conduit pour répondre aux inquiétudes des religieux, qui à l'époque des premières conquêtes musulmanes, voulaient éviter tout risque de corruption de la parole divine pouvant résulter de la manipulation de la langue par les nouveaux convertis à l'Islam d'origine non arabophone. L'objectif de la standardisation de la langue arabe est donc, à l'origine d'assurer la pureté linguistique du texte sacré. Néanmoins, l'un des nombreux atouts d'El-Kitab, est d'une part la description articulatoire fine du système phonologique de l'arabe littéraire classique, d'autre part la description de certaines caractéristiques linguistiques des dialectes arabiques de l'époque. On peut considérer que ce travail fondateur a donné un standard de la langue arabe pour les différents travaux de la reconnaissance écrite ou bien orale.

2. L'arabe Standard contemporain ou moderne

À la coté de l'arabe classique on trouve l'arabe standard qui est un peut différente de l'arabe classique. Elle est utilisée dans les médias (langage de presse) dans la littérature, dans les conférences et les discours politiques. La langue arabe standard soutenue par le pouvoir politique, permet la fixation d'une norme linguistique et l'existence d'une forme écrite stabilisée, diffusée par le biais d'un enseignement formel et par les médias. L'arabe standard conserve ainsi le monopole dans toute la vie officielle, administrative et universitaire. C'est aussi par le biais de cette langue 'supra-nationale', que deux locuteurs arabophones cultivés d'origines dialectales différentes sont susceptibles de se comprendre.

Au niveau linguistique, l'arabe standard contemporain ne peut être distingué de l'arabe classique dont il a conservé presque intégralement la morphologie et la syntaxe, seuls

quelques procédés syntaxiques anciens ont évolué vers de nouvelles formes. Le lexique fortement contrôlé et régi par des contraintes formelles strictes s'organise autour d'un nombre fini de racines et de schèmes.

L'intégration de nouveaux mots, généralement empruntés aux langues européennes comme le français, l'italien ou l'anglais pour traduire les concepts issus du développement technologique du 19° siècle, se fait toujours en fonction des règles imposées par le système arabe. Le plan de la prononciation est théoriquement considéré comme phonologique et tente de suivre les normes classiques. [43]

3. L'alphabet arabe

L'alphabet arabe comporte 28 consonnes et 6 voyelles de l'arabe standard (3 longues et 3 courtes) et quelques autres réalisations vocaliques. Nous pouvons classer les consonnes selon plusieurs critères : des consonnes articulées avec une vibration des cordes vocales et des consonnes qui n'engendrent pas une vibration des cordes vocales, le franchissement de l'air à travers le conduit vocal donne naissance à d'autres variétés de sons.

Les 28 consonnes arabes ont été divisées en deux groupes :

• 14 consonnes solaires qui n'assimilent pas le « ل » de l'article : ن , ل , ظ , ط , ض , ص , ش , س , ز , ر , ذ , د , ث , ت .

• 14 consonnes lunaires qui assimilent le « ل » de l'article : ي , و , م , ه , آ , ق , ف , غ , ع , خ , ح , ج , ب , أ . [68]

4. La prononciation

Il est question pour l'instant de la prononciation des phonèmes symbolisés par les lettres seules. Il ne s'agit pas de la prononciation des mots ni des constructions plus complexes. Certains sons arabes n'existent pas dans les langues latines, tout comme certains sons d'origines latines n'existent pas en arabe.

L'arabe qui est fréquemment qualifiée de langue chantante, a une gamme de sonorités plus riche que l'alphabet latin. Certaines lettres arabes se prononcent comme les lettres latines (à la nuance prêt de l'accent, mais qui est d'ailleurs lui-même variable au sein du monde arabe). Le tableau (3.1) donne un exemple de certaines lettres et leurs prononciations.

lettre	nom francais	T1	T2	son	nom arabe
ة	ta marbuTah	A	Ah	a expiré	التاء المربوطة
ى	Alif maqSurah	A	A/à	a long	الألف المقصورة
ع	ayn	/'	3/'	coup de glotte	عين

Table3.1 Un exemple de certaines prononciations des lettres.

La colonne « T1 » donne la première forme de translittération la plus simple qui est une translittération est une écriture approximative de l'arabe avec les caractères latins. La colonne « T2 » donne une deuxième forme de translittération plus exacte, mais moins évidente pour ceux/celles qui ne la connaissent pas. La colonne « Son » est une indication sur la manière de prononcer la lettre en arabe. Il s'agit de la prononciation du phonème de la lettre, et non pas de la prononciation du nom de la lettre. La prononciation dépend seulement quelquefois du contexte alors que l'écriture d'une lettre dépend toujours du contexte. Le tableau (3.2) représente l'ensemble des caractères de l'arabe avec pour chaque graphème, sa transcription phonétique.

lettre	nom	phoneme	lettre	nom	phoneme
ا	alif	a:	ط	taa	ŧ
ب	baa	b	ظ	thad	z
ت	taa	t	ع	ayn	/ /
ث	thaa	θ	غ	ghayn	/ /
ج	gym	/ /	ك	kaaf	k
ح	haa	h	ق	qaaf	q
خ	khaa	x	ف	faa	f
د	daal	d	ل	laam	l
ذ	dhaal	ð	ن	nuwn	n
ز	zayn	z	م	mym	m
ر	raa	r	ه	haa	h
س	syn	s	و	waaw	u:
ش	shyn	/ /	ي	yaa	i:
ص	saad	Ṣ	ء	hamza	/ /
ض	daad	đ	/	/	/

Table 3.2 : Correspondance graphème phonème de la langue arabe suivant l'alphabet phonétique internationale IPA.

5. Le lexique arabe

Le lexique arabe comprend trois catégories de mots. Verbes, noms (substantifs et adjectifs deux catégories qu'il est difficile de distinguer) et particules (recouvrant adverbes,

conjonctions et prépositions). Les mots des deux premières catégories sont dérivés à partir d'une racine. Une famille de mot peut être générée à partir d'une seule racine à l'aide de différents schèmes. La racine est un squelette de trois consonnes radicales le plus souvent, quatre dans 1 à 2 % des cas. À partir d'une racine, passée dans différents schèmes, une famille de mots peut être engendrée autour d'un même concept sémantique comme celui d'*écriture*. Ainsi, si cette racine (كتب) est passée dans le schème du participe actif, le mot (كاتب)(/ka:tibu/, « écrivain ») est formé, c'est le fait le plus caractéristique de la morphologie arabe et plus généralement sémitique.

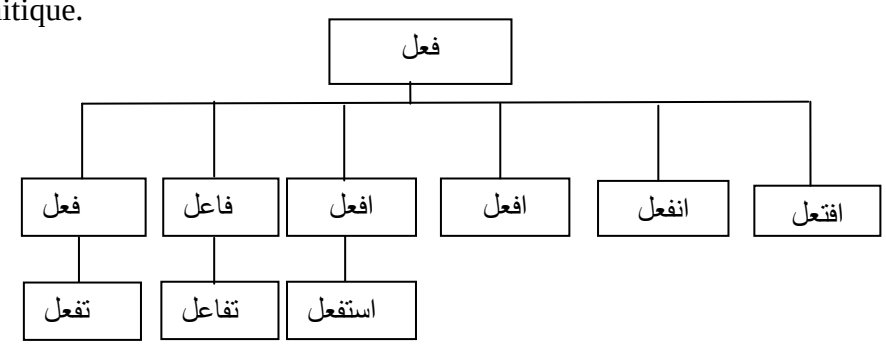


Figure3.1 : Les dérivés verbaux.

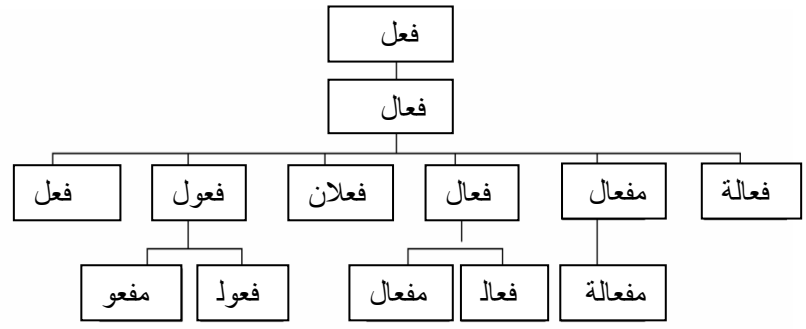


Figure3.2 : les dérivés nominaux.

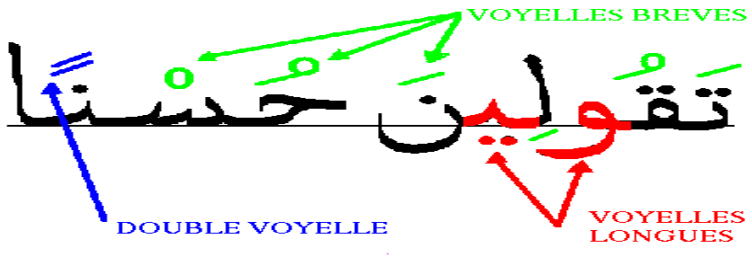
5.1 L'écriture arabe

La langue arabe est représentée par une écriture curviligne composée de consonnes liées entre elles et s'écrivant de droite à gauche contrairement aux écritures occidentales qui s'écrivent de gauche à droite. Certaine de ces consonnes changent de forme suivant la place qu'elles occupent dans le mot.

à la fin d'une lettre non joignable	à la fin	au milieu	au début
ع	ـع	ـعـ	عـ

Table 3.3 Exemple de l'écriture d'une lettre suivant la place qu'elles occupent dans le mot.

L’arabe est une écriture consonantique. Cela se traduit par le fait que toutes ces lettres sont des consonnes. En ce qui concerne les voyelles de l’arabe, on distingue deux types de voyelles. Les voyelles longues et les voyelles brèves ou courtes. La durée d’une voyelle longue est environ double de celle d’une voyelle courte. Ces voyelles sont caractérisées par la vibration des cordes vocales. Les voyelles sont ajoutées au dessous au dessous des lettres.



TA Q OUOUL IINA~H OU S’NAN(tu (au féminin) dis de bien)

Figure 3.3 exemple de l’écriture des voyelles courtes

5.2 Les voyelles

Les voyelles simples sont des petits son que l'on ajoute aux consonnes ou à certaines voyelles pouvant jouer le rôle d'une consonne (Ya/ ي et Waw/و). On distingue deux types de voyelles les voyelles brèves et les voyelles longues. Ce qui nous intéresse dans les voyelles, est le caractère flexionnel qui donne aux mots arabes. Les terminaisons permettent de distinguer le mode des verbes, la fonction des noms… Les signes suivants (déclinaisons ou désinences) sont des indices très importants pour nos règles.

5.2.1 Les voyelles brèves (courtes)

Les voyelles brèves sont facultatives. Elles sont quasiment inexistantes dans les textes contemporains courants. Le lecteur, grâce à son expérience et à ses connaissances lexicales et grammaticales, doit pouvoir les ajouter, à la vitesse de la lecture, comme un calque mental qu'il superpose sur chaque mot avant de le prononcer. Il y en a quatre voyelles courtes classées d’après la position des organes de phonation qui concourent à leur émission, lèvres et langue.

- La première voyelle se prononce en contractant la langue au fond de la bouche et en avançant les lèvres qui s’arrondissent jusqu’à presque se joindre. Elle est représentée par le signe « ُ » placé au-dessus de la consonne, appelé ***damma***. Elle est prononcée comme « ***ou*** ».

- La deuxième voyelle se prononce en ouvrant largement la bouche et en conservant la langue dans une position horizontale. Elle est représentée par le signe « ◌َ » placé au-dessus de la consonne, appelé ***fatha***. Elle est prononcée comme « ***a*** ».
- La troisième voyelle se prononce en portant le devant de la langue en avant et en l'étale largement tandis que l'arrière frôle presque le palais et que les commissures des lèvres s'étirent. Elle est représentée par le signe « ◌ِ » placé au-dessous de la consonne, appelé ***kasra***. Elle est prononcée comme « ***i*** ».
- Soukoun: C'est le signe « ° » placé au-dessus de la consonne pour indiquer que cette consonne n'est pas menue de voyelle : [من ***quiconque***, عن ***loin de***, سل ***interroge***].

5.2.2 Les voyelles longues

Il y en a trois voyelles longues: ا *A*, و *OU* et ي *I*.

مَاذَا	M A DH A QUOI
نُورُ	N OU R OU LUMIERE
عِيدِي	'IDI MA FETE

Table 3.4 : les voyelles longues

Ces trois lettres sont à la fois des voyelles et des consonnes (semi-voyelles). Lorsqu'elle ne joue pas le rôle de voyelles, elle se comporte comme le restant des consonnes, et peut donc porter des voyelles brèves.

5.2.3 Les semi- voyelles : Dans l'alphabet arabe, il existe 2 semi -voyelles:

- [و w, ولد – ***waled***].
- [ي y, بيت – *bayt*].

Généralement l'écriture arabe a noté que les consonnes et semi-consonnes sauf le coran et quelques rares textes didactiques on trouve les voyelles brèves. Les voyelles permettent de distinguer la catégorie grammaticale du mot sa fonction, des indications de temps, d'aspect, de mode, de genre et de nombre. Seule une connaissance apprise ou intuitive, de la grammaire permet au lecteur arabophone de vocaliser son texte correctement et par conséquent de le comprendre.

Une analyse morphologique qui va assigner à chaque mot non-voyellé l'ensemble des mots voyelles correspondants. cette analyse recourt à un lexique supposé exhaustive qui

intègre toutes les formes canoniques et fléchies des mots. La table (3.5) donne les résultats de l'analyse morphologiques du mot كتب/ktb.

mots sans voyelles	1 ère interprétation		2 ème interprétation		3 ème interprétation	
كتب	كَتَبَ	il a écrit	كُتِبَ	il a été écrit	كُتُبُ	des livers

Table 3.5 Interprétation du mot كتب sans voyelles.

6. Les syllabes

Depuis longtemps, la syllabe est implicitement tenue comme l'élément clé de l'architecture prosodique. La syllabe est une unité privilégiée pour décrire le rythme (même si le rôle de cette unité est sujet de débat). Elle est impliquée directement dans les mécanismes rythmiques des langues dans certaines théories phonologiques. Néanmoins, le rythme s'inscrivant dans l'ordre de structuration temporelle de la parole. [9]

Les syllabes dans la langue arabe sont basées sur des éléments contrastés situés à l'intérieur de la frontière de la syllabe. Chaque syllabe a une partie principale connue par le noyau de la syllabe qui est la voyelle. Les éléments restants sont appelés les facteurs marginaux et sont représentés par les consonnes. Une syllabe commence toujours par une seule consonne et se termine soit par une consonne soit par deux consonnes soit par aucune consonne. [78]

La langue arabe comporte cinq types de syellabes classées selon les trais ouvert/fermé et court/long (c=consonne, v=voyelle) une syllabe est dite ouverte si elle se termine par une voyelle, une syllabe est dite fermée si elle se termine par une consonne. elle est dité courte si elle est ouverte et intègre une voyelle brève. Elle est dite longue si elle se termine par une voyelle longue ou par une consonne. La terminologie syllabe lourde (cvv,cvc) est syllabe surlourde (cvvc,cvcc) est également utilsée dans la littérature. La syllabe cvvcc n'est pas attestée en arabe standard. [9,10]

Syllabe	Courte	Longue
Ouverte	CV	CVV
Fermée	N'existe pas	CVC, CVVC, CVCC

Table 3.6 Système syllabique de la langue arabe

Le système syllabique de l'arabe a les caractéristiques suivantes :

1- Toutes les syllabes commencent par une consonne suivie d'une voyelle.
2- Les syllabes comportent une seule voyelle.
3- La syllabe CV peut se trouver en début en milieu en fin de mot exemple : درب /daraba /cvcvcv(il a frappé).
4- La syllabe CVV peut se trouver en début en milieu en fin de mot ou isolée. Comme l'exemple (كان Kana/CVVCV il était. باقي BaAqI/CVVCVV le reste. في/FI/CVV dans).
5- La syllabe CVC peut se trouver en début en milieu en fin de mot ou isolée. Comme l'exemple de (بعد baɛda /CCCV après. مد mud/CVC donne).
6- La syllabe CVVC peut se trouver en début en milieu en fin de mot ou isolée. Exemple du mot (كبير/KaBIr/CVCVVC, grand).
7- La syllabe CVCC se trouve uniquement en fin de mot ou isolée. Exemple du mot (مهر/mahr/CVCC perle).

7. Particularités phonologiques et les traits caractéristiques de la langue Arabe

L'originalité de la phonétique arabe se fonde, pour une grande partie, sur la pertinence de la durée dans le système vocalique et sur la présence de consonnes emphatiques et des consonnes géminées. Les caractéristiques phonologiques de l'arabe sont: l'emphase, la gémination, le madd et la double voyelle.

7.1 Problème de la durée

Le paramètre durée est très important dans la langue arabe. Il caractérise non seulement les voyelles, mais également les consonnes géminées. Concernant ce trait, un double problème se pose en reconnaissance automatique de l'arabe. Il faut déceler les phonèmes allongés tout en s'assurant que ce prolongement est pertinent c'est à dire en le distinguant des allongements dus au débit d'élocution, à un accent particulier du locuteur. Par exemple, les deux mots: 'جمل' (chameau) et 'جمال' (beauté) ne diffèrent que par l'allongement de la voyelle finale. On exige du système de reconnaissance de déceler les 2

voyelles sans altérer la propriété temporelle, et en s'assurant que ce prolongement est pertinent. Un alignement temporel au contraire pénaliserait cette détection. [web4]

7.2 La gémination

La gémination est le phénomène de renforcement de l'articulation consonantique qui en prolonge la durée environ de moitié et en augmente l'intensité. Elle est définie comme étant la succession de deux consonnes identique prononcées consécutivement, elle joue un rôle important dans la définition et le sens de certains mots. Ce phénomène est parfois appelé *redoublement*, bien qu'il n'y ait pas véritablement répétition de la consonne. En arabe standard, la géminée est indiquée par le signe diacritique spécial appelé *chaddah*. [53]

La Chedda s'écrit sur une consonne ou sur une voyelle pouvant jouer le rôle d'une consonne, comme و et ي, et jamais sur ا. Son effet est de doubler la consonne sur laquelle elle est posée. Dans certains cas, on ne double pas véritablement la lettre, mais on insiste plutôt sur celle-ci. La Chedda est toujours accompagnée d'une voyelle brève. À titre d'exemple, si nous prenons un Noun sur lequel on ajoute une Chedda accompagnée d'une Fatha, cela se prononcera comme si on avait (نَ نْ « nna ») doublement du Noun, prononciation du premier Noun sans la voyelle, puis prononciation une seconde fois du Noun, avec la voyelle.

Un consonne géminée est son unique pour lequel les organes de phonation ne changent pas de position, (les lèvres ne se referment pas après le premier /b /dans le mot kabbara/) d'où la transcription (/ kab :ara /) qui est plus appropriée. Dans beaucoup de langues ce phénomène permet de mettre en relief un mot dans son contexte alors qu'il s'avère être un élément distinctif sur les plans morpho-sémantiques en longue arabe. [9]

La durée de la consonne simple est sensiblement différente de celle géminée ainsi que la durée de la voyelle précédant la consonne. Il faut différencier entre une gémination et une consonne simple suivie d'une voyelle longue. La durée de la voyelle précédant une consonne géminée diminue par rapport à une consonne simple, cette diminution peut être expliquée par une tendance du locuteur à insister sur la gémination plutôt que sur la voyelle qui la précède.

Le tableau (3.7) illustre la différence en ms (milli seconde) entre les consones simples et géminées. [52]

Consonne	Simple	Géminé	Consonne	Simple	Géminé
ب	58	99	ت	62	105
ث	65	123	ج	70	121
خ	65	110	ح	63	102
د	69	145	د	68	139
ر	64	95	ه	59	116
س	70	130	ش	72	131
ط	69	134	ع	61	113
ف	62	129	ق	52	101
ك	67	129	ل	68	141
م	62	116	ن	60	124

Table 3.7 la différence en *ms* entre les consonnes simples et géminées

7.3 Les voyelles brèves doubles (tanwine)

Les trois signes qui représentent les voyelles sont quelques fois redoublés à la fin des noms et les voyelles finales se lisent alors comme si elles étaient suivies du son « ***n*** ». Nous appelons ce phénomène *tanwin.*

Voyelles	Nom francais	T1	T2	Son	Nom arabe
ـً	Fathatan	Ane	An	[كتابَ – *kita**ban***].	الفتحةً
ـٍ	Kasratan	Ine	In	[كتاب – *kita**bin***].	الكسرةٍ
ـٌ	Dammatan	Oune	Un	[كتابٌ – *kitab**un***].	الضمةٌ

Table 3.8 Les différentes formes existantes de tanwine.

7.4 Le madd

Ce phénomène concerne l'allongement des voyelles. Il est provoqué par la présence d'une voyelle longue (**و/U, ا/A ou ي /I)**.

La lecture de textes arabes est régie par des règles phonologiques qui ont trait à la contraction des sons leur élision et à l'assimilation homo-organique des nasales. Certaines de ces règles sont obligatoires d'autres facultatives ou réservées à certains types de textes, comme le coran. Nous présentons ci-dessous des définitions brèves de ces phénomènes :

- **La contraction:** elle est utilisée à cause de la liaison de deux phonèmes identiques. Elle peut être obligatoire. Le premier /***l***/ne doit pas être contracté avec le second /***l***/) ou permise (سرر/sarara/=/sarra/). (قل له qul / /lahu= قله/qullahu) , interdite dans (مللت/malaltu).

- **L'élision**: est le changement qui se produit dans la prononciation de la phonème/n/qui porte une *soukoun* devant certaines consonnes.

- **L'assimilation homo-organique des nasales:** elle concerne la substitution d'une consonne nasale par une autre consonne. Elle peut se produit à l'intérieur du mot ou à la frontière de deux mots successifs.

7.5 Les consonnes emphatiques

L'emphase est habituellement utilisé pour rendre compte de manifestations prosodiques liées à l'accentuation volontaire d'une syllabe chez les linguistes arabes il désigne certaines qualités que possèdent les consonnes. Cette qualité décrit le mouvement articulatoire que fait la langue quand elle meut vers la partie postérieure de la cavité buccale avec ou sans tafhim. Plusieurs travaux s'accordent pour dire que les voyelles contribuent à l'emphase. Ainsi qu'elles soient brèves ou longues celles-ci possèdent une disposition spectrale différente selon qu'elles sont au contact d'une consonne emphatique ou non emphatique.

Certaines études affirment que le phénomène de l'emphase dépasse le cadre de la voyelle (ou des voyelles) adjacente et se propage aux phonèmes voisins (dans le mot C1V1C2V2 ….si C1 est emphatique la synthèse est plus naturelle quand la propagation de l'emphase arrive jusqu'àC2). [38] En revanche il existe des divergences sur la portée de cette propagation en d'autres termes, sur la taille du segment sonore affectée par la consonne emphatique. Pour plus de détaille voir. [35, 36, 37]

Les consonnes emphatiques se prononcent différemment selon le dialecte, mais le plus souvent sont pharyngalisées, c'est à dire prononcées comme si on avait la "bouche pleine", par exemple (د non emphatique, ض emphatique). Les consonnes susceptibles d'être emphatiques sont les suivantes : ط , ظ , ص , ض , ق.

8. Caractéristiques phonétiques des phonèmes arabes

8.1 Lieux d'articulation

Le lieu d'articulation est la zone du conduit vocal qui participe à la formation du son, il varie d'un phonème à un autre. Pour les phonèmes arabes, il y a plus de 28 lieux d'articulation, c'est pourquoi nombre de phonéticiens ont pris comme critère de classification, le lieu d'articulation (Table 3.9).

lieux d'articulation	phonèmes
pharyngale	ح,ع
laryngale	هـ , ء
uvulaire	خ,ق
post-palatale	ك
pré-palatale	ى,ر,غ
dentale	ت,ط,د,ض,س,ص,ز,ن
inter-dentale	ث,ذ,ظ
labio-dentale	ف
bilabiale	م,ب,و

Table 3.9 Classification des phonèmes arabes selon le lieu d'articulation. [7]

8.2 Traits distinctifs des phonèmes arabes

La notion de trait exprime une similarité aux niveaux articulatoire et acoustique. Les phonèmes de la langue arabe se regroupent en catégories naturelles dont les éléments partagent des traits distinctifs. Ces traits nous permettent de déterminer les phonèmes en prenant en compte leurs lieux d'articulations. De tous ces traits, nous citons :

Sourd/sonore : lors de la prononciation des phonèmes sourds, les cordes vocales s'écartent mais ne vibrent pas. En revanche pour les phonèmes sonores, les cordes vocales vibrent.

Emphatique : il se traduit par la levée du dos de la langue jusqu'à ce qu'il soit superposé à la zone palatale supérieure. On obtient donc un son de moins en moins aigu.

Nasal : un phonème est dit nasal si sa prononciation se caractérise par l'abaissement du voile du palais, et donc la mise en communication du conduit nasal avec le conduit vocal.

Occlusif / fricatif : ce trait se traduit par une obstruction de l'air dans le conduit vocal. Cette obstruction peut être totale dans le cas des sons occlusifs ou partiels pour les sons fricatifs.

Sonnantes : ces phonèmes ne sont ni fricatifs, ni occlusifs. L'obstacle est donc le plus discret possible.

La classification selon les traits distinctifs est résumée dans la table (3.10).

Trait	Phonèmes
Voisé	ي ب د ذ ر ز ض ط ظ ق ل م ن و
Sourd	ك ت ح خ س ص ف هـ
Emphatique	ص ض ط ظ ق
Nasal	م ن
Occlusif	ب د ط ق ك
Fricatif	ث ح خ ذ ز س ص ض ظ ف هـ
Sonore	ي د ل م ن و

Table 3.10 Classification des phonèmes arabes selon les traits distinctifs.

9- Les problèmes de langage arabe en traitement automatique

La langue arabe est considérée comme faisant partie des langues difficiles à appréhender dans le domaine du traitement automatique du langage écrit et parlé. Cette difficulté est démontrée par les propriétés morphologiques, syntaxiques, phonétiques et phonologiques de la langue arabe.

Une des difficultés de l'arabe en traitement automatique est l'agglutination par laquelle les composantes du mot sont liées les unes aux autres. Ainsi notre étiqueteur morpho-syntaxique identifie-t-il d'abord les composantes du mot. [10]

Dans le domaine du traitement automatique de l'arabe écrit, les recherches ont débuté vers les années 1970, avant même que les problèmes d'édition de textes arabes ne soient complètement maîtrisés. Les premiers travaux concernaient notamment les lexiques et la morphologie. Depuis une dizaine d'années, l'internationalisation du web et la prolifération des moyens de communication en langue arabe, ont révélé un grand nombre d'applications du TALN (traitement automatique du langage naturel) arabe. Les travaux de recherche abordés des problématiques plus variées comme la syntaxe, la traduction automatique, l'indexation automatique des documents, la recherche d'information,… [77]

9.1 Agglutination des mots

La plupart des mots arabes sont composées par agglutination d'éléments lexicaux de base (proclitique +base +enclitique) par exemple la détermination peut s'exprimer par agglutination de l'article ال /al avant le mot الولد/alwaladu (l'enfant) ou par agglutination d'un pronom personnel après celui-ci ولده waladuhu(son enfant)de même les pronoms personnels peuvent se rattacher aux verbes(ضربه / darabahu/ il l'a frappé) les particules régissant le cas

indirect aux noms (كداره/ kadarihi / comme sa maison)et les conjonctions de coordination aux verbes(فذهب /favahaba /il est parti).

9.2Voyellation

Comme nous l'avons évoqué les textes arabes sont ordinairement dépourvus de diacritiques. Pour les lire tout un processus mental est nécessaire: identifier le mot comme appartenant au lexique puis lui attribuer ses voyelles dans son contexte, ce qui nécessite la compréhension du texte ce problème est similaire à celui de l'accentuation automatique des textes en français, mais dans des proportions beaucoup plus importants 28% des mots français en usage sont ambigus contre 95% en arabe (mesures effectuées sur texte de 23000 mots). [9]

10. Le traitement automatique du langage arabe

Des progrès considérables ont été réalisés dans le domaine du traitement automatique de l'arabe parlé, grâce à l'amélioration des technologies du traitement du signal, à l'enrichissement des connaissances sur les caractéristiques prosodiques et segmentales et sur les différentes modélisations acoustiques relatives aux schèmes arabes. Ces résultats devraient permettre de mieux appréhender des domaines variés et innovants tels que la reconnaissance et la synthèse de la parole, la traduction orale ou la reconnaissance automatique du locuteur et de ses origines géographiques. [78]

11. Les corpus

Un corpus est une collection de données langagières qui sont sélectionnées et organisées selon des critères linguistiques et extralinguistiques explicites pour servir d'échantillon d'emplois déterminés d'une langue. A l'origine, le terme désignait des sources documentaires caractérisées par leur exhaustivité, recueils de textes rassemblant exhaustivement tous les documents disponibles pour certains champs d'études. Le corpus Juris par exemple rassemblait tous les documents du droit romain. Le corpus est souvent multimédia (texte, son, image, image animée). L'oral peut ainsi aussi bien s'appuyer sur différents supports: du son, du texte (transcription), de la vidéo (expression). Il a même été constitué un corpus de sites web. [web3]

La plupart des corpus ont été constitués dans le cadre d'une recherche précise et n'ont de pertinence que pour celle-ci. Les conditions de collecte ou le travail très spécifique d'annotation ne permet pas la diffusion de ces données. D'autres corpus ne sont pas disponibles par volonté des chercheurs qui souhaitent garder. Une priorité scientifique sur un

travail de collecte couteux et laborieux. Mais il existe des corpus conçus comme des bases de données qui prennent le statut de corpus de référence par le simple fait que ce sont les seuls disponibles. Ces corpus sont alors utilisés simplement parce qu'ils sont là.

Un corpus permet d'assurer la pérennité des données de travail par leur stockage et leur accessibilité, mutualiser des espaces et des systèmes de stockage adéquats. Car un corpus doit être stocké. Les corpus contient des données acquièrent une valeur patrimoniale. Permettre la conservation d'un patrimoine scientifique, servir à des recherches épistémologiques ou autres que celles qui ont été à l'origine du corpus, permettre des actions de vulgarisation pour un plus large public (participation à la scénographique multimédia d'expositions scientifiques...). Pour cela il faut mobiliser une ingénierie et un outillage spécialisés sur des ensembles plus vastes.

Pour étudier un phénomène langagier il faut un corpus qui dispose d'un grand nombre de données. Les corpus au sens habituel du terme existent depuis la fin du 19^{e} siècle et ont été constitués et utilisés d'abord manuellement pour des études statistiques. La période 1920-1950 a vu l'utilisation des corpus pour l'extraction de listes de mots.

Un corpus représentatif contient toutes les attestations nécessaires pour établir des règles suffisamment générales, il peut être utilisé de deux façons opposées et complémentaires. Il peut servir à établir des règles ou à tester celles-ci. L'intérêt des corpus s'est imposé à toute la communauté scientifique, toutes les écoles ont compris l'intérêt de disposer de grandes quantités de données attestées pour déduire ou tester leurs théories.

Les corpus d'anglais sont les plus nombreux. Le Brown corpus (Brown University Standard Corpus of Present-Day American English), LOB, le Corpus Susanne (Geoffrey Sampson's Suzanne), Le Lancester Oslo Bergen Corpus, le Bank of English (comporte plus de 200 Millions de mots)…, le Brown Corpus est un corpus de référence qui a servi de modèle à d'autres réalisations, souvent cité comme première expérience de corpus linguistique, c'est un corpus d'un million de mots en anglais américain développé à l'université de Brown en 1964. La mise à disposition de ce corpus informatisé dans le domaine public a largement contribué au renouveau de la linguistique de corpus et à l'exploitation nouvelle de l'informatique dans ce domaine au début des années 80.

Le projet NESPOLE, cofinancé par l'union européenne et la NSF (USA), adressait la problématique de la traduction automatique de parole et ses éventuelles applications dans le domaine du commerce électronique et des services. Les langues impliquées étaient: l'Italien,

le Français, l'Allemand et l'Anglais. Les partenaires du projet étaient un ensemble des agences et des laboratoires d'Italie, Allemagne, USA et la France. [web5]

L'oral pose un problème ardu, les corpus oraux sont encore plus difficile à réaliser, même pour les corpus les plus célèbre, par exemple le British National Corpus (BNC) qui est un très grand corpus d'anglais moderne (plus de 100 millions de mots), écrit et parlé. On trouve que la partie réservée à l'oral est de 10% environ10 millions de mots.

Les corpus oraux de l'arabe sont encore moins disponible que les corpus oraux d'autre langues, on ne trouve pas de données même sur le web, on trouve assez peu de renseignements sur la façon de constituer ces corpus. La disponibilité de tels corpus donnera le coup d'envoi aux divers travaux de recherches qui utilisent ces corpus.

12. Conclusion

Ce chapitre a été consacré à la description de la langue arabe avec ses particularités phonologiques et les traits caractéristiques de cette langue et leurs problèmes en traitement automatique. Aussi le problème des corpus puisque le développement des corpus écrits et oraux de l'arabe doit être un enjeu capital pour la politique linguistique des pays arabes. Alors que la plupart des langues européennes disposent de corpus accessibles en ligne, et souvent gratuitement, un tel outil n'existe pas dans notre pays, ce qui a des conséquences néfastes pour la visibilité et la vitalité de cette langue. C'est un enjeu pour la recherche linguistique et pour le développement de l'ingénierie linguistique (reconnaissance et synthèse de la parole, traitement automatique des langues), aussi pour l'enseignement de ces langues, la sauvegarde et la diffusion du patrimoine oral.

Les particularités phonologiques et les traits caractéristiques de la langue arabe ont donné une idée pour le développement d'un système de reconnaissance automatique de la parole arabe. Les détailles de la conception et réalisation du système de reconnaissance automatique sera dans le chapitre suivant.

Chapitre 4

Conception et réalisation

1. Introduction

La parole est le moyen de communication le plus naturel chez l'homme, celui-ci a très vite cherché de l'intégrer dans l'interface homme-machine, cette dernière est une orientation de la recherche en informatique, qui requiert la compétence de plusieurs sciences et techniques telles que: l'intelligence artificielle, le traitement du signal, la linguistique, la phonétique, et bien qu'elle n'a cessé de susciter un intérêt croissant dès les balbutiements du traitement automatique de l'information. Nous pensons que des bases intéressantes d'un système de reconnaissance de la parole passeraient par un modèle cognitif de la machine qui soit fortement inspiré de l'être humain.

Nous proposons dans ce chapitre un système de reconnaissance de la parole arabe indépendant du locuteur, basé sur une approche évolutionnaire. Ce système repose sur un algorithme génétique muni d'un algorithme de recherche tabou avec des entrées fournies par l'algorithme KPPV.

L'algorithme génétique évalue une population d'individus et extrait toutes les solutions possibles. On appliquera par la suite un algorithme de recherche tabou pour extraire la solution optimale et éviter les optimaux locaux. L'algorithme de recherche tabou assure la survie des individus (solutions) voisines de la solution présentée par l'algorithme génétique (solution courant). Il permet au système de reconnaissance de converge vers un optimum globale, et d'être un système de reconnaissance vocale de meilleurs performances.

2. Les modules d'un système de reconnaissance de la parole

Les systèmes classiques de RAP sont composés essentiellement de cinq modules. Un module de paramétrisation du signal, les modèles acoustiques ; qui représentent les unités acoustiques choisies: phonèmes, diphones et mots…, les modèles linguistiques ; qui doivent être une représentation la plus vraisemblable possible du langage, le dictionnaire qui doit contenir l'ensemble des mots que l'on souhaite pouvoir reconnaître et le système de reconnaissance lui-même. Ces différents modules d'un système de RAP, sont relativement indépendants les uns des autres, bien qu'ils sont tous nécessaires pour la reconnaissance de parole.

2.1 Le module de reconnaissance utilisé dans notre système

Durant ces dernières années, plusieurs recherches ont été introduites dans le domaine de la reconnaissance de la parole. Malgré l'efficacité des approches classiques, qui s'avèrent être des outils performants de la parole; grâce à leur capacité d'apprentissage, de généralisation et de classification. Ils existent encore des problèmes liés à leur style d'apprentissage à savoir: le temps d'apprentissage est long, les paramètres initiaux peuvent avoir des effets étendus sur les concepts appris, absence de méthodologies pour le choix d'une technique adéquate au problème de la reconnaissance de la parole.

Les algorithmes génétiques constituent une des approches intéressantes dans ce domaine. Ils fournissent des solutions proches de la solution optimale à l'aide des mécanismes de sélection, de croisement et de mutation. Il est constaté que les solutions fournies par ces algorithmes sont généralement comparables et parfois meilleures que celles obtenues par les méthodes plus classiques, pour un même temps de calcul.

Les algorithmes génétiques ont montrés leur capacité à éviter la convergence des solutions vers des optima locaux, aussi bien lorsqu'ils sont combinés avec des méthodes de recherche locale [6,62,67]. La capacité des algorithmes génétiques dans l'optimisation nous conduit à proposer une approche évolutionniste pour la reconnaissance de la parole. Le risque de la convergence vers des maximums locaux se pose, pour éviter ce problème et améliorer le taux de reconnaissance, on utilise une technique de recherche locale (la recherche tabou).

L'algorithme génétique évalue une population d'individus dans un environnement bien définit, il favorise la survie et la reproduction des individus les mieux adaptés à la solution, à fin de trouver un système de reconnaissance vocale performant.

La performance de l'algorithme génétique dépend de la taille de la population. Selon De jong la taille idéale d'une population est de 50 à 100 individus. [29] L'augmentation de la taille du vocabulaire implique un nombre important des sous populations parcourues par l'AG, pour cela on utilise l'algorithme KPPV, qui limite le nombre de sous populations traitées par l'AG.

3. Structure générale du système proposé

L'objectif de ce projet est d'appliquer les opérateurs génétiques (sélection, mutation, croissement et remplacement) à partir d'un vocabulaire donné et représenté sous forme d'une population d'individus, pour avoir à la fin un système de reconnaissance de la parole arabe

avec un taux de reconnaissance considérable. L'algorithme génétique est muni d'un algorithme de recherche tabou et la population initiale est filtrée par l'algorithme KPPV.

L'algorithme génétique passe par les étapes suivantes :

- Le choix d'un codage des éléments de la population. Le codage binaire a été très utilisé à l'origine. Le codage réel est désormais largement utilisé, notamment pour l'optimisation de problèmes à variables réelles.
- Définition d'un mécanisme de génération de la population initiale.
- La définition de la fonction d'évaluation des individus. Celle-ci est appelée fonction d'adaptation (*fitness*). La solution optimale du problème est obtenue à partir du résultat de la fonction d'évaluation.
- L'utilisation des opérateurs de reproduction permettant de diversifier la population au cours des générations et d'explorer l'espace d'état.
- L'application de l'algorithme de recherche tabou pour l'évitement des maximums locaux.
- La définition des paramètres de dimensionnement: taille de la population, nombre total de générations ou critère d'arrêts, probabilités d'application des opérateurs de croisement et de mutation.

L'organigramme suivant présente l'architecture générale du système proposé.

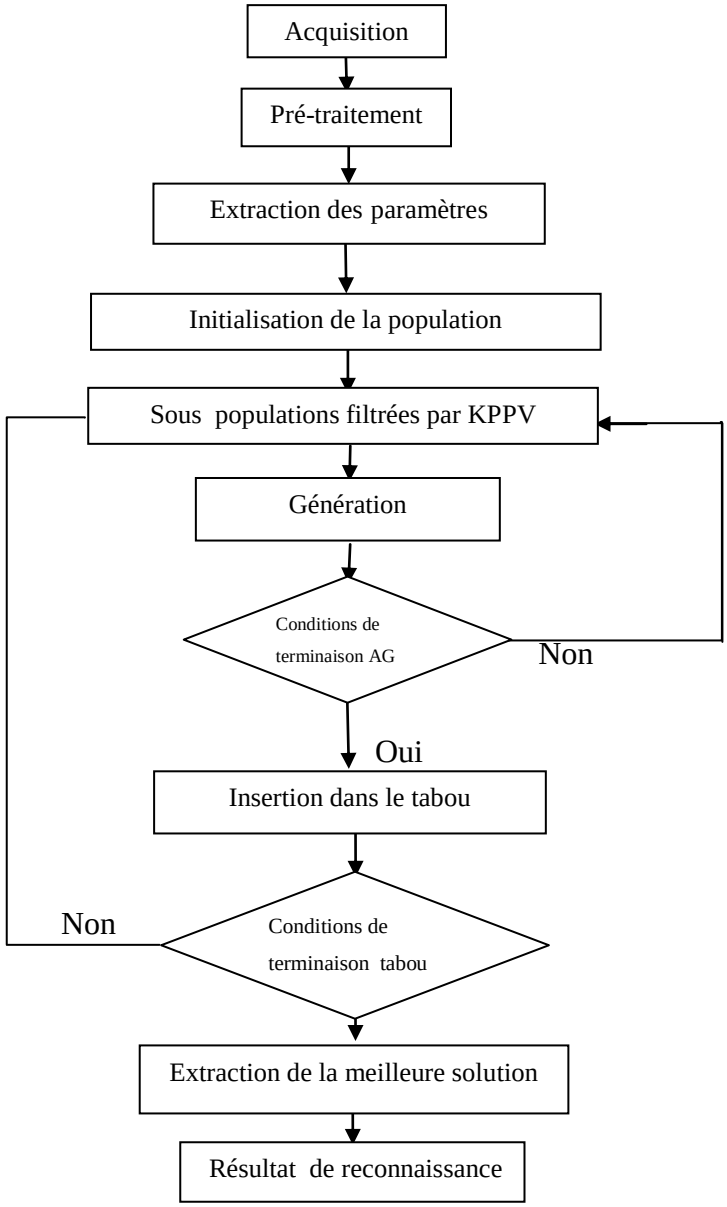


Figure 4.1 Organigramme du système proposé

3.1 Paramétrisation du signal

Il est possible de calculer des coefficients représentatifs du signal traité pour résoudre les problèmes liés à la complexité de la parole. Ces coefficients sont calculés à intervalles temporels réguliers, le signal de la parole est transformé en une série de vecteurs de coefficients. Les vecteurs de paramètres doivent être pertinents (précis, de taille restreinte et sans redondance), discriminants (pour faciliter la reconnaissance) et robustes (aux différents bruits et/ou locuteurs).

3.2 Mise en forme du signal vocal

Avant l'extraction des paramètres acoustiques du signal vocal, la mise en forme du signal de la parole est nécessaire (figure 4.2), pour cela, l'ensemble des opérations suivantes sont prises en considération :

3.2.1 L'échantillonnage

L'opération d'échantillonnage consiste à transformer le signal à temps continu ***x(t)*** en signal à temps discret ***x (nTe)*** défini aux instants d'échantillonnage, multiples entiers de la période d'échantillonnage ***Te***, celle-ci est elle-même l'inverse de la fréquence d'échantillonnage ***fe***. Ce qui concerne le signal vocal, le choix de ***fe*** résulte d'un compromis. Son spectre peut s'étendre jusqu'à 12 kHz. Il faut donc choisir une fréquence ***fe*** égale à 24 kHz au moins pour satisfaire raisonnablement au théorème de Shannon. [13]

3.2.2 La préaccentuation

La préaccentuation est un filtre numérique du premier ordre qui passe après l'échantillonnage selon l'équation suivante : $H(z)= 1 - \alpha\ z\text{-}1$ 4.1

Avec (α=0,97).

3.2.3 Application de fenêtre de pondération (fenêtre de Hamming)

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

$$W(n) = 0.54 + 0.46.\cos\left(2\pi \frac{n}{N-1}\right) \qquad 4.2$$

La fenêtre de hamming est couramment utilisée en reconnaissance de la parole. Ce traitement implique une hypothèse importante du fait des limitations postérieures qu'elle occasionne: le signal vocal est supposé stationnaire sur une courte période.

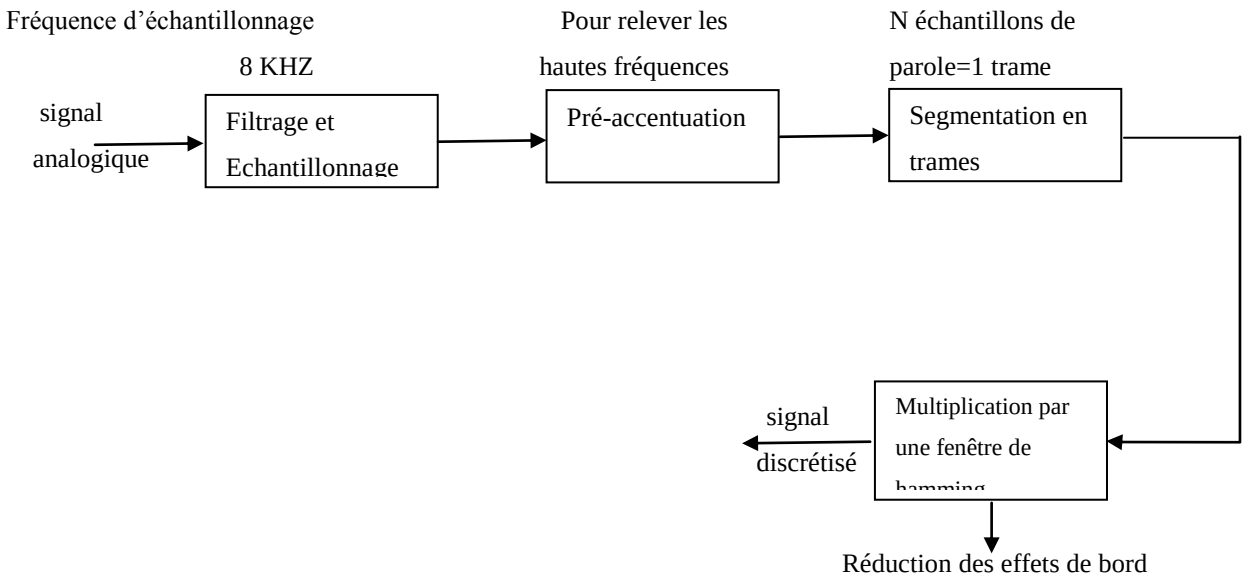


Figure 4.2 Mise en forme du signal vocal. [54]

3.3 Extraction des paramètres acoustiques

L'extraction de caractéristiques est une étape importante dans le processus de reconnaissance de la parole. Elle est divisée en deux parties: étape de codage et l'étape d'extraction d'informations nécessaire à la reconnaissance. Ces dernières doivent représenter au mieux le signal qu'elles sont censées modéliser et extraire le maximum d'informations utiles pour la reconnaissance.

3.3.1 Paramètres acoustiques

Après la phase de mise en forme du signal qui est commune dans la plupart des méthodes d'analyse de la parole, la phase suivante est de déterminer les paramètres pertinents pour la reconnaissance de la parole.

Il existe de nombreux algorithmes pour calculer des vecteurs acoustiques. Ils visent tous à obtenir des vecteurs acoustiques représentatifs de l'information linguistique contenue dans le signal de la parole. Ces algorithmes sont insensibles que possible aux causes non-linguistiques de variabilités tel que: l'identité du locuteur, l'environnement acoustique (le bruit d'ambiance) ou le canal de transmission (la distorsion induite par une ligne téléphonique ou un microphone inadapté). La représentation de dispositif la plus populaire actuellement utilisée est les coefficients de Cepstral de Mel-fréquence ou le MFCC.

3.3.2 La technique d'analyse MFCC (Mel Frequency Cepstral Coefficient)

L'analyse MFCC est l'évaluation de coefficients cepstraux à partir d'une répartition fréquentielle selon l'échelle des Mels. Après la préaccentuation et l'application de la fenêtre de pondération une transformation de Fourier FFT(Fast Fourier Transform) est appliquée puis on applique un banc de filtres Mels. Afin de rapprocher l'analyse en banc de filtres de la perception humaine, les filtres ne sont généralement pas répartis de manière linéaire mais en fonction d'une échelle Mel. La correspondance entre une fréquence en Hz et en Mel se calcule selon l'équation suivante :

$$F_{mel}=2595*\log(1+\frac{F}{700}\text{Hz}) \qquad 4.3$$

Avec F la fréquence en Hertz. Chaque filtre va donner un coefficient cepstrale :

$$s_{i,k}=\sum_{n=0}^{n/2} y_{i,n}\, m_{n,k}, k=0...K \qquad 4.4$$

Avec : $Y_{i\,n}$, le $n^{ème}$ coefficient de la transformée ($n\in[1,N]$), de la $i^{ème}$ fenêtre ($i\in[1, I]$), et $M_{n\,k}$, le $n^{ème}$ coefficient ($n\in[1, N]$) du $k^{ème}$ filtre ($k\in[1,K]$). On utilise communément 12 coefficients, on utilise alors K=13 filtres (pour obtenir 12 coefficients, il faut un filtre de plus car le $0^{ème}$ est inutile).

On a donc $S_{i\,k}$, la matrice de sortie du $k^{ème}$ filtre pour la $i^{ème}$ fenêtre. On a, à cette étape, ce qu'on appelle un Spectre Mel (Spectrum Mel).

Dans l'étape finale, on transforme les données dans l'échelle des Mels (fréquentielle) vers l'échelle des temps. Le résultat de cette étape sera les MFCC proprement dit. Il suffit d'effectuer l'inverse de la transformée de Fourier. Pratiquement, on effectue une transformée en cosinus discrète inverse (iDCT), ce qui revient au même puisque la transformée en cosinus inverse donne la partie réelle de la transformée de Fourier, ici on a que des réels. Il faut noter que la transformée en sinus donnera la partie imaginaire de la transformée de Fourier.

$$C_{i,n}=\sum_{k=1}^{k}(\log_{10}(s_{i,k})).\cos(n(k-\frac{1}{2})\frac{\Pi}{k} \qquad 4.5$$

On exclut le $0^{ème}$ coefficient (C I,0) car il transporte peu d'informations caractéristiques du signal (il représente la valeur moyenne de l'intensité du signal). On a donc K-1 coefficients. Il ne reste plus qu'à recoller les fenêtres pour obtenir le cepstre. [32]

La chaîne complète de calculs des coefficients MFCC est définie par la figure suivante 4.4.

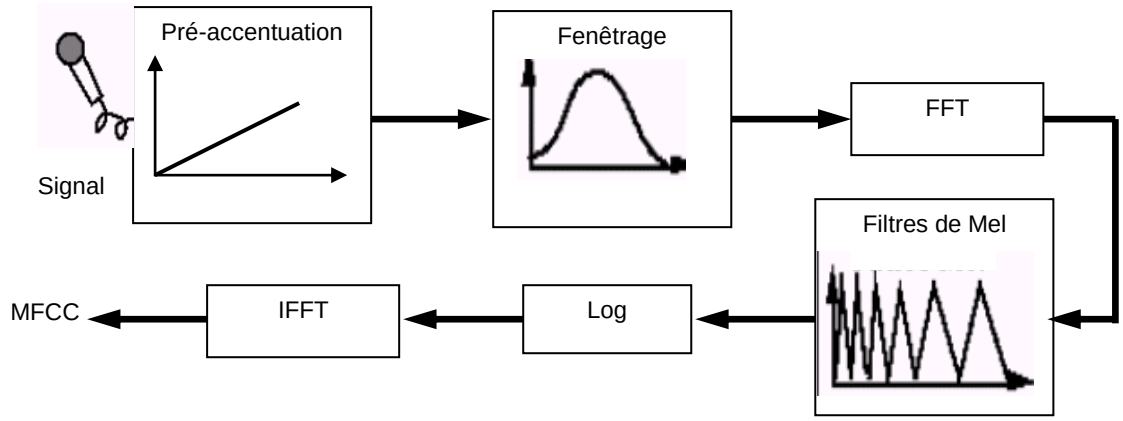


Figure 4.3 schéma de calcul des MFCC

4 La base de données utilisée

Dans notre processus de reconnaissance, les expériences sont effectuées sur une base de données qui a été utilisée pour le développement du système de reconnaissance de la parole arabe présentée par mihoubi. [61]

Le vocabulaire de la base de données contient la plupart des traits caractéristiques de la langue arabe: des mots simples, géminés, emphases, durées et tanwine. La base de données est constituée de 6 sous corpus selon les traits caractéristique de la langue arabe. Chaque sous corpus contient 5 mots prononcées par 6 locuteurs, chaque locuteur a répété chaque mot 3 fois, donc chaque sous corpus contient 90 fichiers sons. En tout le corpus contient 540 fichiers sons. Chaque sous corpus est divisé en deux sections une pour les hommes, l'autre pour les femmes, donc chaque mot du corpus possède 18 occurrences.

Les enregistrements de la base de données sont effectués tel que chaque locuteur lis les mots avec une vitesse moyenne et assure une bonne articulation et évite les perturbations dus aux hésitations, les reprises, les respirations, …

La fréquence d'échantillonnage choisie est de 11025 Hz, les échantillons ont été codés sur 16 bits par échantillon. La base de données contient 540 fichiers utilisés pour l'apprentissage. Nous utilisons dans les différents types des tests :

- Un corpus de test multi locuteur contient 174 fichiers sons.
- Un corpus de test indépendant du locuteur contient 174 fichiers sons.
- Un corpus de test dans environnement bruité contient 174 fichiers sons.

5. Les algorithmes utilisés dans le système de reconnaissance

Les performances du système dépendent de la taille de la population initiale (le nombre des mots utilisés dans le vocabulaire). L'augmentation de la taille du vocabulaire

signifie plus de calcul par l'algorithme génétique, c'est-à-dire il augmente le temps de reconnaissance. On utilise l'algorithme KPPV pour minimiser la taille de la population initiale et fixer le nombre de sous populations traitées par l'algorithme génétique.

5.1 L'algorithme de k- plus proches voisins (KPPV)

L'algorithme de K- Plus proches voisins est lié à la notion de proximité ou de ressemblance. L'idée de cet algorithme est relativement simple. Elle consiste, étant donné un point $x \in \Re^n$ représentant la forme à reconnaître, à déterminer la classe des k points les plus proches de x parmi l'ensemble des formes d'apprentissage et à retenir pour la décision, la classe la plus représentée. Si $k = 1$, le point x est donc simplement attribué à la classe de son plus proche voisin. Cette notion de voisinage est quantifiée par une mesure de similarité. Sachant que la mesure de similarité la plus utilisée est la distance euclidienne.

La méthode des K- plus proches voisines KPPV offre l'avantage d'être très simple et efficace. Son principe consiste à calculer la distance entre l'individu à classer et les individus connus, puis à attribuer le premier à la classe présentant le plus grand effectif parmi ses k plus proches voisins. Outre sa simplicité, cette méthode est couramment employée en reconnaissance des formes, parce qu'elle s'y prête à de nombreux titres tout d'abord, elle ne nécessite pas de connaitre la distribution de probabilité des classes de la population, ce qui est rarement le cas. [54]

Ensuite, on fixe le nombre k de voisins et pas le volume, la méthode ne dépend pas de la densité de probabilité. L'expérience montre de plus que cette méthode présente souvent un bon pouvoir prédictif. [54]

5.2 Justification de l'utilisation de l'algorithme k plus proche voisins KPPV

On utilise l'algorithme KPPV pour limiter le nombre des sous population traitées par l'algorithme génétique et on évite le parcours de toutes les sous populations. L'algorithme KPPV donne le classement le plus probable pour les sous populations. Il choisit un représentant au hasard parmi les individus de chaque sous population, puis calcule la distance euclidienne entre le mot cherché et le représentant, selon cette distance, il ordonne les sous populations. En fin, il fixe le nombre des sous populations présentées à l'algorithme génétique.

5.3 La technique de recherche Tabou (Tabu search). [39,40]

La recherche Tabou est une méthode itérative à solution unique basée sur un algorithme de recherche de voisinage qui commence avec une solution initiale pour l'améliorer pas à pas en choisissant une nouvelle solution dans son voisinage. Elle a montré sa performance sur de nombreux problèmes d'optimisation. Elle n'a aucun caractère stochastique et utilise la notion de mémoire pour éviter la convergence vers un optimum local.

Le principe de l'algorithme est le suivant : à chaque itération, le voisinage (complet ou sous ensemble de voisinage) de la solution courante (extrait par l'algorithme génétique) est examiné et on sélectionne la meilleure solution. La méthode autorise de remonter vers des solutions qui semblent moins intéressantes mais qui ont peut être un meilleur voisinage.

Cette méthode risque de cycler entre deux solutions, pour éviter ce phénomène, la méthode à l'interdiction de visiter une solution récemment visitée. Donc il faut garder la trace de dernières solutions visitées. Ainsi, la recherche de la solution suivante se fait dans le voisinage de la solution courante sans considérer les solutions appartenant à la liste tabou.

Cette méthode ne s'arrête pas d'elle-même, il faut déterminer un critère d'arrêt en fonction du temps de recherche que l'on s'octroie. Ce critère peut être, par exemple, l'exécution d'un certain nombre d'itérations ou la non-amélioration de la meilleure solution pendant un certain nombre d'itérations. Ainsi, tout au long de l'algorithme, la meilleure solution doit être conservée car l'arrêt se fait rarement sur la meilleure solution. Les différentes étapes de l'algorithme peuvent être décrites comme suit :

Algorithme de recherche tabou

Procédure : *φ fonction de fitness*

Variables *Solut_courante, solut_meilleur, solut_init, T liste tabou, iter itération*

Initialiser la liste tabou T=vide

iter= 0 ;

Générer une solution initiale : soul_init ;(par AG) ;

Mettre a jour la solution initial dans le tabou

début

Tant que *(iter<max_iter & tabou non plein) faire*

Application des opérations de l'AG (sélection, croisement, mutation, remplacement)

Calculer la fonction fitness φ(S)

Iter=iter+1

Si condition vérifie (l'individu est une solution courante)

Mise à jours solut(ajouter la solution dans le tabou)

Fin tant que

Pour toutes les solutions qui existent dans le tabou faire

Choisir best_solut

Si φ(solut_courant) > φ (solut_best) alors

Solut_best=solut_courant

Retourner solut_best

Fin programme

6. Détaille du système de reconnaissance proposé

6.1 La démarche de l'algorithme évolutionnaire

La démarche de l'algorithme évolutionnaire utilisé dans notre système est présentée dans la figure (4.4).

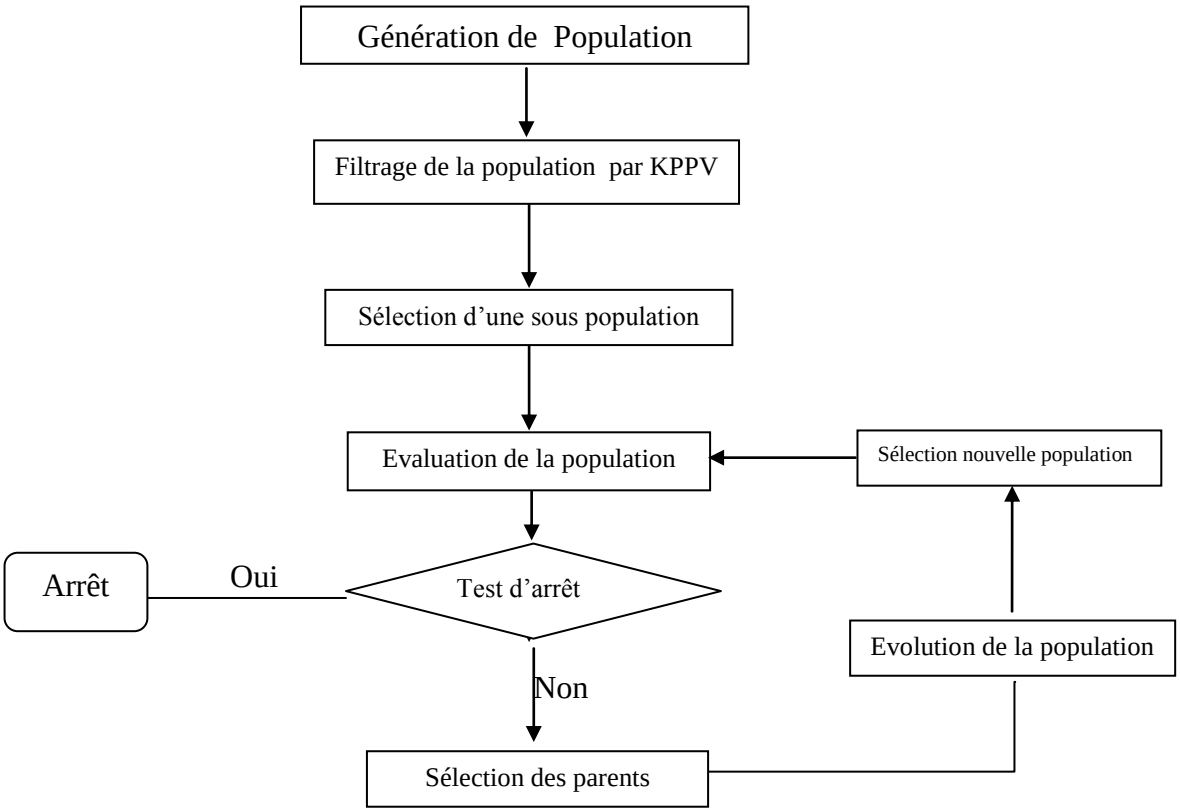


Figure 4.4 schéma de l'algorithme génétique utilisé.

6.2 Le codage

Dans n'importe quel système de reconnaissance, l'objectif dans l'établissement d'un système de codage est de minimiser la distorsion moyenne intra-classe et de maximiser la distance inter-classe.

Le choix de la population initiale d'individus conditionne fortement la rapidité de l'algorithme. Les individus sont générés dans un dictionnaire de références qui contient tous les paramètres MFCCs de tous les mots du corpus d'apprentissage. Le nombre d'unités dans le dictionnaire de références représente la population gérée par l'algorithme génétique. Les individus sont structurés sous forme d'un tableau dans le dictionnaire de références. Chaque ligne du dictionnaire représente une sous population, on trouve dans chaque sous population tous les occurrences du même mot pour tous les locuteurs. Les mots dans le dictionnaire représentent les individus manipulés par l'algorithme génétique.

Les individus de la population sont des cepstres MFCC. Chaque cepstre est une matrice, le nombre de lignes correspond au nombre de coefficients MFCC, le nombre de colonne correspond au nombre de fenêtres dans le signal, chaque élément de cette matrice est un nombre réel, les gènes d'un individu sont les colonnes de la matrice.

6.3 La fonction d'évaluation (fitness)

Chaque individu de la population peut être une solution potentielle au problème à résoudre, mais ces solutions n'ont pas toutes le même degré de pertinence. C'est à la fonction de performance, dite aussi fonction d'adaptation ou *fitness*, de mesurer cette efficacité pour permettre à l'algorithme génétique de faire évoluer la population dans un sens bénéfique pour la recherche de la meilleure solution. Autrement dit, la fonction d'adaptation doit pouvoir attribuer à chaque individu un indicateur positif représentant sa pertinence pour le problème qu'on cherche à résoudre. [28]

La fonction d'évaluation ou d'adaptation (*fitness*) associe donc un coût à chaque individu. C'est grâce à la *fitness* que l'algorithme évalue l'adaptation des individus à leur environnement, et calcule leurs chances de survivre et de se reproduire en favorisant la sélection d'individus dans la direction de l'optimum qui est a priori inconnue. Elle constitue donc le critère à base duquel l'individu serait ou pas sélectionné pour être une solution. La qualité de cette fonction conditionne pour une grande part l'efficacité d'un algorithme génétique. Il est souhaitable de lui rechercher une approximation plus simple, parce que dans le cas où la fonction d'adaptation apparaît excessivement complexe, elle nécessite une puissance importante de calcul.

Dans notre système, la fonction d'évaluation consiste à maximiser la distance euclidienne entre les différents cepstres du même mot (les spectres des occurrences du même mot). La fitness d'un individu est la distance euclidienne entre les deux cepstres du mot recherché et celle de l'individu.

6.4 Techniques d'évolution

La réussite du système dépend de la technique d'évolution utilisée, pour cela il est nécessaire d'effectuer des choix concernant la manière d'évolution de la population. Dans notre système l'algorithme génétique évolue dans des générations pour trouver la référence la plus proche du mot recherché. De tel sorte, dans chaque sous population il exécute un cycle de génération dans laquelle il applique les opérateurs génétiques suivants: sélection, croisement, mutation et remplacement, s'il n'atteint pas le seuil de reconnaissance dans cette sous population après cinq générations, il change la sous population initiale jusqu'à ce qu'il atteigne le seuil de reconnaissance.

6.4.1 La sélection

La sélection permet d'identifier statistiquement les individus de la population courante qui seront autorisés à se reproduire. Cette opération est fondée sur la performance des individus, estimée à l'aide de la fonction d'adaptation. Dans notre étude une fois la population initiale créée, il faut sélectionner les meilleures sous populations. L'algorithme KPPV définit les sous populations qui contiennent des individus qui ont une forte chance d'être des solutions, puis il les présente à l'algorithme génétique. L'utilisation de ce type de sélection à pour but de marquer un maximum dans le plutôt possible et de définir les meilleurs voisinages pour l'algorithme de recherche tabou. Dans une deuxième phase, après le choix de la sous population, on applique une sélection élitiste où les individus de chaque sous population sont ordonnées selon la fonction d'évaluation des individus (fitness d'individu), pour accélérer la convergence vers un maximum.

6.4.2 Le croisement

La puissance essentielle des algorithmes génétiques vient de la reproduction associée au croisement. Ce dernier a pour objectif d'enrichir la diversité de la population en introduisant de nouvelles solutions obtenues par combinaisons des solutions parents. [69] Le type de croisement utilisé dans notre système consiste à utiliser le meilleur individu dans chaque sous population et le combiner avec l'individu en cour du traitement (on combine deux parents pour obtenir deux individus). Donc il est important de garder dans chaque sous population les meilleurs individus pour l'utilisés dans le croissement. Une fois la sous

population est changée on garde de nouveau les meilleurs individus dans la sous population. La probabilité de croisement assigne à cet opérateur dans notre système est de *80%*.

6 .4.3 La mutation

La mutation est l'une des opérateurs utilisés dans la génération des individus (des descendants) avec un taux très faible (généralement 0,01). Il apparaît qu'elle joue bien un rôle secondaire dans la mise en œuvre des algorithmes génétiques. Elle est cependant nécessaire parce que, bien que la reproduction et le croisement explorent et recombinent efficacement les notions existantes, ils peuvent parfois devenir trop zélés et perdre de la matière génétique potentiellement utile. Dans les systèmes génétiques artificiels, l'opérateur de mutation protège contre de telles pertes irrécupérables. [25]

La mutation confère aux algorithmes génétiques une propriété très importante: l'ergodicité (tous les points de l'espace de recherche peuvent être atteints). Elle permet donc d'explorer l'espace de recherche en déplaçant, avec une faible probabilité, des individus dans leurs voisinages. Cet opérateur est donc d'une grande importance, il est loin d'être marginal. [69] Il a de fait un double rôle: celui d'effectuer une recherche locale, et/ou de sortir d'une trappe (optima locaux). [62]

Le principe de mutation utilisé dans notre système consiste à injecter un gène qui se trouve dans un locus déterminé de manière aléatoire du mot recherché dans les individus du dictionnaire de références avec une probabilité de 0,01. [61]

6.4.4 Le remplacement

A chaque génération, un individu existant de la population est choisi par l'algorithme génétique pour libérer une place pour un enfant. Tandis que le processus de reproduction est important pour découvrir des nouvelles régions prometteuses, cet opérateur de remplacement est également très important car il sélectionne des régions qui sont abandonnées. [27]

Après l'application des opérateurs génétiques (sélection, croissement et mutation) dans notre système, il reste à appliquer l'operateur de remplacement pour introduire les descendants obtenus par ces opérateurs génétiques dans la population initiale. Nous utilisons un remplacement élitiste, de sorte qu'un enfant prend place au sein de la population s'il est plus performant que les individus de la population précédente, c'est-à-dire un enfant qui a une fitness meilleure que celles des individus de la population précédente prend une place dans la nouvelle population. [61]

6.5 Critère d'arrêt

Dans un algorithme génétique, il faut déterminer un critère d'arrêt. On peut décider la terminaison des calculs après un certain délai (temps ou nombre d'itérations), après que le meilleur individu ait atteint un certain niveau de qualité, après un certain nombre d'itérations sans amélioration du meilleur individu ou bien lorsque l'écart type de la fitness de la population passe en deçà d'un certain seuil. [30]

Dans notre étude l'arrêt du système dépend du nombre des sous populations choisies par l'algorithme KPPV, ce nombre a été fixé à 15 sous population. Dans le parcours de chaque sous population, on détermine le nombre de générations effectuées avant de passer à la sous population suivante.

Dans une deuxième phase l'arrêt du système dépend de l'état du tabou:

- Si la liste du tabou est pleine on s'arrête sans parcourir toutes les sous populations présentées par l'algorithme KPPV, et on choisit la solution qui a la meilleure fitness.
- Si toutes les sous populations sont parcourues, et la liste du tabou n'est pas pleine on choisit la solution qui a la meilleure fitness dans la liste du tabou.
- Si toutes les sous populations sont parcourues et la liste du tabou est vide, le mot recherché n'est pas reconnu.

7. Evaluation du système

Toutes les méthodes développées dans notre système ont été testées sous Matlab. Tandis que les calculs de MFCC sont issus de Toolboxes pour Matlab.

L'évaluation d'un système de reconnaissance automatique de la parole RAP est obtenue en comparant le décodage d'un certain nombre des mots de test avec un étiquetage de référence de ces mots. On peut alors estimer un taux de réussite du système réalisé. La précision de cet estimateur dépend du nombre d'unités testées qui permet de définir l'intervalle de confiance de l'estimation. La fiabilité de l'estimation du taux de reconnaissance sur un corpus donné dépend du nombre de tests réalisés.

7.1 Reconnaissance en mode multi locuteurs

Le résultat du test en mode multi locuteurs atteint 100% par l'application de l'algorithme génétique dans le système RAPAAG (reconnaissance automatique de la parole arabe par un algorithme génétique) présenté dans [61]. L'application de l'algorithme KPPV dans notre système permet à l'algorithme génétique d'éviter plus de calcul, et les résultats

atteignent 100% pour tous les sous corpus. L'application de la technique de recherche tabou avec l'algorithme génétique donne les même résultats, mais avec plus de calcul qu'un algorithme génétique seul, ou bien un algorithme génétique et l'algorithme KPPV.

7.2 Reconnaissance en mode indépendant du locuteur

Le deuxième test réalisé dans notre étude est la reconnaissance en mode indépendant du locuteur. Ce test est réalisé par un corpus contient les occurrences des mots des locuteurs qui n'ont pas participés où corpus d'apprentissage.

7.2.1 Le premier type de test en mode indépendant du locuteur (RAPAGT: Reconnaissance Automatique de la Parole Arabe par un algorithme Génétique et l'algorithme Tabou).

Le premier type du test consiste à munir l'algorithme génétique par la technique de recherche tabou. Nous avons fait trois essaies pour arriver à une taille parfaite de la liste du tabou: la première avec une taille qui peut contenir jusqu'à cinq solutions, la deuxième avec une taille de quatre solutions, dans la troisième on trouve qu' une liste de trois solutions est suffisante pour marquer toutes les solutions possibles (locale et globale) avec moins de calcul, parce que avec une longue liste du tabou, l'algorithme génétique est obligé de parcourir toutes les sous populations dans la plupart des cas .

7.2.2 Le deuxième type de test en mode indépendant du locuteur (RAPAGK : Reconnaissance Automatique de la Parole Arabe par un algorithme Génétique et l'algorithme Kppv).

Le deuxième test réalisé est un algorithme génétique avec des sous populations filtrées par l'algorithme KPPV. L'application d'algorithme KPPV dans notre système permet de minimiser le nombre des sous populations traitées par l'algorithme génétique, donc au lieu de parcourir toutes les sous populations comme dans le premier test on applique l'algorithme génétique sur les quinze sous populations sélectionnées par l'algorithme KPPV. Dans ce cas l'algorithme KPPV est considère comme une sélection élitiste des sous populations.

7.2.3 Le troisième type de test en mode indépendant du locuteur (RAPAGKT: Reconnaissance Automatique de la Parole Arabe par un algorithme Génétique et l'algorithme KPPV et l'algorithme Tabou).

Le troisième type du test consiste à appliquer avec l'algorithme génétique les deux techniques, la recherche tabou et l'algorithme KPPV afin d'exploiter les avantages de chacune, ce qui constitue une approche intéressante pour un système de reconnaissance de la parole.

Le taux de reconnaissance de ces tests est comparé avec le taux de reconnaissance du système **RAPAAG** présenté par Mihoubi [61]. La table (4.1) présentée les résultats de reconnaissance:

Taux de reconnaissance / Corpus	Test1 RAPAGT	Test 2 RAPAGK	Test 3 RAPAGKT	Système RAPAAG [61]
Corpus simple	80%	76,66%	83 ,33%	74%
Corpus gémination	86,66%	86,66%	90%	70%
Corpus emphase	27%	50%	36,66%	27%
Corpus durée	70%	70%	66 ,66%	53%
Corpus tanwine	75%	62 ,5%	70,83%	54%
Corpus mélange	73,33%	66,66%	73,33%	50%

Table 4.1 le taux de reconnaissance en mode indépendant du locuteur

Les résultats obtenus par les trois types de test en mode indépendant du locuteur sont meilleurs que les résultats du système RAPAAG qui utilise seulement un algorithme génétique.

7.3 Test de reconnaissance dans un environnement bruité

Dans ce test le système de reconnaissance utilisé est un algorithme génétique muni par les deux techniques: l'algorithme de recherche tabou et l'algorithme KPPV (RAPAGKT dans un environnement bruité). Ce test est réalisé par un corpus contient les occurrences des mots du vocabulaire utilisé dans l'apprentissage et enregistré dans un environnement bruité.

La table 4.2 présente les résultats de ce test comparé avec les résultats du système **RAPAAG**, présente par Mihoubi [61] dans le même environnement.

Taux de reconnaissance / Corpus	Test RAPAGKT	Système RAPAAG [61]
Corpus simple	73 ,33%	70%
Corpus gémination	46,66%	50%
Corpus emphase	67%	60%
Corpus durée	60%	30%
Corpus tanwine	47%	25%
Corpus mélange	73,33%	30%

Table 4.2 résultats de reconnaissance dans un environnement bruité

8. Discussion des résultats

Après les différents tests réalisés dans notre étude, on arrivera à des résultats de reconnaissance intéressants sur plusieurs points :

Le système qui utilisé l'algorithme génétique muni d'un algorithme de recherche tabou, permet d'éviter l'erreur de substitution qui est une conséquence de la convergence vers un optimum local. Le problème rencontré avec ce système est le temps du calcul, surtout dans le cas où il n'y a pas des optimums locaux et la solution est unique; l'algorithme continue le parcours de toutes les sous populations, aussi il y a le cas d'utilisation d'une liste longue où l'algorithme ne trouve pas des solutions (absence des optimums locaux) pour remplir cette liste ce qui l'oblige de parcourir toutes les sous populations.

L'utilisation de l'algorithme KPPV permet de remédier le problème du temps, il minimise le nombre des sous populations présentées à l'algorithme génétique. Aussi on peut considérer le calcul réalisé par l'algorithme KPPV comme une première génération ce qui augmente l'espace d'exploration de la population. Aussi il est utilisé comme une sélection élitiste des sous populations. Donc on peut dire que, l'algorithme génétique proposé est un algorithme à double sélection. Le problème qui se pose avec l'algorithme KPPV est dans le représentant de chaque sous population, parce que leur choix est au hasard. Dans le cas où il ne représente pas bien la sous population, il permet d'écarter complètement une solution potentielle de toutes les sous populations présentes à l'algorithme génétique, au bien présentée dans une

position qu'on ne peut pas atteindre par l'algorithme génétique puisque ce dernier marque un critère d'arrêt.

Selon les résultats présentes précédemment dans les différents tests on peut conclure que le système s'arrêt sur le type de corpus, la différence dans le taux de reconnaissance entre les types des sous corpus permet de dire qu'on peut réaliser un système de reconnaissance vocal pour une application spécifique ou bien un domaine défini avec le choix soigneusement du vocabulaire, on utilise des mots simples et géminés.

9. Conclusion

Dans ce chapitre nous avons présenté le système réalisé dans notre étude, il se base sur un algorithme génétique pour la manipulation d'une base de données, cette dernière construire la population initiale d'un algorithme génétique et la technique k-plus proche voisines (KPPV) qui a un rôle de filtrage et de minimisation du nombre des sous populations présentées à l'algorithme génétique. Aussi la technique de recherche tabou pour marquer tous les solutions et éviter les maximums locaux.

Le système de reconnaissance de la parole a subi une série des tests. Les résultats sont satisfaisants et permettent de conclure les avantages des techniques utilisées dans ce système qui s'ajoute aux peu des systèmes de reconnaissance de la parole arabe.

Conclusion et perspective

Conclusion et perspective

La parole est le principal moyen de communication dans toute société humaine, et certainement le moyen le plus naturel de communication. Pour autant il est tout aussi certain qu'il est plus facile, de point de vue sémantique. Il constitue un défi pour les chercheurs dans le développement des systèmes de reconnaissance de la parole, pour faciliter la communication homme / machine et permettre la manipulation des machines en langage naturel.

Dans cette étude nous avons proposé un nouveau système de reconnaissance de la parole arabe; qui se base sur une approche évolutionniste. Nous avons essayé d'éviter deux problèmes dans cette approche. Le premier est la taille de la population initiale de l'algorithme génétique et le deuxième est le problème des maximums locaux. Pour cela, nous avons proposé deux niveaux d'amélioration. Dans la première amélioration, nous avons proposé un algorithme KPPV pour le filtrage de la population initiale, et la minimisation du nombre de sous populations présentées à l'algorithme génétique. La seconde amélioration a été mise en œuvre par l'application de la technique de recherche tabou. Nous sommes arrivés à notre système final après une série des tests de combinaisons des algorithmes:

- Le premier système est un algorithme génétique muni d'un algorithme recherche tabou, ce type du test est utilisé pour éviter l'erreur de substitution (erreur fatale dans les systèmes de RAP), qui est une conséquence de la convergence vers des optimums locaux. Ce système est testé seulement en mode indépendant du locuteur.
- Le deuxième système est un algorithme génétique avec l'algorithme KPPV. Le rôle de ce dernier a été la minimisation du nombre de sous populations traitées par l'algorithme génétique. L'algorithme KPPV est considéré comme une sélection élitiste des sous populations, aussi une première génération de l'algorithme génétique. Ce système est testé en mode indépendant du locuteur.

- Le troisième système est un algorithme génétique muni d'une technique de recherche tabou et des entrées fournies par la technique KPPV. Dans ce système nous avons exploité les avantages des algorithmes génétiques dans l'optimisation des fonctions complexes et ses différents domaines d'application et les deux techniques: la

recherche tabou, l'algorithme KPPV, ce qui a permis d'avoir un système de reconnaissance vocale avec des performances comparables aux systèmes réalisés par les approches classiques. Le troisième système a subi à trois modes de test: multi locuteur, indépendant du locuteur et dans un environnement bruité.

La reconnaissance automatique de la parole reste une tâche difficile, un grand nombre de voies de recherche restent ouvertes pour traiter tous les aspects de la reconnaissance automatique de la parole. Les résultats que nous avons obtenus par nos systèmes sont très intéressants. Cependant, tous les problèmes liés aux approches que nous avons proposés sont loin d'être résolus. De nombreuses études peuvent être envisagées afin de valider ces approches et il reste à faire pour améliorer les systèmes proposés. Nous suggérons quand même quelques axes de recherche en guise de prolongement à ce travail.

- Elargir le vocabulaire du corpus d'apprentissage et varier les conditions d'enregistrement pour contribuer aux testes de la robustesse des systèmes de reconnaissance automatique de la parole Arabe.

- Il faut étendre ce système et d'appliquer une approche analytique, en utilisant des modèles de phonèmes pour la reconnaissance de la parole continue.

- D'autre part, pour un vocabulaire étendu, il est intéressant d'utiliser des modèles de phonèmes au lieu de mots, du fait que le nombre de phonèmes qui permet la construction de n'importe quel mot est faible, ce qui facilite l'entraînement avec des bases relativement petites.

- Comparer les performances du système de reconnaissance proposé pour les différentes techniques d'analyse acoustique, les mêmes conditions d'expériences doivent être réutilisées pour les différents tests (corpus d'apprentissage, nombre d'itérations, ...), en utilisant les coefficients LPC, PLP, …, ainsi que des combinaisons éventuelles de ces derniers avec l'énergie (E) et le TPZ (taux de passage par zéro), ou même des combinaisons entre ces coefficients.

Références bibliographies

Références

[1] Aissiou. M, Guerti. M "Modèle génétique en vue de l'identification acoustique des voyelles de l'arabe standard". 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications, pp 27-31, Tunisia, March 2005.

[2] Aissiou. M, Guerti. M "Genetic algorithms application for the automatic recognition of the arabic stop sounds". Journal of Applied Sciences Research, 2007, INSINET Publication.

[3] Allauzen Alexandre "Modélisation linguistique pour l'indexation automatique de documents audiovisuels". Thèse de doctorat. Lab Lims _Cnrs colaboration avec institut national de l'audiovisuel (laurent vient). France. 2003.

[4] Amiar Lotfi "Un système hybride AG/PMC pour la reconnaissance de la parole arabe". Thèse de magistère. Université d'Annaba. Algérie. 2005.

[5] Atal B.S, Hanauer S "Speech analysis and synthesis by linear prediction of the speech wave". Journal of the Acoustical Society of America. pp 637-655, 1971.

[6] Back T "Self-Adaptation in genetic algorithms". Proceedings of the first European conference on Artificial Life, 1992.

[7] Bahi-abibet Halima " *NESSR :* Un système neuro-expert pour la reconnaissance de la parole". Thèse de doctorat. Université d'Annaba. Algérie. 2005.

[8] Bali.H, Sellami M "Connexionnist expert system for Arabic speech recognition" The International Arab Journal of Information Technology, pp 149-154. 2005.

[9] Baloul.Sofiane "Développement d'un système automatique de synthèse de la parole à partir du texte arabe standard voyellé". Thèse de doctorat le mans, France.

[10] Baloul S, Boula Philippe " Un modèle syntactique prosodique pour la synthèse de la parole à partir du texte en arabe standard voyellé". Université de Maine. France.

[11] Barras Claude "Reconnaissance de la parole continue: Adaptation au locuteur et contrôle temporel dans les modèles de Markov cachés". Thèse de doctorat. Paris VI. 1996.

[12] Bechet Frédéric "Système de traitement de connaissances phonétiques et lexicales: Application à la reconnaissance de mots isolés sur de grands vocabulaires et à la recherche de mots cibles dans un discours continu". Thèse de doctorat, Université d'Avignon, LIA, France 1994.

[13] Bellanger Maurice "Traitement numérique du signal, théorie et pratique". Editions Masson, 1ère édition en 1980. Actuellement en 5ème édition, ISBN 2-225-84997-8, 1995.

[14] Bellik. Yacine "Multimodal text editor interface including speech for the blind". Speech Communication, 23, 319-332. 1997

[15] Bellman Richard "Dynamic programming". Princeton University Press, 1957.

[16] Bontemps Christophe “Principes mathématiques et utilisations des algorithmes génétiques”. 1995.

[17] Bourlard H, Wellekens C-J “Multi-layer perceptrons and automatic speech recognition”. Proceedings of the IEEE First Annual International Conference on Neural Networks, pp: 407-416, San Diego, 1987.

[18] Bourlard H. Wellekens C-J “Links between markov models and multi-layer perceptrons”. IEEE Transactions on Pattern Analysis and Machine Intelligence, 12 pp 1167-1178, 1990.

[19] Calliope “La parole et son traitement automatique”. Editions Masson, Paris 1989.

[20] Carbonell N, Damestoy J-P, Fohr D, Haton J-P , Lonchamp F, Aphodex, “design and implementation of an acoustic-phonetic decoding expert system”. Proceedings of International Conference on Acoustics Speech and Signal Processing (ICASSP’1986), tome 11, pp 1201- 1204, Tokyo, Japan, 1986.

[21] Chapanis. A “Interactive communication: A few research answers for a technological explosion”. Communication présentée au cours de la CEE. Orsay, F .1979 .

[22] Christophe Gérard “Etude de la paramétrisation du signal de parole à partir de représentation en ondelettes”. décembre 1995.

[23] Christophe Lévy “Modèles acoustiques compacts pour les systèmes embarqués”. Thèse de doctorat. Université d’Avignon et des pays de Vaucluse. 2006.

[24] Charbuillet.C, Gas.B, Chetouani.M, Zarader J.L “Combinaison de codeurs par algorithme génétique : Application à la vérification du locuteur”. Université Pierre et Marie Curie-Paris6, Colloque GRETSI, pp 11-14, Troyes, France.

[25] Cerf R “Une théorie asymptotique des algorithmes génétiques”. Thèse de Doctorat, Université de Montpellier II. Passive, France .1994.

[26] Claire Waast-Richard “Contribution à l'élaboration d'un système de reconnaissance de parole continu à grand vocabulaire”. Thèse de doctorat 1994.

[27]Daniel Cosmin Porumbel “Algorithmes heuristiques et techniques d'apprentissage applications au problème de coloration de graphe”. Thèse de doctorat Université Angers. 2009.

[28] Davis Lawrence “Handbook of genetic algorithm”, Ed. VNR, New York, 1992.

[29] De Jong K “An Analysis of the behavior of a class of genetic adaptive systems”. Doctoral disertation, University of Michigan. 1975.

[30] Delaplace Alain “Approche évolutionnaire de l’apprentissage de structure pour les réseaux bayésiens”. Thèse de doctorat Université de tours. France. 2007.

[31] Derouault. A “Context-dependent phonetic markov models for large vocabulary speech recognition”. ICASSP, Dallas, pp 360-363.1987

[32] Dominique Vaufreydaz “Modélisation statistique du langage à partir d'Internet pour la reconnaissance automatique de la parole continue ”. Thèse de doctorat de l’université Joseph Fourier - grenoble1. France 2002.

[33] Fabrice Lefevre “Estimation de probabilite non-parametrique pour la reconnaissance markovienne de la parole ”. Université de Pierre et Marie Curie 2000.

[34] Gauvain J, Mariani J, Liénard J.S. “On the use of time compression for wordbased speech recognition”. IEEE – ICASSP, Boston 1983.

[35] Ghazali. Salem “Back consonants and backing coarticulation in arabic”. Thèse Phd. Université de Texas. 1977.

[36] Ghazali. Salem “La diffusion de l’emphase l’inadéquation d’une solution auto-syllabique”. Analyse théorie.

[37] Ghazali.Salem, A.brahim “voyelles longues et voyelles brèves en arabe standard : organisation temporelle”. Actes des19ème journées d’études sur la parole, bruxelles, pp.89-93. mai 1992.

[38] Ghazali.Salem, m.zrigui, Z.miled et H.jemni, “Synthèse de l’arabe standard à partir du texte par TD-PSOLA : Le traitement des processus phonologiques”. Actes des19ème journées d’études sur la parole, bruxelles, pp.89-93. mai 1992.

[39] Glover. F “Future paths for integer programming and links to artificial intelligence”. Computers and Operations Research, 1986.

[40] Glover. F and Laguna. M “Tabu Search”. Springer, 1997.

[41] Goldberg D “Genetic algorithms in search, optimization and machine learning”. Addison Wesley, Massachusetts, 1989.

[42] Gravier G, Bonastre J-F, Geoffrois E, Galliano S, McTait K, Choukri K. ESTER “Une campagne d’évaluation des systèmes d’indexation automatique d’émissions radiophoniques en français”. Journées d’études sur la parole JEP’2004, pp 253-256, Fès, Maroco, 2004.

[43] Hamdi Rym “La variation rythmique dans les dialectes arabes”. Thèse de doctorat Université Lumière Lyon2 France, 2007.

[44] Haton Jean-Paul “Reconnaissance et compréhension automatique de la parole”. Publication H1940, 1982.

[45] Haton Jean-Paul “Speech recognition for unknown communication channels”. Proceeding of the ESCA-NATO tutorial and research workshop on robust pont-à-mousson, France, 1997.

[46] Haton J-P, Bonneau A, Fohr D, Laprie Y, Gong Y et Pierrel J- M "Décodage acoustico-phonétique: Problèmes et éléments de solution". Traitement du Signal, pp 293-313, 1990.
[47] Holland J.H "Adaptation in natural and artificial systems". Ann Arbor, The University of Michigan Press, 1975.
[48] Jelinek. F "Continuous speech recognition by statistical methods". In Proceedings of the IEEE, volume 64, pp 532 - 556, 1976.
[49] Juidette H "Contribution à la mise en œuvre de techniques de planification de chemin et d'optimisation". Thèse doctorat, faculté des sciences, Rabat. 2002
[50] Kada Ahmed "Contribution aux techniques et algorithmes évolutifs de traitement de l'information et de communication : applications en télécommunications, sécurité et réseaux". Thèse de doctorat. Rabat Maroc.
[51] Kevin Power "The listening telephone - automating speech recognition over". The PSTN BT Technology Journal, pp 112-126, Janvier 1996.
[52] Khairallah Khouja Mohamed, Mounir Zrigui. " Durée des consonnes géminées en parole arabe : Mesures et comparaison". Laboratoire RIADI, unité de Monastir. Récital Doudan, juin 2005.
[53] Khairallah Khouja Mohamed, Mounir Zrigui et Mohamed Benahmed "Etude acoustique de la durée de la gémination pour la parole arabe". 3rd International Conference: Sciences of Electronic, Technologies of Information and Telecommunications , pp 27-31, Tunisia 2005.
[54] Lazali Lilia "Système neuro- markovien basé sur la fusion de données floues et génétiques : Application pour la reconnaissance automatique de la parole". Thèse de doctorat Université de Annaba, Algérie. 2007.
[55] Léon Bottou "Une approche théorique de l'apprentissage connexionniste : Applications à la reconnaissance de la parole". Thèse de doctorat. Université de paris sud. France 1991.
[56] Lévy-Schoen A. "L'étude des mouvements oculaires". Paris : Dunod. 1969.
[57] Lee. Kai-Fu "Large-vocabulary speaker-independant continuous speech recognition: The SPHINX System". PhD Thesis Carnegie Mellon, Pittsburgh 1988.
[58] Man K.F, Tang K. S, Kwong S. "Genetic algorithms: Concepts and designs". 2000.
[59] Mariani J, Prouts B, Gauvain J.L, Gangolf J.T. "Man machine speech communication systems, including word-based recognition and text to speech synthesis". IFIP World Computer Congress, Paris, 1983.
[60] Miet G "Towards wideband speech by narrowband speech bandwitdth extension : magic effect or wideband recovery". Thèse de doctorat, Université de Maine, 2001.

[61] Mihoubi F “La reconnaissance automatique de la parole approche évolutionniste cas de l’arabe”. Thèse de magistère, Université de Oum el Bouaghi Algérie 2007.
[62] Nicolas Durand “Algorithmes génétiques et autres outils d’optimisation appliqués à la gestion de trafic aérien”. 5 octobre 2004.
[63] Olivier Deroo “Modèles dépendants du contexte et méthodes de fusion de donnés à la reconnaissance de la parole par modèles hybrides HMM/MLP”. Thèse de doctorat. Université Mons France. 1998.
[64] Philippe Gelin “Détection de mots clés dans un flux de parole : Application à l’indexation de documents multimédia”. Thèse de doctorat. Ecole Polytechnique Fédérale de Lausanne.
[65] Rebreyend Pascal “Algorithmes génétiques hybrides en optimisation combinatoire ”. Thèse de doctorat Université de Lyon France.
[66] Renaud D “Synthèse de comportements animaux individuels et collectifs par algorithmes génétiques”. Département Informatique, Institut d'Intelligence artificielle, Université de Paris-8, 1995.
[67] Richard k. Belew, Jhon Mclnerney, Nicol Schraudolph, “Evolving networks: Using the genetic algorithm with connectionist learning”, Cognitive Computer Science, Research Group, California at San Diego, June 1990.
[68] Saidane Tahar, Mounir Zrigui, Mohamed Ben Ahmed. “La transcription orthographique-phonetique de la langue arabe”. Récital, Fès, pp 19-22 avril 2004.
[69] Salah Eddine Merzouk “Problème de dimensionnement de lots et de livraisons: Application au cas d’une chaine logistique”. Thèse de doctorat Université de technologie de Belfort-Montbéliard 2007.
[70] Sébastien Demange “Contributions à la reconnaissance automatique de la parole avec données manquantes”. Thèse de doctorat. Université Henri Poincaré Nancy1 France 2007.
[71] Souquet Amédée, Radet François-Gérard “algorithmes génétiques”. Thèse de fin d’année, Tutorat de Mr Philippe Audebaud 2004.
[72] Spalanzani Anne “Algorithmes évolutionnaires pour l’étude de la robustesse des systèmes de reconnaissance automatique de la parole”. Thèse de doctorat de l’Université Joseph Fourier - Grenoble I. France 1999.
[73] Stern P.E “An expert system for speech spectrogram reading”. Proceedings of International Conference on Acoustics Speech and Sign al Processing (ICASSP), tome 11, pp 1193 -1196, Tokyo, Japan, 1986.
[74] Syswerda Gilbert “Uniform crossover in genetic algorithms”. Proceedings of the third

conference on genetic algorithms, ICGA, pp 2-9 1989.
[75] Tremain T "The government standard linear predictive coding algorithm". Speech Technology Magazine, pp 40-49, 1982.
[76] Tubach.J "La parole et son traitement automatique". Masson, collection technique et scientifique des télécommunications édition, 1989.
[77] Waibel A, Hanazawa T, Hinton G, Shikano K, Lang K "Phoneme recognition using time-delay neural networks". IEEE Transactions on Acoustics, Speech, and Signal Processing, 37: pp 328-339, 1989.
[78] Zaki , A. Rajouani , M. Najim . "Un Modèle prédictif de la durée segmentale pour la synthèse de la parole arabe à partir du texte" XXIVèmes Journées d'Étude sur la Parole, Nancy, 24-27 juin 2002.
[79] Zue V, Lamel L "An expert spectrogram reader: A knowledgebased approach to speech recognition". Proceedings of International Conference on Acoustics Speech and Signal Processing, ICASSP, tome 11, pp 1197- 1200, Tokyo, Japan, 1986.

Références Internet

[web1] Jean-Philippe Genetic Algorithm Viewer : Démonstration d'un algorithme génétique. http://www.rennard.org/alife/english/gavgb.html Jean-Philippe phd 2000.
[Web 2] Algorithmes génétiques Jean-Marc Alliot, Nicolas Durand.
[web3] Marc Antoniotti. «recueil d'un corpus électronique a partir du web » – 2002 m.antoniotti.free.fr/memoire.htm.
[web4] Sid-Ahmed Selouani, Jean Caelen, « Reconnaissance de traits phonétiques de l'arabe : comparaison d'un système connexionniste modulaire et d'un système a base de connaissances », institut d'électronique-Alger. http://www.clips.imag.fr.
[web5] http://nespole.itc.it.

Annexe

1. Langage de programmation utilisé

Nous décrivons brièvement le langage de programmation que nous avons utilisé pour le développement de notre système de reconnaissance de la parole arabe.

MATLAB (Matrix Laboratory) est un langage de calcul numérique, utilisé des matrices comme objets de base. Ces matrices peuvent aussi être des vecteurs (ligne ou colonne), ainsi que des scalaires. Il manipule aussi bien les nombres réels que les nombres complexes.
Matlab a une syntaxe simple, ce qui offre avec la structure de ce langage les mêmes possibilités que les langages de programmation structurée. Il fait sa popularité, grâce à leur facilité avec laquelle on peut programmer des méthodes numériques, les tester et visualiser les résultats sous forme graphique. C'est un logiciel qui se destine plus à l'expérimentation numérique.

Matlab contient des fonctions spécialisées, pour cela on peut le considéré comme un langage de programmation adapté pour les problèmes scientifiques. Il fonctionne dans plusieurs environnements. Il est devenu aujourd'hui un langage de programmation complet dans un environnement de développement simple, puissant et multi plateforme (Windows / Unix Linux). La puissance de ce langage vienne des bibliothèques *Toolboxes* qui sont des collections de fichiers (M-file) développés pour des domaines d'application spécifiques (Signal Processing Toolbox, System Identification Toolbox, Control System Toolbox, u-Synthesis and Analysis Toolbox , Robust Control).

Signal processing toolboxes et auditory toolboxes sont les plus importante boites à outils utilisées pendant la réalisation de notre système. Matlab présente d'importants avantages pour l'étude des algorithmes génétiques et pour le traitement du signal.

2. La base de données utilisée

Les expériences sont effectuées sur une base de données qui a été utilisée pour le développement du système RAPAAG de reconnaissance de la parole arabe présentée par mihoubi.[61]

La base de données a été enregistrée dans d'excellentes conditions de prise de son, dans la salle acoustique du studio NUMERA (Constantine), et à travers un microphone de très haute qualité, mis à la même distance de la bouche de chacun des locuteurs participant aux enregistrements (environ 10cm), tous dans la même salle acoustique. Les mots du vocabulaire sont lus avec une vitesse moyenne, ce qui assure une bonne articulation et évite les

perturbations dus aux hésitations, les reprises, les respirations. L'âge des locuteurs est entre 22 et 55 ans pour les deux sexes.

Les mots du corpus ont été sélectionnés par des linguistes de l'institut de la langue Arabe de l'université de Constantine, ils contiennent la plupart des traits caractéristique de la langue arabe : des mots simples, géminés, emphases, durées et tanwine.

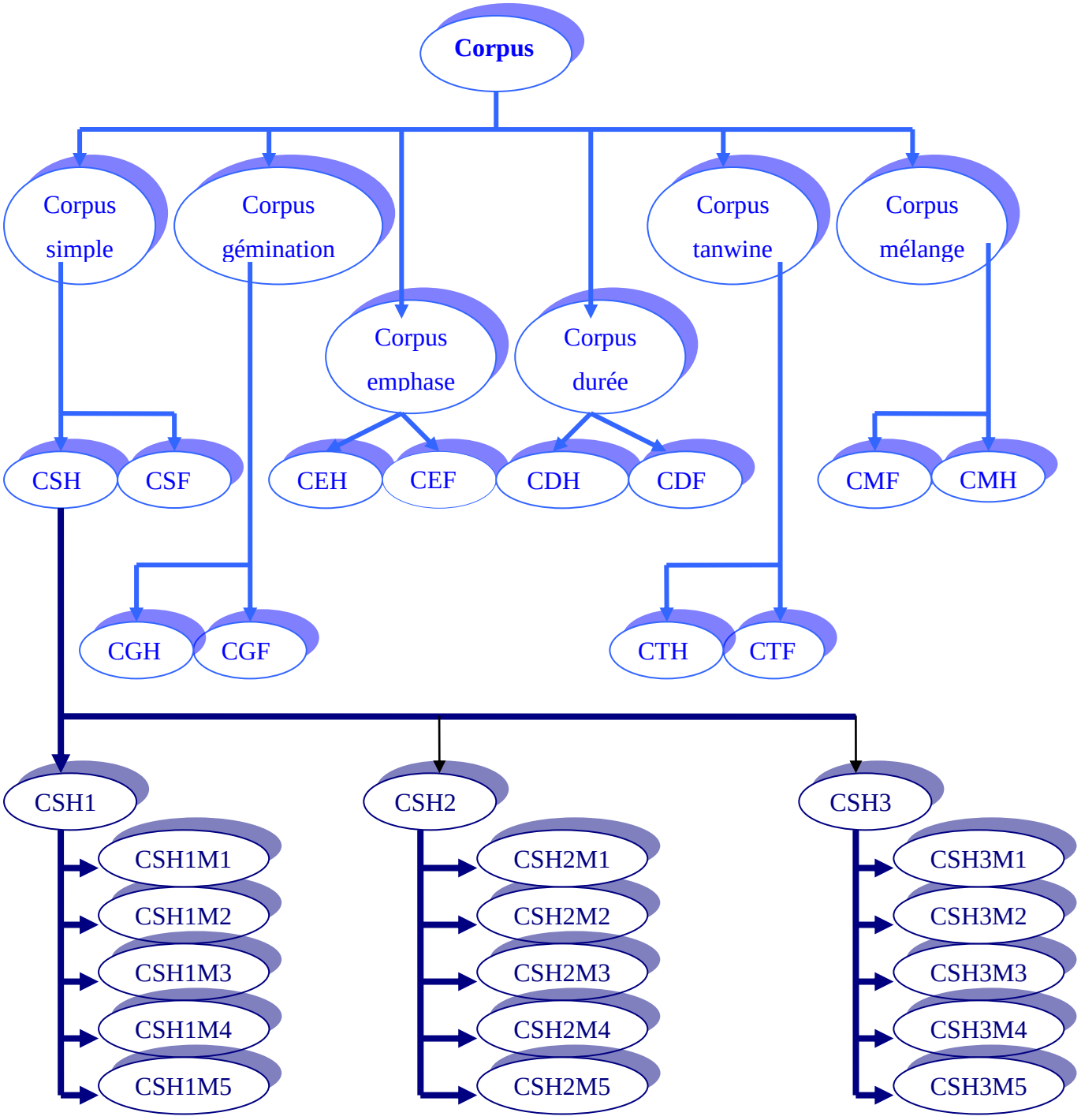
Le vocabulaire du corpus est regroupé dans le tableau suivant :

Le type du corpus	Les mots du corpus
Corpus simple	وزن – كتب – عرف – دحرج – شحذ
Corpus gémination	قيِّد – الشّمس – فسّر – كرّس – رَدَّ
Corpus emphase	طبع ، صرف ,ضرب ، قرب ، نظر
Corpus durée	هتاف ، غروب ، يؤول ، مماليك ، نادى
Corpus tanwine	عزف ، فرش ، منزل ، لون ، مكتب
Corpus mélange	مشتط ، قض، خزاعة ، اضطر ، مواد

2.1 La structure du corpus

La base de données est constituée de 6 sous corpus selon les traits caractéristique de la langue Arabe. Chaque sous corpus contient 5 mots prononcées par 6 locuteurs, chaque locuteur doit répéter chaque mot 3 fois, donc chaque sous corpus contient 3*5*6 = 90 fichiers sons. En tout le corpus contient 90*6 = 540 fichiers sons. Chaque sous corpus est divisé en deux sections une pour les hommes, l'autre pour les femmes, donc chaque mot du corpus possède 18 occurrences. Ce corpus est utilisé dans la phase d'apprentissage.

Les feuilles de l'arbre suivant contiennent les enregistrements des locuteurs. Par exemple la feuille CSH1M1 rassemble les trois fichiers sons (.wav) : CSH1M11, CSH1M12, CSH1M13. Le fichier CSH1M11 sauvegarde la première prononciation du mot simple M1 par l'homme H1.



Légende : CSH : corpus simple hommes, CSF : corpus simple femmes

CSH1 : corpus simple homme n=°1, CSH1M1: le premier mot simple prononcé par l'homme H1, …

La fréquence d'échantillonnage choisie est de 11025 Hz, les échantillons ont été codés sur 16 bits par échantillon.

On utilise dans les différents types de test trois corpus structurés de la même manière que le corpus d'apprentissage:

- Un corpus de test multi locuteur contient 180fichiers sons. Les fichiers sont prononcées par deux locuteurs qui participants en corpus d'apprentissage. Ce corpus est constitué de 6 sous corpus. Chaque sous corpus contient 5 mots prononcées par 2 locuteurs, chaque locuteur doit répéter chaque mot 3 fois, donc chaque sous corpus contient 3*5*2 = 30 fichiers sons. En tout le corpus de test contient 30*6 = 180 fichiers sons.
- Un corpus de test indépendant du locuteur contient 180fichiers sons. Les fichiers sont prononcées par deux locuteurs qui ne participants pas en corpus d'apprentissage.
- Un corpus de test dans un environnement bruité contient 180fichiers (sons + bruit).

Les trois corpus de test sont structurés de la même manière que le corpus d'apprentissage.

3. Le système réalisé (RAPAG)

Le système RAPAG (**R**econnaissance **A**utomatique de la **p**arole **A**rabe par un algorithme **G**énétique) est un système de reconnaissance des mots isolés, il se compose en réalité de trois systèmes de reconnaissance indépendants. La figure suivante illustre le système réalisé.

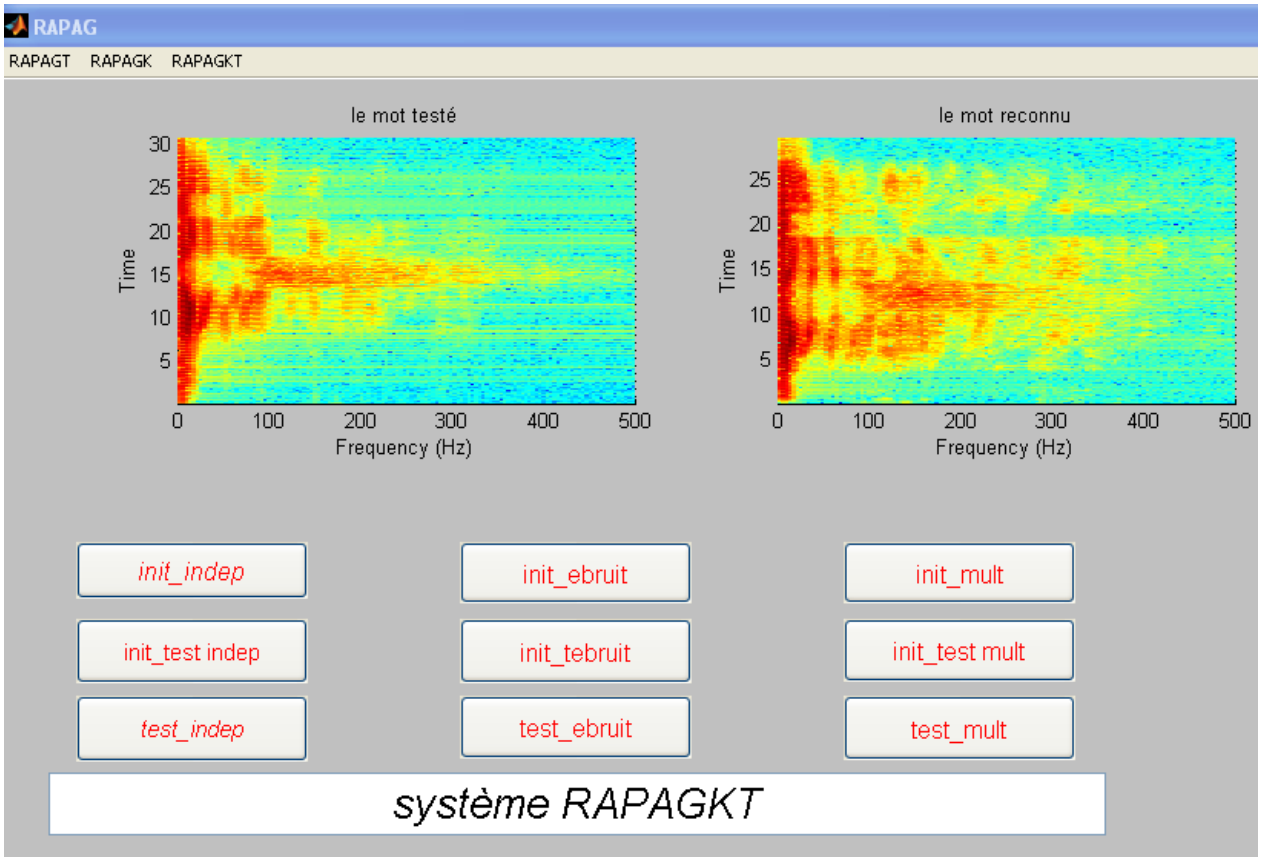


Figure a.1: une interface du système réalisé(RAPAG)

Le système RAPAGT

Le système **RAPAGT** est un l'algorithme génétique muni d'un algorithme de recherche tabou. Ce système est testé seulement en mode indépendant du locuteur.

Le système RAPAGK

Le système **RAPAGK** est un algorithme génétique avec des sous populations filtrés par l'algorithme KPPV. Le système RAPAGK utilise aussi un seul type de test en mode indépendant du locuteur.

Le système RAPAGKT

Le système principal **RAPAGKT** qui est un algorithme génétique avec l'application des deux techniques: la recherche tabou et l'algorithme KPPV. Dans ce système trois types de test sont réalisés: un test en mode multi locuteur, un test en mode indépendant du locuteur et un test dans environnement bruités.

4. Les phases d'exécutions du système

L'exécution de chaque sous système passe par les étapes suivant :

La phase d'initialisation

Dans cette phase le système **RAPAG** structure les individus sous forme d'un tableau dans un dictionnaire de référence, dont chaque linge du dictionnaire représente une sous population, qui contienne toutes les occurrences du même mot.

Le dictionnaire de référence contient touts les paramètres MFCCs de touts les mots du corpus d'apprentissage, où les coefficients MFCC sont calculés dynamiquement lors de la lecture des fichiers d'apprentissage.

Les individus de la population sont des cepstres MFCC. Un cepstre est une matrice, le nombre de lignes correspond au nombre de coefficients MFCC, le nombre des colonnes correspond au nombre des fenêtres dans le signal.

La phase d'initialisation du test

Dans cette phase le système RAPAG charge les fichiers à tester pour la phase suivante. Les fichiers de test sont structurés aussi sous forme d'un tableau.

La phase de test

La phase de test est la plus intéressante, le système charge les fichiers à tester. Le mot cherché subi au même prétraitement qu'a été faite dans la phase d'apprentissage, puis le système applique le module de reconnaissance utilisé pour chaque sous système.

A la fin de chaque test un processus statistique calcule: le taux de reconnaissances, le taux de substitutions (faux reconnaissance) et le taux d'élisions (non reconnaissance).