وزارة التعليم العالي و البحث العلمي

Université Badji Mokhtar -Annaba-
Badji Mokhtar -Annaba- University

جآمعة بآجي مختآر عنآبة

Année : **2015**

Faculté des sciences de l'ingéniorat
Département d'informatique

THÈSE

Pour obtenir le diplôme de
Docteur $3^{ème}$ cycle

Évaluation de la prononciation

Filière : Informatique
Spécialité : Science et Technologie de l'Information et de la Communication

Par :
Khaled NECIBI

Directeur de Thèse : **Halima BAHI - ABIDET** Prof. Université de Annaba

Devant le jury :

Président : Hayet Farida MEROUANI	Prof. Université de Annaba
Examinateur : AbdelKrim BOUKABOU	MCA. Université de Jijel
Examinateur : NourEddine DOGHMANE	Prof. Université de Annaba
Examinateur : Reda ADJOUDJ	Prof. Université de Sidi-Bel-Abbes
Invité : AbdelHalim BAAZIZ	Dr. Université de Annaba

Dédicace

Afin d'être reconnaissant envers ceux qui m'ont appuyés et encouragés à effectuer ce travail de recherche, je dédie ce mémoire :

À Madame Abidet Bahi Halima.

Qu'elle trouve ici l'expression de ma profonde gratitude.

Remerciements

J'adresse mes remerciements infinis au professeur Madame Abidet Bahi Halima, Directrice de cette thèse, pour son important rôle dans l'aboutissement de ce travail de recherche. Je n'aurais pas pu faire ce travail sans leur ferme volonté de m'accompagner jusqu'au bout. J'ai su par ailleurs, à travers elle, qu'il y a des gens pour qui la justice, la solidarité et le partage ne sont pas de vains mots, un témoignage que partagent tous ceux qui l'ont côtoyée. Je resterai très attaché aux nombreuses marques positives qu'elle a gravée sur mon chemin et qui me seront indéniablement utiles pour la suite de ma carrière. Qu'elle trouve ici l'expression de ma profonde gratitude.

Je remercie tous ceux sans qui cette thèse ne serait pas ce qu'elle est, aussi bien par les discussions que j'ai eu la chance d'avoir avec eux que par leurs suggestions.

Je tiens a remercier Madame le professeur Hayet Merouani qui m'a fait l'honneur de présider le jury de cette thèse. Messieurs Abdelkrim Boukabou, NourEddine Doghmane, Reda Adjoudj qui m'ont fait l'honneur de participer au jury de soutenance ; je les en remercie profondément. Je tien à remercier aussi Monsieur le Docteur AbdelHalim Baaziz d'avoir accepter l'invitation pour assister à ma soutenance.

Je remercie tout les membres de ma famille ainsi que tous mes amis pour leurs encouragements et leur assistance aussi bien matérielle que morale qui m'ont permis de faire cette thèse dans des meilleurs conditions.

Enfin, et vu que la difficulté tient toujours dans le fait de n'oublier personne ! je tiens à remercier tous ceux dont le nom n'apparait pas dans les remerciements et qui m'ont aidé d'une manière ou d'une autre. Ils se reconnaitront, je veux simplement leur dire merci.

ملخص

هذه الأطروحة هي نتيجة البحوث التي أجريت في ثلاثة مجالات لتكنولوجيا الكلام: التعرف على الكلام التلقائي، الكشف عن خطأ النطق وتقييم النطق عن طريق ردود فعل بالمعلومات. تعلم النطق الصحيح للغة، وخصوصا عندما الطلاب الشباب الذين لديهم صعوبات النطق والمشاركة، يشكل تحديا كبيرا. تنفيذ نظام يسمح بتحليل الكلام المستقبل لتمكين التقييم التلقائي للنطق أمر صعب. مع دمج التقنيات التي تستند إلى التعرف على الكلام التلقائي، وأصبح تقييم النطق ممكنا. هذا التقييم هو، في معظم الحالات، يقدم في شكل ردود بالمعلومات وعلامات يلخص الجودة الشاملة من النطق. هذا هو النهج إلى استعمال تكنولوجيا الكلام التلقائي في خدمة نظام تعليم النطق.

والهدف الرئيسي من هذه الرسالة هو تصميم وتنفيذ أداة تقييم النطق بالعربية، التي تسمح بفصل الأصوات الصحيحة من تلك غير صحيحة. هكذا، ردود الفعل الناتجة عن تطبيق النظام المبرمج لتقييم النطق تكون مفهومة و مستوعبة من طرف المتعلم بطريقة أفضل.

ومن بين التطبيقات الممكنة من التعرف على الكلام التلقائي هناك أنظمة التدريس النطق بمساعدة الحاسوب. هذه الأخيرة يمكن استخدامها للكشف عن الأخطاء المحتملة في النطق وجود اللغة. ولكن أيا كان تطبيق المستهدف، فان عنصرا أساسيا في النظم هو تقييم النطق باعتبارها ردود فعل من طرف البرنامج و التي هي بمثابة تقييم مختص في النطق.

سنقترح، عبر هذه الاطروحة، نظام تقييم نطق اللغة العربية الذي من خصائصه أن لا يكون عام بما فيه الكفاية، و لكن أن يتميز بقدرة التمييز بين النطق الصحيح و النطق الغير متقن وذلك بتطبيق ثلاثة تقنيات التي تستعمل بطريقة مباشرة نماذج ماركوف المخفية. هذه الثلاث تقنيات هم: تقنية تقييم النطق بالاحصاء الاوتوماتيكي، تقنية المقارنة بين جميع علامات النطق المقترحة و التقنية المبنية على المنطق الغير مرئي لتقييم النطق.

كلمات المفتاح: تعلم النطق، تعلم اللغة، التعرف الاوتوماتيكي على الكلام، التقييم، نماذج ماركوف المخفية، الحكم على النطق، التحقق الاحصائي، المنطق الغير مرئي، تقييم إختصاصيي النطق.

Abstract

This thesis is the result of the research performed in three fields of speech technology domain: automatic speech recognition, mispronunciation detection and pronunciation assessment in form of an informative feedback. Learning the correct pronunciation of a language, especially in the case where the learners may have pronunciation difficulties, is a big challenge. Implementing a system that can receive and analyze the speech to enable the automatic assessment of the pronunciation is a hard task to do. With the integration of the technics that are based on the automatic speech recognition technology, pronunciation assessment is becoming feasible. This evaluation, in most cases, is given in forme feedback score that summarize the global quality of the pronunciation. In other words, this is done by taking advantages of automatic speech recognition in order to implement an Arabic pronunciation teaching system.

Among the possible applications of automatic speech recognition, we can find computer assisted pronunciation teaching systems. The former can find their use in language learning context in order to detect possible mispronunciations. But, whatever the application needed, an important key in computer assisted pronunciation teaching systems is pronunciation evaluation in form of an informative feedback.

We propose an Arabic pronunciation assessment system which will be not too generic, but in the same time a system that can separate a good pronunciations from those that were badly realized. This is done on the basis of three technics that apply the use of hidden Markov models. Those three technics are: a statistical decision technique, a comparison technique of pronunciation scores that exists in the literature, and finally a technique based on the use of fuzzy logic for the evaluation of the pronunciation.

Key words: Pronunciation learning, language learning, speech recognition, evaluation, HMM, decision, student test, fuzzy logic evaluation, expert evaluation.

Résumé

Cette thèse est le résultat de la recherche effectuée dans trois domaines de la technologie de la parole : la reconnaissance automatique de la parole, la détection des erreurs de prononciation et l'évaluation de la prononciation sous forme de feedbacks informatifs. Apprendre la prononciation correcte d'une langue, surtout quand des jeunes écoliers qui ont des difficultés de prononciation sont impliqués, représente un grand challenge. Mettre en œuvre un système qui prend en charge la réception et l'analyse de la parole pour permettre une évaluation automatique de la prononciation est une tâche difficile. Avec l'intégration des techniques qui sont basées sur la reconnaissance automatique de la parole, l'évaluation de la prononciation est devenue faisable. Cette évaluation est, dans la plus part des cas, fournie sous forme de feedbacks informatifs ou scores résumant la qualité globale de la prononciation. Il s'agit de mettre les approches de la reconnaissance automatique de la parole au service d'un système d'enseignement de la prononciation.

L'objectif principal de cette thèse vise à la conception et la réalisation d'un outil d'évaluation de la prononciation en Arabe qui ne soit pas aussi global mais qui arrive à séparer les sons correctes de ceux qui sont incorrectes. Ainsi les feedbacks fournies seront plus correctes et plus précis.

Parmi les applications possibles de la reconnaissance automatique de la parole on trouve les systèmes d'enseignement de la prononciation assisté par ordinateur (CAPTs). Ces derniers trouvent leur utilisation en apprentissage de langues afin de détecter les erreurs de prononciation possible en la présence d'une langue. Mais quel que soit l'application ciblée, un élément incontournable dans les systèmes CAPTs est l'évaluation de la prononciation sous forme d'un feedback informatif.

Nous proposerons un système d'évaluation de la prononciation en Arabe qui ne soit pas aussi générique mais qui arrive à séparer une prononciation correcte de celle mal réalisée en se basant sur trois techniques qui appliquent d'une manière concrète l'utilisation des modèles de Markov cachés. Ces trois techniques sont : une technique statistique de décision, une technique de comparaison de scores de prononciation existants et une technique

basée sur la logique floue pour l'évaluation de la prononciation.

Mot clés : Apprentissage de la prononciation, apprentissage de langue, reconnaissance de la parole, évaluation, HMM, décision, teste student, logique floue, évaluation experts.

Table des figures

Liste des tableaux

Table des matières

Introduction Générale

1.1 Introduction

L'arrivée des ordinateurs dans nos écoles et nos maisons nous incite à réfléchir à leur utilisation bénéfique, en particulier, par l'introduction de l'apprentissage des langues assisté par ordinateur (en anglais CALL pour Computer Assisted Language Learning) dans le processus de l'apprentissage des langues. Le but d'un système CALL est d'aider et d'accompagner les débutants dans leur processus d'apprentissage.

Dans ce contexte, la disponibilité et les performances spectaculaires de la technologie de reconnaissance automatique de la parole (RAP) (en anglais ASR pour Automatic Speech Recognition), a conduit à l'émergence d'une partie des systèmes CALL : les systèmes d'apprentissage de la prononciation assisté par ordinateur (en anglais CAPT pour Computer Assisted Pronunciation Teaching). En effet, la technologie RAP permet la reconnaissance de ce que dit l'apprenant mais aussi l'évaluation de cette prononciation.

Ainsi, les besoins en logiciels d'enseignement de la prononciation assisté par ordinateur sont de plus en plus grands, que ce soit comme aide à l'enseignement en classe ou comme outil d'apprentissage autonome. Avec l'intégration des techniques qui sont basées sur la reconnaissance automatique de la parole, les systèmes CAPT sont devenus de plus en plus performants. Ainsi, l'ordinateur peut comprendre ce que l'apprenant est entrain de prononcer et réagit en conséquence, il en résulte un processus d'apprentissage en temps réel en fournissant des feedbacks sur la qualité de la prononciation de l'apprenant. En effet, un des problèmes majeurs concernant les applications basées sur le principe des systèmes CAPT est le feedback. D'une façon similaire à celle de l'enseignement classique, le but des systèmes CAPT est de fournir des feedbacks instantanés sur la qualité globale de la prononciation de l'apprenant. Les tentatives de la réalisation de cet objectif ont vu le jour de nombreuses façons différentes ; Certaines applications offrent des courbes d'intonation et d'autres fournissent des spectrogrammes, des codes avec des couleurs ou des scores numérique...etc. Malheureusement, les modes dont ces feedbacks sont fournis n'informent que rarement l'apprenant concernant les erreurs de prononciation qu'il a commises au lieu de le motiver afin d'améliorer sa prononciation, de plus la justesse de ces feedback est souvent contestée au vue de la délicatesse de la tâche.

Nous nous attachons dans ce travail à produire une appréciation quant à la prononciation de jeunes élèves Algériens. La conduite de ce travail a abouti, notamment, à une caractérisation des scores en rapport avec la langue Arabe.

1.2 Contexte de la thèse

L'apprentissage automatique de la prononciation est une orientation à part entière dans les travaux de recherche. Elle trouve ses applications principalement dans l'appren-

tissage des langues (L2), mais aussi dans des systèmes de tests de performances de la maitrise d'une langue, ces tests se font soit en ligne mais le plus souvent par téléphone.

D'un autre côté, les avancées technologiques réalisées dans le domaine de la reconnaissance automatique de la parole ont eu un impact indéniable sur le domaine de l'apprentissage de la prononciation.

Nous nous situons dans ce travail dans le contexte des systèmes CAPT basé ASR. Il s'agit de mettre les avancées de la reconnaissance automatique de la parole au service des systèmes CAPT. En effet, un système CAPT basé ASR comprend deux modules principaux : le module de reconnaissance vocale et le module d'évaluation qui est aussi appelé le module de notation (en anglais scoring). A cet effet, il existe plusieurs recherches, la plupart d'entre elles prennent les modèles de Markov cachés (en anglais HMM pour Hidden Markov Models) en tant que noyau du système de reconnaissance de la parole. La tâche d'évaluation est basée sur les sorties du module de reconnaissance, ces sorties permettent le calcul des mesures utilisées pour la notation de la prononciation et de son évaluation.

1.2.1 La reconnaissance automatique de la parole

La reconnaissance automatique de la parole est le processus par lequel la machine tente de « décoder » le signal de la parole qui lui est destiné. Les recherches relatives à la RAP débutèrent dans les années 1950, dans une conjoncture optimiste, car on pensait que les avancées technologiques des ordinateurs rendraient la reconnaissance une tâche aisée. Quelques dizaines d'années plus tard, on se rendait compte que c'était faux, et que la reconnaissance automatique de la parole demeurait un problème difficile. Aujourd'hui encore nombre de questions restent posées, les difficultés majeures étant associées à la taille du vocabulaire à reconnaître, à la reconnaissance de la parole continue, à la reconnaissance indépendamment du locuteur, à la parole spontanée...etc.

C'est aussi, une discipline qui prend de plus en plus d'ampleur et dont les applications sont aussi nombreuses que diversifiées.

1.2.2 L'apprentissage de la prononciation assisté par ordinateur

L'une des applications possibles de la reconnaissance automatique de la parole est l'apprentissage de la prononciation. Comme nous l'avons précédemment souligné, les systèmes CAPT trouvent leur utilisation en apprentissage des langues, mais on peut aussi envisager d'autres application telles que : l'apprentissage de la lecture du Coran ou encore comme outil pour détecter et diagnostiquer des troubles éventuels du langage (la dyslexie par exemple), ou encore détecter les trouble de la parole comme les troubles d'articula-

tion. Mais quel que soit l'application ciblée, un élément incontournable dans les systèmes CAPT est l'évaluation de la prononciation en termes d'un feedback informatif et surtout très proche de ce qu'aurait prodigué des experts humains.

Lorsqu'on adopte une approche CAPT basé ASR, on se base sur la technologie des HMM et le feedback généré est globalement construit sur le calcul du logarithme de vraisemblance (log likelihood) entre un modèle de référence et la prononciation actuelle (il peut être vu comme étant une distance entre une prononciation correcte et celle qui ne l'ai pas) afin de bien distinguer entre ce qui a été bien prononcé par l'apprenant et ce que ne l'a pas été.

C'est dans ce contexte de l'évaluation de la prononciation que nous nous situons et particulièrement pour l'évaluation de la prononciation de la langue Arabe où très peu de travaux existent.

Bien que comme l'affirment [1] : "The Arabic language is both challenging and interesting", la langue Arabe manque de ressources ce qui entrave l'apparition d'applications commerciales compétitives. En ce qui est de l'évaluation de la prononciation en particulier, affirment de leur coté que : "Arabic speech processing research is actually in its beginning, thus there is no research or system found for pronunciation scoring of Arabic. Therefore studies are needed". Par ce travail nous souhaitons contribuer auprès des travaux qui se sont intéressés à l'évaluation de la prononciation de la langue Arabe.

1.3 Contribution

Ce travail de thèse nous a conduit à investiguer le champ de l'apprentissage de la prononciation et plus spécifiquement celui de l'évaluation de la prononciation. Ainsi, comme première contribution au sujet de cette thèse, nous avons réalisé un état de l'art sur les systèmes CAPT afin de bien comprendre les différents concepts liés à ces systèmes, ainsi que les stratégies adoptées pour permettre une évaluation automatique de la prononciation.

Sur le plan des propositions et réalisations, nous avons proposé une méthode de décision quant à la justesse ou non d'une prononciation basé sur un test statistique, celui de *Student*. Après une étude approfondie sur les évaluations des experts humains et où on fait le constat qu'on est loin d'un consensus entre les différents évaluateurs, survient notre seconde proposition qui consiste en la réalisation d'un système de décision basé sur *la logique floue*.

En marge de cette thèse, nous avons pu réaliser aussi un autre état de l'art sur la détection des différents de troubles de la parole dont le but était de comprendre le processus de l'évaluation quant aux apprenants qui ont un trouble de la voix. Ce travail a été publié sous forme de « Book chapter » dans un livre intitulé "Speech, Image and Lan-

guage Processing for Human Computer Interaction : Multi-modal Advancements" publier par IGI-global dans sa $31^{ème}$ édition. Le titre du book chapter [2] est : "Speech disorders recognition using speech analysis".

1.4 Organisation de la thèse

Cette thèse traite du thème de l'évaluation de la prononciation, elle vise à présenter l'essentiel du travail effectué et tente d'offrir une référence aux chercheurs qui souhaitent s'investir dans ce domaine. Elle est structurée comme suit :

La thèse débute par une introduction générale au travail, citée dans le premier chapitre, au travers de sa problématique et son positionnement dans son contexte.

Le deuxième chapitre est composé de trois parties. La première partie est une introduction à la reconnaissance automatique de la parole et une présentation des modèles de Markov cachés. La seconde partie introduit les outils de reconnaissance de parole les plus connus et en particulier le fonctionnement de la librairie de Sphinx, que nous avons utilisé pour la mise en œuvre de l'application. Quand à la troisième partie, les considérations pratiques qui ont permis la mise en place du système de reconnaissance sont données.

Le troisième chapitre est composé de deux parties. La première partie est dédiée à la présentation du domaine de l'apprentissage de la prononciation assisté par ordinateur. En particulier, on y présente un certain nombre de systèmes illustratifs. La seconde partie est dédiée à la présentation des mesures utilisées pour l'évaluation de la prononciation ; ces mesures sont des scores obtenus suite à la phase de reconnaissance.

Le quatrième chapitre présente notre première contribution qui consiste en une étude comparative entre ces différents scores dans le contexte de la langue Arabe et de l'Anglais.

Le cinquième chapitre est dédié à la présentation de notre deuxième contribution qui est la proposition d'une approche statistique de décision pour l'évaluation de la prononciation.

Le sixième chapitre fait ressortir concrètement le rôle central des experts lors de la mise en place d'un système CAPT et tente d'y remédier par le biais d'une approche basé sur les ensemble *flous en évaluation*.

Finalement, une conclusion et des perspectives du travail, cités dans le septième chapitre, sont présentées.

Reconnaissance automatique de la parole : Application à la langue Arabe

2.1 Introduction

Si l'homme a la faculté de comprendre un message vocal provenant d'un locuteur quelconque, dans des environnements souvent perturbés par le bruit, quel que soit son mode d'élocution, la syntaxe et le vocabulaire utilisé, la machine est-elle capable d'en faire autant ? Une solution peut-elle répondre en globalité à ces difficultés ? Le problème de la reconnaissance vocale est un sujet d'actualité et pour l'instant, seules des solutions partielles sont aptes à répondre aux différentes tâches que la machine doit effectuer.

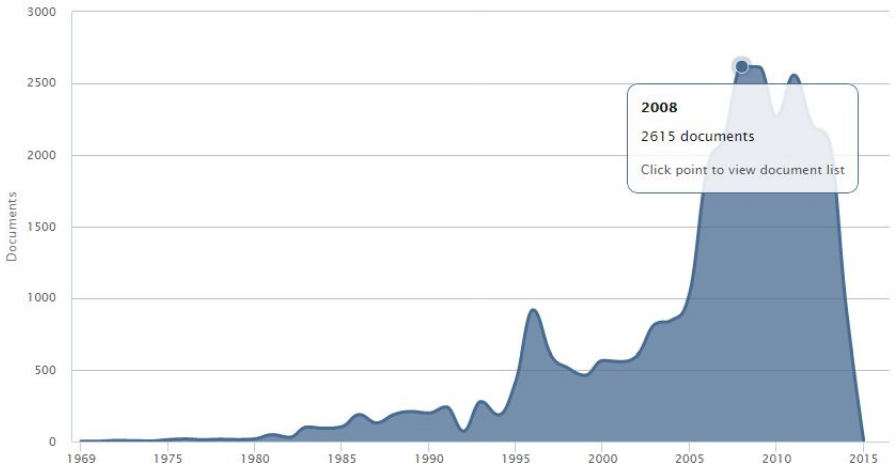

FIGURE *2.1* – Progression du nombre d'articles parus sur la RAP par année

L'intérêt pour ce domaine de recherche et de développement apparaît dans la figure 2.1 qui nous montre la progression significative du nombre d'articles parus et qui traitent de ce sujet.

La reconnaissance automatique de la parole est une technique informatique qui permet d'analyser la parole captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine. La reconnaissance de la parole, ainsi que la synthèse de la parole, l'identification du locuteur ou la vérification du locuteur, font partie des techniques de traitement de la parole. Ces techniques permettent notamment de réaliser des interfaces vocales c'est-à-dire des interfaces homme-machine (IHM) où une partie de l'interaction se fait grâce à la voix. Parmi les nombreuses applications, on peut citer les applications de dictée vocale sur PC où la difficulté tient à la taille du vocabulaire et à la longueur des phrases, mais aussi les applications téléphoniques de type serveur vocal où la difficulté tient plutôt à la nécessité de reconnaître n'importe quelle voix dans des conditions acoustiques variables et souvent bruyantes (téléphones mobiles dans des lieux publics).

La technologie de la reconnaissance automatique de la parole impose plusieurs limites importantes sur le fait de rendre possible l'implémentation d'une certaine application.

Par exemple, il est impossible de reconnaitre n'importe quel mot de la langue Arabe, seuls les mots qui sont définit dans un modèle de langage sont pris en charge. Autrement dit, le moteur de la reconnaissance automatique de la parole doit utiliser un ensemble de restrictions fournit avec un modèle de langage pour améliorer les performances des résultats.

Ensuite, nous avons besoin de vérifier la disponibilité des ressources nécessaires pour l'apprentissage, le test et l'optimisation du système. En effet l'ensemble de test représente une issue critique pour n'importe quelle application de reconnaissance automatique de la parole. L'ensemble de test devrait être assez représentatif. Mais l'ensemble de test ne doit pas nécessairement être très grand. Pour l'ensemble de l'apprentissage et les modèles, il faudra vérifier les ressources qui sont mises en disposition.

Le présent chapitre est divisé en trois parties. La première partie est dédiée à une introduction au domaine de la reconnaissance automatique de la parole. On y présente en particulier, les phases principales pour la mise en place d'un système de reconnaissance ; il s'agit de l'analyse du signal, du décodage acoustico-phonétique et de la décision (reconnaissance). Cette étape induit une modélisation des entités à reconnaître, les modèles principalement utilisés sont les modèles de Markov cachés. La deuxième partie est une présentation du *toolkit* que nous avons utilisé dans cette thèse, en l'occurrence : *Sphinx*, quand à la troisième partie, elle regroupe les considérations pratiques que nous avons mis en place pour pouvoir disposer d'une application basée sur la reconnaissance de la parole. Le but du travail étant de faire le point sur la possibilité de concevoir un outil d'évaluation automatique de la prononciation basé sur la reconnaissance automatique de la parole Arabe.

2.2 Partie 1 : La reconnaissance automatique de la parole

2.2.1 Un peu d'histoire

La reconnaissance automatique de la parole (RAP) est une discipline quasi contemporaine de l'informatique. Vers 1950 apparut le premier système de reconnaissance de chiffres, appareil entièrement câblé et très imparfait. Vers 1960, l'introduction des méthodes numériques et l'utilisation des ordinateurs changent la dimension des recherches. Néanmoins, les résultats demeurent modestes car la difficulté du problème avait été largement sous-estimée, en particulier en ce qui concerne la parole continue. Vers 1970, la nécessité de faire appel à des contraintes linguistiques dans le décodage automatique de phrases apparaît clairement, alors que la reconnaissance de la parole avait été jusque-là considérée comme un problème d'ingénierie. La fin de la décennie 1970 voit se terminer la première génération des systèmes commercialisés de reconnaissance de mots. Les générations suivantes, mettant à profit les possibilités sans cesse croissantes de la micro-informatique, possèderont des performances de plus en plus grandes (systèmes multi locuteurs, parole continue). On notera, que l'utilisation des modèles de Markov cachés dans la réalisation des systèmes de RAP représente un Tournon décisif dans la lignée des systèmes de RAP.

2.2.2 Applications

Si au départ, beaucoup des systèmes RAP étaient pour beaucoup basés sur des composants matériels, depuis les applications de la RAP bénéficient de l'évolution technologique qui se traduit par le fait qu'un système de reconnaissance complet peut désormais être entièrement implanté sous forme logicielle. Cette évolution a largement contribué au développement d'applications nouvelles à faible coût. Nous citons comme applications :

2.2.2.1 Saisie de données

Les machines à dicter peuvent être utilisées pour dicter des textes généraux (courrier, rapports...etc.) ou spécialisés (par exemple, des comptes rendus radiologiques ou autres faits par des médecins, comme dans le système vendu par Philips). Les dernières versions de ces machines sont bien intégrées dans le système d'exploitation du micro-ordinateur hôte, de sorte que des commandes au système peuvent être données oralement. Les performances, après une phase d'apprentissage sont très bonnes. A ce titre, on cite le logiciel « Dragon naturally speaking » que nous avons essayé dans le cadre de la langue française, et qui avec un ensemble d'apprentissage non contraignant offre de bonnes performances en dictée.

2.2.2.2 Aide aux handicaps

La reconnaissance de la parole intervient aussi comme aide pour certains handicaps. Elle permet ainsi à un handicapé moteur un contrôle efficace de son environnement. Ce créneau d'application encore limité, devrait se développer à l'avenir avec la diminution du coût des systèmes. Un autre domaine est celui de l'aide à l'apprentissage de la langue parlée pour un malentendant. L'idée est de compenser le manque de contre-réaction auditive chez le malentendant par une contre-réaction visuelle pour l'aider à prononcer les sons de parole et à acquérir la prosodie de la langue. Plusieurs systèmes de ce type ont été réalisés. Cette même idée a été reprise pour aider à l'apprentissage oral d'une langue étrangère. On notera que, les malvoyants sont peut-être la communauté qui a le plus bénéficié des avancés de la RAP et du traitement de la parole en général.

2.2.2.3 Commande de machine

La commande orale d'un appareil ou d'une machine a été une des premières applications de la reconnaissance automatique de la parole. Le contexte est en effet dans l'ensemble favorable :

- le vocabulaire est limité à quelques dizaines, ou parfois quelques centaines, de mots.
- les commandes sont composées soit de mots isolés ou enchaînés soit de phrases à structure simple et rigide.

2.2.2.4 Traduction parole-parole

La Traduction parole-parole (TPP) est un axe de recherche très prometteur qui pose plusieurs défis scientifiques importants. Le principe est de permettre à un locuteur de s'exprimer dans sa langue pour s'adresser à un interlocuteur ne parlant pas la même langue. Son message est reconnu, traduit et synthétisé (en un temps aussi proche que possible du temps réel) dans la langue de l'interlocuteur. Il est alors nécessaire non seulement de gérer les problèmes de reconnaissance et de synthèse de la parole, et du traitement du langage naturel, mais également ceux de la traduction automatique.

2.2.3 Concept de base

La démarche classique suivie lors du processus de reconnaissance automatique de la parole est illustrée par la figure 2.2, ce schéma fait ressortir les étapes principales dans un tel processus.

Ainsi, étant donné un signal en entrée du système, celui-ci va subir un pré-traitement qui consiste généralement en un filtrage et un échantillonnage qui permet de passer d'un signal continu à des valeurs discrètes, de ces valeurs dont le nombre est important seront extraites des caractéristiques qui permettent de représenter de façon compacte et perti-

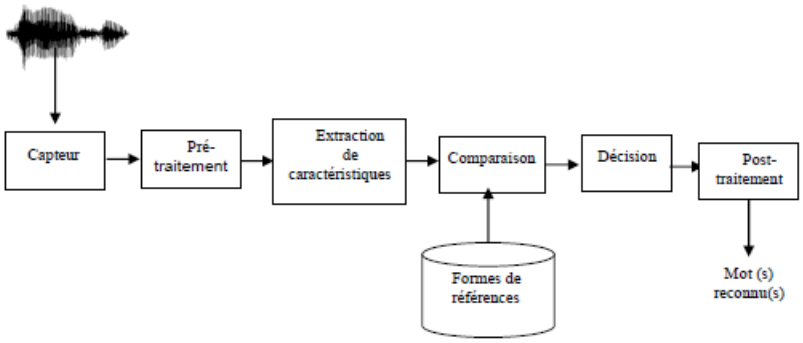

FIGURE *2.2* – Organigramme d'un système de reconnaissance automatique de la parole (d'après [3])

nente le signal originel. Cette étape permet d'avoir une première représentation du signal, ensuite et selon l'approche adoptée par le système de reconnaissance, ce modèle représentatif du signal sera comparé à des formes d'autres signaux que le système « connaît ». Sur la base du résultat de cette comparaison une décision quant au mot reconnu sera prise, celle-ci sera éventuellement validée en considérant les connaissances du domaine [3].

2.2.4 Analyse du signale de la parole

Le problème de la reconnaissance automatique de la parole consiste d'abord, à extraire l'information contenue dans un signal de parole, et qui nous oblige à faire appel au domaine du traitement du signal afin d'analyser et fournir une description détaillée sur le signal acoustique ainsi que réduire les redondances si elles existent, ou encore à extraire des paramètres pertinents pour la reconnaissance. La figure 2.3 montre un signal représentant la prononciation du mot /جَميل/.

En effet, le signal de la parole est porteur d'une grande redondance de l'information. Le traitement automatique de la parole nécessite de réduire cette redondance, à l'aide de traitements appropriés, pour diminuer les temps de traitement et l'encombrement en mémoire. Par ailleurs, et quelquefois simultanément, le traitement du signal vocal permet d'extraire des paramètres pertinents pour la reconnaissance (caractéristiques de sons bruités, fréquences des formants...etc.).

Les dispositifs utilisés peuvent être analogiques, cependant, avec l'évolution de l'électronique numérique et de l'informatique, les techniques numériques sont désormais généralisées. Après numérisation du signal vocal à l'aide d'un convertisseur analogique-numérique (CAN), les traitements sont alors effectués par logiciel soit par des composants spécialisés permettant de faire l'analyse de la parole en temps réel, soit de plus en plus par les puces de microprocesseurs. Les CAN permettent d'avoir à partir du signal continu, un ensemble de valeurs discrètes résultant du signal échantillonné.

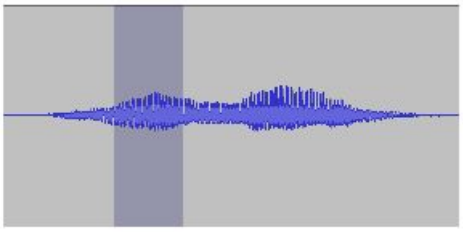

FIGURE *2.3* – Le signal du mot /جَميل/ (La partie grisée correspond à la voyelle /فَتحة/)

La quantité de points d'échantillonnage est extrêmement volumineuse, il est donc nécessaire de réduire ce nombre et d'éliminer la redondance. La trame acoustique est un ensemble de coefficients ou paramètres, calculés sur un bloc d'échantillons. Comme les techniques utilisées pour l'extraction de ces coefficients supposent que le signal sur lequel on opère est stationnaire, la plupart des algorithmes d'analyse opèrent donc sur un bloc d'échantillons de taille fixe dans lequel le signal est supposé stationnaire, il correspond à un temps de parole de 20 à 40 ms. La suite de vecteurs d'analyse est obtenue en déplaçant ce bloc de 10 à 20 ms ; il y a recouvrement de blocs [3].

Une fois le fenêtrage effectué, il est courant de pondérer ces fenêtres par des fonctions appropriées, on cite en particulier la fenêtre de Hamming ; très utilisée en traitement de la parole.

Nous avons précédemment souligné que le signal de la parole présente des particularités telles que la redondance qui justifient tout à fait la recherche d'une représentation plus compacte du signal. Pour cela, il existe deux grandes catégories pour l'extraction de caractéristiques d'un signal :

- les méthodes générales, valables pour tout signal évolutif dans le temps, en particulier les analyses spectrales.
- les méthodes se référant à un modèle de production du signal vocal ou d'audition.

2.2.4.1 Méthodes générales

Les méthodes spectrales occupent une place prépondérante en analyse de la parole, l'oreille effectue, entre autres, une analyse fréquentielle du signal qu'elle perçoit de plus, les sons de la parole peuvent être assez bien décrits en termes de fréquences. La transformée de Fourier est une technique très utilisée en traitement du signal qui vise à décomposer un signal en ses différentes composantes fréquentielles. La transformée de Fourier permet d'obtenir le spectre d'un signal en particulier son spectre fréquentiel, c'est-à-dire sa représentation amplitude-fréquence. La figure 2.4 montre de tels spectres pour une portion

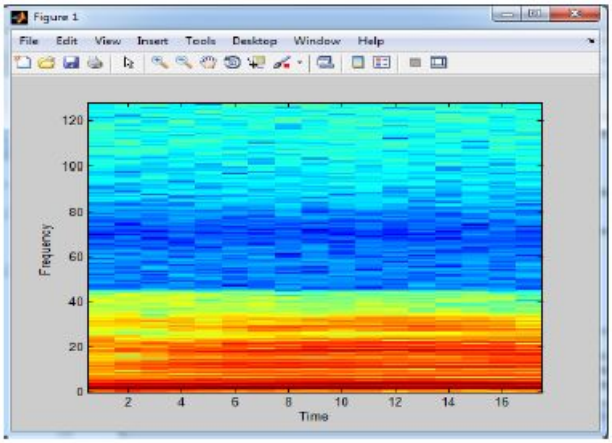

FIGURE *2.4* – Spectre obtenu par FFT de la voyelle /فَتحة/

du signal vocal pris dans les voyelles /فَتحة/. Ce spectre a été obtenu par un algorithme de transformation rapide de Fourier (FFT Fast Fourier Transform) permettant un calcul des coefficients du spectre en temps réel. Les maxima de ces spectres correspondent aux formants des voyelles. Les pics secondaires masquant en partie les formants sont dus au fondamental de la voix (fréquence de vibration des cordes vocales). La parole étant un phénomène non stationnaire, il importe de faire intervenir le temps comme troisième variable dans la représentation. Ainsi, dans l'exemple de la figure 2.4, le calcul du spectre a été effectué sur une tranche de signal limitée par une fenêtre temporelle (matérialisée sur la figure 2.3 par la zone grisée). La juxtaposition des spectres obtenus pour des tranches successives permet d'apprécier l'évolution du signal au cours du temps. Un analyseur spectral formé d'un banc de filtres passe-bande (par exemple, quinze filtres dans la bande de fréquences 100 à 5 000 Hz) fournit une approximation du spectre d'un signal vocal.

2.2.4.2 Méthodes avec modélisation

Dans cette catégorie, les méthodes dites de codage prédictif linéaire (LPC) ont été largement utilisées pour l'analyse de la parole. Elles font référence à un modèle du système de phonation, que l'on représente en général comme un tuyau sonore à section variable. L'idée sous-jacente revient à considérer que la valeur dû à l'ajustement des paramètres de ce modèle permet, en particulier, de déterminer à tout instant sa fonction de transfert, cette fonction fournit une approximation de l'enveloppe du spectre du signal à l'instant d'analyse, sur laquelle il est plus aisé de repérer les fréquences des formants car les pics secondaires dûs au fondamental de la voix présents dans le spectre de Fourier (figure 2.4) sont éliminés. La figure 2.5 montre un exemple de tels spectres lissés. L'analyse par codage prédictif est utilisée essentiellement en codage et en synthèse de la parole.

Les coefficients cepstraux sont obtenus en appliquant une transformée de Fourier nu-

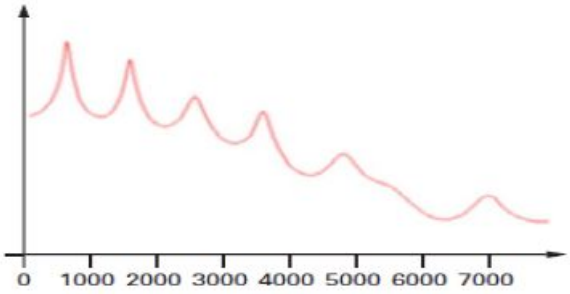

FIGURE *2.5* – Spectre lissé par LPC

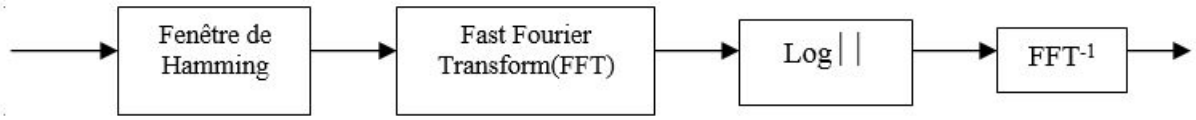

FIGURE *2.6* – Analyse cepstrale sur une fenêtre temporelle

mérique inverse au logarithme du spectre d'amplitude (figure 2.5). Le signal ainsi obtenu est représenté dans un domaine appelé cepstral ou quéfrentiel ; les échantillons se situant en basses quéfrences correspondent à la contribution du conduit vocal et donnent les paramètres utilisés en RAP, tandis que la contribution de la source n'apparaît qu'en hautes quéfrences.

Les deux familles de coefficients cepstraux les plus utilisées en RAP sont issues de deux analyses différentes pour obtenir le spectre. Lorsque le spectre d'amplitude résulte d'une FFT (voir figure 2.6) sur le signal de parole pré-traité, lissé par une suite de filtres triangulaires répartis selon l'échelle Mel, les coefficients sont appelés Mel Frequency Cepstral Coefficients (MFCC).

Lorsque le spectre correspond à une analyse LPC, les coefficients se déduisent des coefficients LPC par développement de Taylor, d'où leur nom de Linear Prediction Cepstral Coefficients (LPCC).

A l'issue de l'étape d'analyse du signal, on dispose pour chaque signal vocal en entrée d'un ensemble de vecteurs acoustiques en sortie. Ce sont ces vecteurs de caractéristiques qui seront utilisés dans la suite du processus de reconnaissance.

2.2.5 Reconnaissance de mot

L'absence dans le signal vocal d'indicateurs sur les frontières de phonèmes et de mots constitue une difficulté considérable. De ce fait, la reconnaissance de mots prononcés artificiellement de façon isolée représente une simplification notable du problème. La reconnaissance d'un mot est alors un problème typique de reconnaissance de formes. Tout système de reconnaissance de formes comporte les trois parties suivantes :

- un capteur permettant d'appréhender le phénomène physique considéré (dans notre cas un microphone).
- une étape de paramétrisation des formes (par exemple, un analyseur spectral).
- une étape de reconnaissance chargée de classer une forme inconnue dans l'une des catégories possibles.

En reconnaissance de la parole, il existe deux approches pour appréhender les phases de modélisation et de décision, que nous relions à l'étape de reconnaissance. On cite : l'approche globale et l'approche analytique.

2.2.5.1 Approche globale

Dans l'approche globale, la forme de base à reconnaître est le *mot* ; que l'on considère comme une entité non décomposable. Cette méthode fournit une image acoustique de chaque mot à identifier et permet donc d'éviter l'influence mutuelle des sons à l'intérieur des mots. Elle se limite aux petits vocabulaires : on peut citer des applications comme les systèmes de commande vocale. Ou par exemple, une application d'identification de code qui comprend uniquement les chiffres.

2.2.5.2 Approche analytique

Dans cette approche, la modélisation tire partie de la structure des mots, identifie les composantes élémentaires « phonèmes, syllabes, diphones...etc. ». Celles-ci sont les unités de base à reconnaître.

Cette approche est plus générale que la précédente, elle est préconisée pour reconnaître de grands vocabulaires, il suffit d'apprendre à la machine les principales caractéristiques des unités de base. Le plus souvent, l'unité de base retenue est le phonème.

Aussi bien pour l'approche globale que pour l'analytique la modélisation adoptée pour le mot ou le phonème, ce sont les modèles de Markov cachés.

2.2.6 Modèles de Markov cachés

La variabilité inhérente au signal de la parole (plus spécialement en contexte multi-locuteurs) peut être approchée par une modélisation stochastique, en particulier sous forme de modèles markoviens. Dans ces modèles, chaque entité à reconnaître est représentée par une source de Markov capable d'émettre le signal vocal correspondant à ce mot. Les méthodes stochastiques comptent parmi les méthodes les plus performantes actuellement disponibles. On les trouve en particulier dans la plupart des produits commercialisés.

La chaîne interne $X(t)$ est une chaîne de Markov que l'on suppose à chaque instant dans un état où la fonction aléatoire correspondante engendre un segment élémentaire (de l'ordre de $10ms$ ou plus), représenté par un vecteur de paramètres, de l'onde acoustique

observée. Un observateur extérieur ne peut voir que les sorties de ces fonctions aléatoires, sans avoir accès aux états de la chaîne sous-jacente, d'où le nom de modèle caché. En général. $X(t)$ est modélisé par une chaîne de Markov d'ordre 1 dont l'état à l'instant t ne dépend que de l'état à l'instant précédent $t-1$.

2.2.6.1 Les modèles de Markov cachés MMC

Une chaîne de Markov est de manière générale un processus de Markov à temps discret et à espace d'états discret. En mathématiques, un processus de Markov est un processus stochastique possédant la propriété de Markov, très utilisé car il permet de mettre en correspondance le phonème identifié et le langage proprement dit. Un HMM (Hidden Markov Model) est caractérisé par un double processus stochastique : un processus interne, non observable, $X(t)$ et un processus externe observable $Y(t)$. Ces deux chaînes se combinent pour former le processus stochastique.

Un HMM est caractérisé par le quintuplet $\{\pi_i, S, X, A, B\}$ où :

- π_i est la distribution de la probabilité de l'état initial.
- S est l'ensemble d'états $S = \{S_1, S_2, \dots S_N\}$.
- X est l'ensemble des observations ou les symboles à émettre par les états $X = \{X_1, X_2, \dots X_M\}$.
- A est la probabilité de déplacement d'un état à un autre $S_i \rightarrow S_j$ avec $A_{i,j} = P(S_j|S_i)$.
- B est la fonction de distribution de probabilité des observations X_k pour l'état j à l'instant t avec $B_t(k) = P(X_k|S_j)$.

2.2.6.2 Apprentissage d'un HMM

Un des grands intérêts des HMM réside dans l'automatisation de l'apprentissage des différents paramètres et distributions de probabilités du modèle à partir de données acoustiques représentatives de l'application considérée, essentiellement les probabilités de transition d'un état du HMM à un autre état et surtout les lois d'émission B $b_i(o)$ avec $b_i(o)$ la probabilité d'émettre une certaine observation o, sachant que le processus markovien est dans l'état i. Ces probabilités sont en général représentées sous forme d'une somme de fonctions gaussiennes (parfois plusieurs dizaines, permettant de mieux approcher la loi réelle du phénomène), comme l'illustre la figure 2.7.

Cet apprentissage est assuré par des algorithmes itératifs d'estimation des paramètres, notamment l'algorithme de Baum-Welch, cas particulier de l'algorithme EM (Expectation-Maximisation) fondé sur le principe de maximum de vraisemblance. La taille du corpus de données nécessaires pour converger vers une valeur convenable des paramètres est très grande. Pour des applications de grande envergure (parole continue et très grands

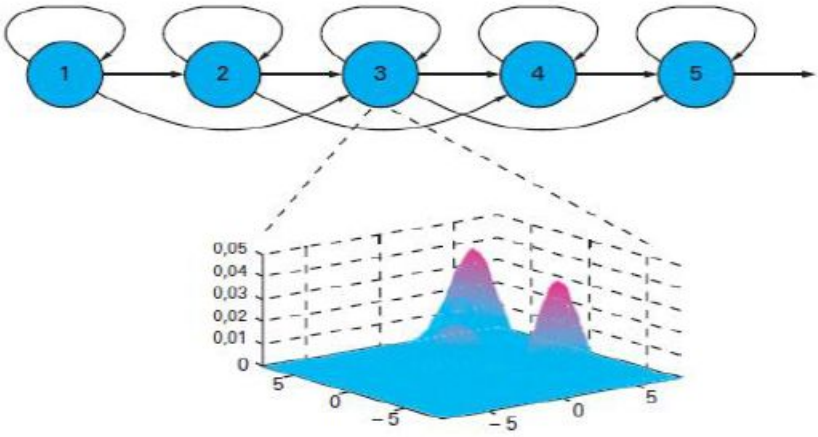

FIGURE *2.7* – Modèle de Markov à cinq états

vocabulaires par exemple), il faut disposer de centaines d'heures de parole étiquetée phonétiquement pour obtenir des modèles de qualité. Il est également possible de réaliser un apprentissage initial assez complet puis d'effectuer un simple ajustement des paramètres (adaptation) en fonction d'une utilisation particulière : locuteur, vocabulaire, conditions d'enregistrement.

2.2.6.3 Mise en œuvre d'un système de reconnaissance à HMM

Les HMM peuvent être utilisés de plusieurs façons en RAP, selon l'importance de l'application (taille du vocabulaire et type de parole : mots isolés ou parole continue). Pour la reconnaissance de mots isolés avec de petits vocabulaires (environ 100 mots au plus), il est possible de modéliser chaque mot par un HMM, comme celui de la figure 2.7.

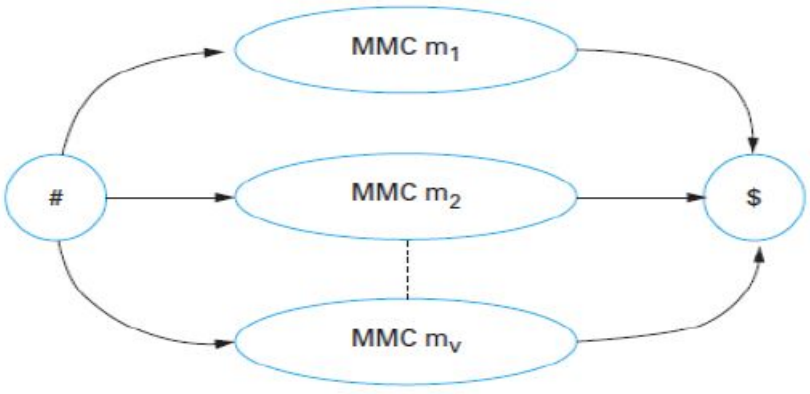

FIGURE *2.8* – Principe de la reconnaissance de mots isolés par HMM

La reconnaissance revient alors à calculer la vraisemblance de la suite d'observations acoustiques constituant le mot à reconnaître par rapport à chacun des modèles de mots.

Le modèle présentant la plus grande vraisemblance d'avoir émis cette suite d'observations fournit le mot reconnu. L'algorithme permettant d'optimiser ce calcul est à nouveau utilisé mais dans un cadre stochastique, l'algorithme de Viterbi. La figure 2.8 donne le schéma général d'un tel système.

Pour la reconnaissance de grands vocabulaires ou celle de la parole continue, l'utilisation de modèles globaux pour chaque mot pose divers problèmes : espace mémoire de stockage, volume de données acoustiques nécessaires pour l'apprentissage de tous les HMM...etc.

La solution adoptée est d'utiliser des HMM pour représenter les unités phonétiques. Ces unités peuvent être de nature variée : phonèmes, di- phones, syllabe, état acoustique élémentaire, allophones (variantes d'un phonème dans différents contextes)...etc.

Les modèles de mots sont construits par concaténation des modèles analytiques élémentaires correspondant aux transcriptions phonétiques de ces mots. Il est alors possible de tenir compte de variantes de prononciation. On introduit aussi des connaissances phonologiques pour prendre en compte les phénomènes inter-mots (liaisons...etc.).

Pour mettre au point des HMM aussi indépendants du locuteur que possible, il est nécessaire d'augmenter le nombre de paramètres des HMM. Les solutions disponibles sont de deux types :

- les multi modèles, le principe est de représenter le même mot par plusieurs HMM correspondant par exemple à différentes classes de locuteurs obtenues par une méthode de classification automatique.
- les mélanges de densités de probabilité, au lieu de représenter la probabilité d'émission d'un segment de parole pour une loi de probabilité (en l'occurrence une gaussienne), on utilise un mélange de lois gaussiennes (parfois plusieurs centaines dans chaque état des HMM) permettant de mieux approcher la loi réelle du phénomène.

2.3 Partie 2 : Le moteur de reconnaissance automatique de la parole Sphinx

2.3.1 Introduction au moteur de reconnaissance Sphinx

La disponibilité des outils de reconnaissance de la parole avec leur architecture modulaire et leur open source permet aux chercheurs et programmeurs d'implémenter et tester de nouveaux algorithmes. Ces caractéristiques facilitent le développement de puissants systèmes de reconnaissance automatique de la parole (RAP) [4].

En pratique, différentes boite à outils ont été utilisées pour la création des systèmes de reconnaissance vocale sur n'importe quelle langage, on cite : *HTK*, *Sphinx*, *Kaldi*, *ASR de Matlab* et *java speech API*...etc. On notera toutefois, que souvent les systèmes disponibles sont dédiés à une tâche spécifique et ne permettent pas beaucoup de modifications de la part des développeurs. De plus, nombre de ces outils sont payants et il n'est pas toujours facile de les acquérir pour les chercheurs et les programmeurs.

Comme précédemment mentionné, il existe plusieurs librairies pour le domaine de RAP, on va s'intéresser en particulier au trois (3) librairies les plus utilisés : *HTK* et *Sphinx* et la *Toolbox de Matlab.*

2.3.1.1 Sphinx

Sphinx est une librairie de reconnaissance vocale gratuitement téléchargeable, avec la possibilité de modifier le code source, il a la capacité d'implémenter des systèmes avec un large vocabulaire, indépendant du locuteur. Les premières versions du sphinx (1,2 et 3) sont écrites en langage C, mais la version récente "Sphinx4" est codée en java. Pour plus de détails voir le tutoriel de Sphinx [5]. Sphinx 1, 2, 3 et 4 sont des décodeurs, quand à l'outil d'apprentissage il s'appelle *SphinxTrain.* Sphinx4 [5] est très souple dans sa configuration, avec une architecture modulaire qui permet aux programmeurs et chercheurs de tester de nouveaux algorithmes. La figure 2.9 représente l'architecture du Sphinx 4 qui s'articule autour de trois modules principaux [6].

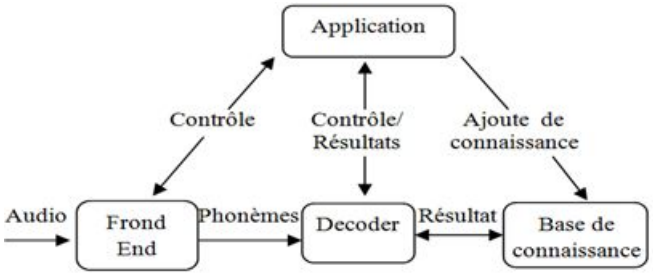

FIGURE *2.9* – Architecture d'application du Sphinx 4, d'après [6]

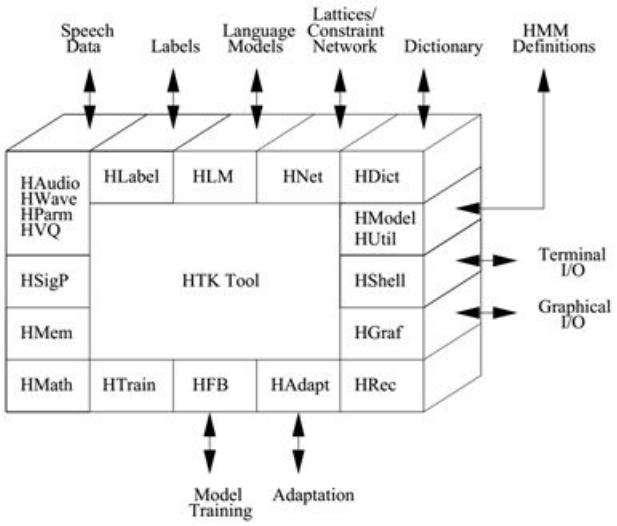

FIGURE *2.10* – Architecture de HTK, d'après [8]

2.3.1.2 HTK

Hidden Markov Model Toolbox est la traduction de HTK développée par l'université de Cambridge. Cette boite à outils dédiée aux Modèles de Markov Cachés est principalement utilisée pour la reconnaissance de la parole. Elle se compose d'un ensemble de modules et d'outils disponibles gratuitement et téléchargeable à partir du site. HTK est implémenté en langage C et il exécuté en ligne de commande, il est capable de mettre en œuvre un grand vocabulaire, indépendamment du locuteur et est applicable sur n'importe quelle langue. La documentation sur HTK est très riche avec des exemples pratiques (vers 300 pages). Pour plus de détails voir [7]. La figure 2.10 montre l'architecture de l'outil HTK [8].

2.3.1.3 Matlab

Matlab comprend une boite à outil (*toolbox*) incluant des algorithmes d'apprentissage artificiel basé sur les modèles de Markov cachés et des algorithmes de détection des séquences temporelles hors ligne et en ligne.

2.3.1.4 Comparaison

Au sein de notre équipe du laboratoire, des travaux récents ont permis une étude comparative entre les librairies HTK, Sphinx et Matlab [4]. Le tableau 2.1 montre les différences entre deux de ces librairies qui sont les plus répandues.

Des expériences ont montré que la qualité des modèles acoustiques formés par Sphinx a été jugée meilleure que celle par HTK [9], si on ne prend en considération que la base TIMIT.

Le tableau comparatif 2.2 est inspiré d'une expérience de l'institut HPI (Hasso Plattner

TABLE *2.1* – Tableau comparatif (HTK *Vs.* Sphinx) d'après [4]

Critères	HTK	Sphinx
Structure	Un système complexe	Un système bien structuré (modulaire)
Mis à jour	Mis à jour régulièrement	Mis à jour régulièrement
Documentation	Bien documenté à la fois théorique et pratique	La documentation de ce système est relativement pauvre
Langage	Langage C	Langage C et java
Traitement du signal	Banque de filtre, MFCC, LPC, PLP, LPreflexC, ClpC, IREFC, et MELSPEC	MFCC, PLP, spectre
Formats d'entrée	WAV, TIMIT, NIST, OIG, AIFI	WAV, SPHEG, MSWAV
Apprentissage	HRest/HERest	SphinxTrain
Décodeur	HVite	Sphinx 4

TABLE *2.2* – Comparaison entre Sphinx et HTK d'après [10]

Critère	Sphinx 4	HTK
Taux de reconnaissance (60% de score total)	6.5	6
Indépendance (15%)	8	8
Coût (5%)	10	7
Modularité (15%)	10	0
Actualité (5%)	7	6
Score totale	7.45	6.35

Institut) de l'Université de Potsdam [10] qui montre que les performances de Sphinx4 sont supérieures à celles de HTK.

Au vu des recherches que nous avons effectué au début de notre travail de thèse, notre choix s'est porté sur Sphinx comme outil de développement de notre système de reconnaissance de la parole et ultérieurement du système d'évaluation de la prononciation.

2.3.2 Le moteur de reconnaissance Sphinx

Pour aller dans le sens de l'innovation dans le domaine de la reconnaissance automatique de la parole, le moteur de la reconnaissance de la parole Sphinx a été créé. Il s'agit d'une plateforme open source qui intègre des méthodes de l'état de l'art et qui prend en charge les besoins des domaines de recherche émergeants.

Sphinx est un Framework modulaire et portable qui inclut une conception de formes

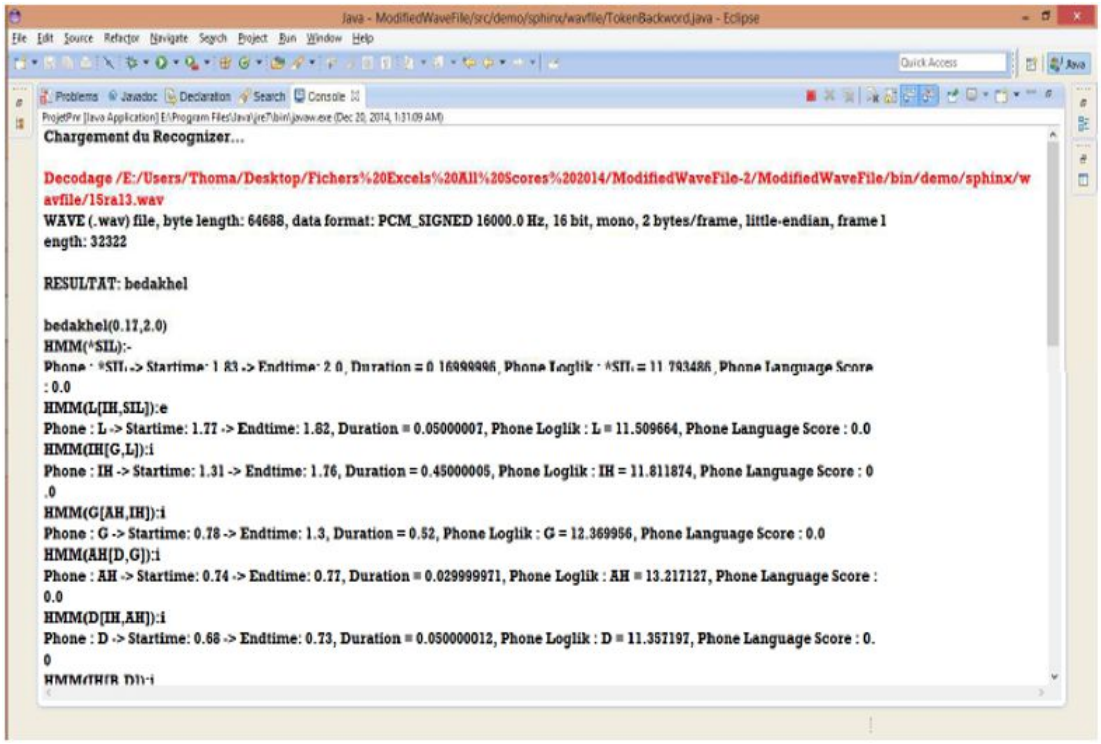

FIGURE *2.11* – Interface de Sphinx

à partir des systèmes existants, avec une flexibilité assez suffisante pour supporter des domaines émergeants dans un cadre de recherche. Il contient des composants séparables dédiés à des tâches spécifiques et aussi différents modules qui implémentent les techniques de l'état de l'art en reconnaissance automatique de la parole. La figure 2.11 montre un exemple d'exécution d'un processus d'évaluation de la prononciation en Arabe en utilisant la librairie Sphinx.

L'approche traditionnelle pour la conception d'un système de reconnaissance automatique de la parole avait comme but de créer un système complet et optimisé autour d'une méthodologie particulière. Comme montré par les anciens systèmes de recherche sur la reconnaissance automatique de la parole, tel que [11] et [12], cette approche a prouvé son importance.

Chacun des systèmes précédemment mentionnés a été dédié pour explorer un domaine innovant spécifique de la reconnaissance de la parole. Par exemple, Baker a introduit les HMMs dans son système Dragon [11] [13], les derniers prédécesseurs de Sphinx ont explorés les variantes des HMMs comme les HMMs discrets [14], les HMMs semi continus [15] et les HMMs continus [16]. Les autres systèmes ont exploré des stratégies de recherche spécifiques comme l'utilisation des arbres de décision pour des modèles large N-Gram [17].

Vu qu'ils étaient focalisés sur de telles théories fondamentales, les créateurs de ces systèmes ont tenté de programmer leurs implémentations à un niveau très élevé. Par exemple,

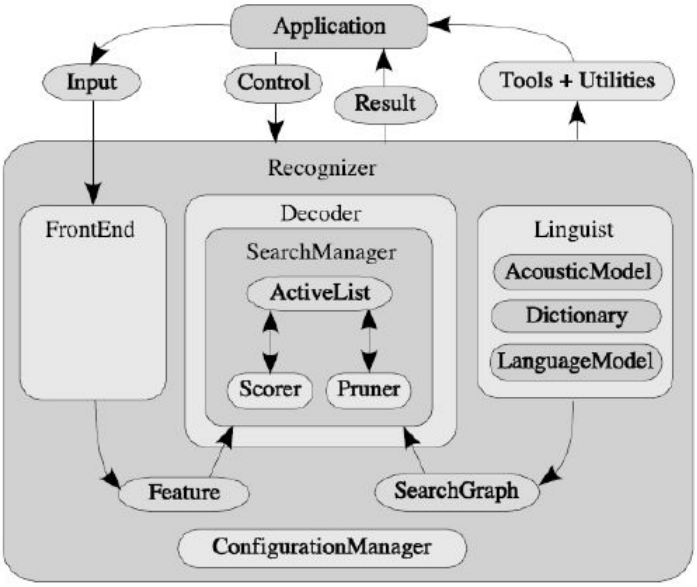

FIGURE *2.12* – L'architecture générale du système Sphinx, d'après [18]

les versions anciennes de Sphinx limitent l'ordre des HMMs à une valeur constante et fixent l'unité en contexte à un seul contexte gauche-droite. Ces versions ont éliminé aussi la prise en charge des grammaires à contexte libre (CFGs ; Context Free Grammars) due à la spécialisation sur des modèles large N-Gram. De plus, la stratégie de décodage de ces systèmes a une tendance à être profondément désordonnée avec le reste du système. Suite à ces contraintes, et comme résultat, les moteurs de reconnaissance automatique de la parole open source existants étaient difficiles à modifier pour des expérimentations spécifiques. C'est pour ces raisons que nous avions choisi d'effectuer nos expérimentations sur la version la plus récente de sphinx : Sphinx 4 dont nous allons détailler le mode de fonctionnement dans ce qui suit. Il faut noter aussi que, vu la performance de ce système ainsi que sa rapidité, le module dédié à la création des modèles acoustiques a été exclu de l'implémentation de Sphinx 4, et pour cette raison, les développeurs doivent effectuer la phase d'apprentissage en Sphinx 3. Dans notre cas, les expérimentations et les tests effectués durant le projet de thèse étaient sur la version sphinx 4, alors que la construction des modèles acoustiques et du modèle linguistique s'est faite sous Sphinx 3.

2.3.2.1 La plateforme Sphinx 4

Comme précédemment mentionné, le Framework sphinx 4 a été conçu avec un degré élevé de flexibilité et de portabilité. La figure 2.12 montre l'architecture générale de Sphinx 4.

Chaque entité dans l'architecture de Sphinx 4 représente un module qui peut être facilement remplacé, permettant ainsi aux chercheurs de faire des expérimentations et des modifications sans avoir besoin de modifier le reste du système.

Comme le montre la figure 2.12, il y a trois modules primaires dans Sphinx 4, le front-end, le décodeur (*Decoder*) et le linguiste (*Linguist*). Le module front-end prend un ou plusieurs signaux en entrée pour les paramétrer en une séquence de caractéristiques (*Feature*). Le module linguistique traduit n'importe quel type de modèle de langage standard ainsi que des informations concernant la prononciation à partir du dictionnaire (*Dictionary*) et des informations structurales à partir d'un ou plusieurs ensembles de modèles acoustiques (*AcousticModels*) en un graphe de recherche (*SearchGraph*). Le gestionnaire de recherche (*SearchManager*) dans le module décodeur utilise les caractéristiques à partir du module front-end et le graphe de recherche à partir du module linguistique pour effectuer le décodage du signal de la parole et générer des résultats (*Results*). À n'importe quel moment, durant le processus de la reconnaissance automatique de la parole, l'application peut envoyer des requêtes de contrôles (*Controls*) à chacun des modules et devenir un partenaire durant le processus de la reconnaissance.

Sphinx 4 est comme la plupart des systèmes de reconnaissance automatique de la parole et c'est pour cela qu'il dispose d'un grand nombre de paramètres configurables, comme le nombre de transitions dans le graphe de recherche, pour ajuster les performances du système. Le gestionnaire de configuration de Sphinx 4 est utilisé pour configurer de tels paramètres. Contrairement à d'autres systèmes, le gestionnaire de configuration donne à Sphinx 4 la possibilité de configurer et de charger dynamiquement les modules pendant l'exécution rendant ainsi le système flexible. Par exemple, Sphinx 4 est configuré avec le module *front-end* qui produit des vecteurs de caractéristiques de type MFCC (Mel Frequeny Cepstral Coefficients). En utilisant le gestionnaire de configuration, il est possible de configurer Sphinx 4 pour construire un front-end diffèrent qui produit des caractéristiques de type PLP (Perceptual Linear Preediction) sans avoir besoin de modifier le code source ou de recompiler le système pour prendre en charge le changement de paramètres.

Afin de donner aux applications et aux développeurs la possibilité de suivre les statistiques du module décodeur, comme par exemple le taux d'erreur de la reconnaissance, la vitesse d'exécution et les informations concernant l'usage de la mémoire, Sphinx 4 fournit un certain nombre d'outils. Comme le reste du système, ces outils sont configurables permettant aux utilisateurs d'effectuer une analyse sophistiquée du système. En outre, ces outils fournissent aussi un environnement d'exécution interactif qui permet aux utilisateurs de modifier les paramètres du système durant son exécution, permettant une expérimentation plus rapide avec plusieurs paramètres.

Sphinx 4 fournie aussi un support pour le module utilités (*Utilities*) qui prend en charge un traitement à un niveau applicatif des résultats de la reconnaissance. Par exemple, ces utilités incluent des supports pour obtenir des résultats sous forme de treillis, des scores de confiances et la compréhension du langage naturel.

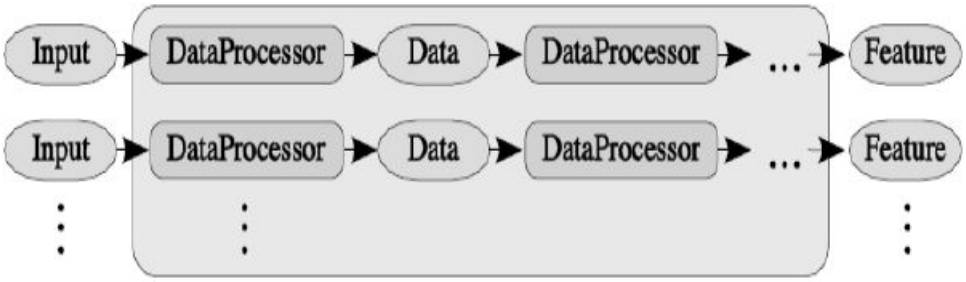

FIGURE *2.13* – Les chaines de processeurs de données en parallèles d'après [18]

1) Le Front-End

Le but de ce module est de paramétrer le signal de la parole en entrée en une séquence de caractéristiques. Comme montré dans la figure 2.13, ce module contient une ou plusieurs chaines en parallèles de modules de traitement de signal communicants remplaçables nommés les processeurs de données (*DataProcessors*).

Supporter plusieurs chaines permet au système de faire des calculs simultanés de différents types de paramètres à partir des mêmes ou différents signaux en entrée. Cela permet aussi la création des systèmes qui peuvent, simultanément, effectuer le décodage du signal en utilisant différents types de paramètres, comme les MFCC et les PLP, et même aussi les types de paramètres dérivés à partir des signaux qui ne contenaient pas de la parole comme les vidéo.

Comme le système ISIP [19], chaque processeur de données fournie une entrée et une sortie qui peuvent être connectées à un autre processeur de données, permettant ainsi une longue séquence de chaines arbitraires. Les entrés et les sorties de chaque processeur de données sont des objets de données (Data) génériques qui encapsulent aussi bien les données d'entrée traitées que les marqueurs qui indiquent les évènements de classification de données comme la détection de la fin du processus (*end-point*). Le dernier processeur de données dans chaque chaine est le module responsable de la production des objets de données composés de signaux, nommés caractéristique (*Feature*), utilisés par le décodeur.

2) Le module linguistique

Le module linguistique dans tout moteur de reconnaissance est indispensable, il donne la probabilité d'une phrase dans la langue. Ceci est fait en général de manière très simplifiée, dans le cadre d'une hypothèse markovienne d'ordre n : la probabilité de la phrase est le produit des probabilités de chacun des mots de la phrase sachant les mots précédents, en se restreignant à un passé de quelques mots. Ces probabilités sont estimées par comptage sur de grandes quantités de textes de référence (par exemple plusieurs années d'archives de journaux contenant des centaines de millions de mots...), en se limitant à un passé de deux ou trois mots.

Ce module dans Sphinx génère un graphe de recherche qui est utilisé par le décodeur durant la recherche. Vu que la procédure de la génération de ce type de graphe est compliquée, elle sera cachée par ce module. Comme c'est le cas à travers Sphinx 4, le module linguistique est un module portable permettant une configuration dynamique de systèmes avec différentes implémentations linguistiques possibles.

Une implémentation linguistique construit un graphe de recherche en utilisant une structure de langage comme une représentation par un modèle de langage (*LanguageModel*) et la structure topologique du modèle acoustique (*AcousticModel*) (les HMMs pour les unités de base tels les phonèmes). Ce module peut utiliser aussi un dictionnaire de prononciation pour mapper les mots à partir d'un modèle de langage en séquences d'éléments acoustiques (le modèle acoustique). Quand le graphe de recherche est généré, le module linguistique peut aussi incorporer des unités semi-mot en contexte d'une taille arbitraire. Le module linguistique contient trois composantes principales : un modèle de langage, un dictionnaire et un modèle acoustique. Nous donnons plus de détails concernant ces trois composantes dans ce qui suit.

A. Le modèle de langage

La composante modèle de langage fournit une structure linguistique au niveau mot, qui peut être représentée par n'importe quel nombre d'implémentations. Ces implémentations trouvent leur catégories dans : des grammaires dérivés-graphe et des modèles stochastiques N-Gram. Une grammaire dérivée-graphe représente un graphe de mots orienté, où chaque nœud représente un seul mot et chaque lien représente la probabilité du mot que la transition a eu lieu. Les modèles stochastiques N-Gram génèrent des probabilités pour des mots selon l'observation des $n-1$ mots précédents.

Les implémentations des modèles de langage dans Sphinx 4 revêtent une variété de formats :

- **SimpleWordListGrammar** : définit une grammaire basée sur une liste de mots. Un paramètre optionnel définit quand est ce que la grammaire est en boucle ou non. Si la grammaire n'est pas en boucle, alors elle sera utilisée pour la reconnaissance de mots isolés. Si la grammaire est en boucle, alors elle sera utilisée pour supporter la reconnaissance des mots connectés, ce qui est l'équivalant d'une grammaire *unigram* avec des probabilités égales.
- **JSFGrammar** : supporte le format de grammaire de l'API JAVA Speech.
- **LMGrammar** : définit une grammaire basée sur un modèle de langage statistique. Ce format génère un nœud par mot et fonctionne bien sur de petites grammaires *unigram* et *bigram*, avec une taille approximative de 1000 mots.
- **FSTGrammar** : supporte des grammaires de type FST (Finite-State Transducer)

dans le format de grammaire ARPA FST [20].

- **SimpleNGramModel** : supporte les modèles *N-Gram ASCII* dans le format ARPA. Ce format est connu pour être gourmand en mémoire, pour cela il marche bien avec des modèles de langage de petite taille.
- **LargeTrigramModel** : supporte les modèles *N-Gram* générés et fournit par l'outil CMU-CSLMT (CMU-Cambridge Statistical Language Modeling Toolkit) [21]. Contrairement au format *SimpleNGramModel*, ce format n'est pas gourmand en espace mémoire permettant ainsi le traitement des modèles de langage de grande taille.

B. Le dictionnaire

Le dictionnaire fournit les prononciations des mots lus à partir du modèle de langage. Les prononciations découpent les mots en une séquence de phonèmes qui sera lue à partir du modèle acoustique. Le dictionnaire supporte aussi la classification des mots et permet à un mot d'appartenir à plusieurs classes.

Sphinx 4 fournit des implémentations de l'interface dictionnaire. Ces implémentations optimisent l'usage de formes basées sur la taille du vocabulaire actif. Par exemple, une implémentation chargera le vocabulaire entier au moment de l'initialisation du système, tandis qu'une autre implémentation va seulement obtenir les prononciations à la demande.

C. Le modèle acoustique

Le module responsable de la génération du modèle acoustique (*AcousticModel*) fournit un mapping entre une unité de parole et un modèle HMM à qui un score est attribué face aux caractéristiques fournies par le module front-end. Comme avec les autres systèmes, le mapping peut prendre en considération les informations concernant le contexte et la position du mot. Par exemple, dans le cas où le mot est modélisé en *triphones*, le contexte représente les seuls phonèmes à droite et à gauche d'un phonème donné, et la position du mot représente la position du *triphone* que ce soit au début, milieu ou à la fin du mot (ou le mot lui-même). La définition contextuelle n'est pas fixée par Sphinx 4, permettant ainsi la création des modèles acoustiques qui contiennent tous les phonèmes et les modèles acoustiques dont leurs contextes n'ont pas besoin d'être adjacents à l'unité.

Comme précédemment mentionné, le module linguistique découpe chaque mot dans le vocabulaire correspondant en une séquence d'unité de parole (unités syllabiques, plus souvent phonétiques) dépendantes du contexte (*context-dependent*). Le module linguistique ensuite passe ces unités ainsi que leur contexte au module *AcousticModel*, récupérant ainsi les graphes HMM associés avec ces unités. Ensuite, il utilise ces graphes HMM en conjonction avec le modèle de langage pour construire le graphe de recherche *SearchGraph*.

Contrairement aux autres systèmes de reconnaissance automatique de la parole connus,

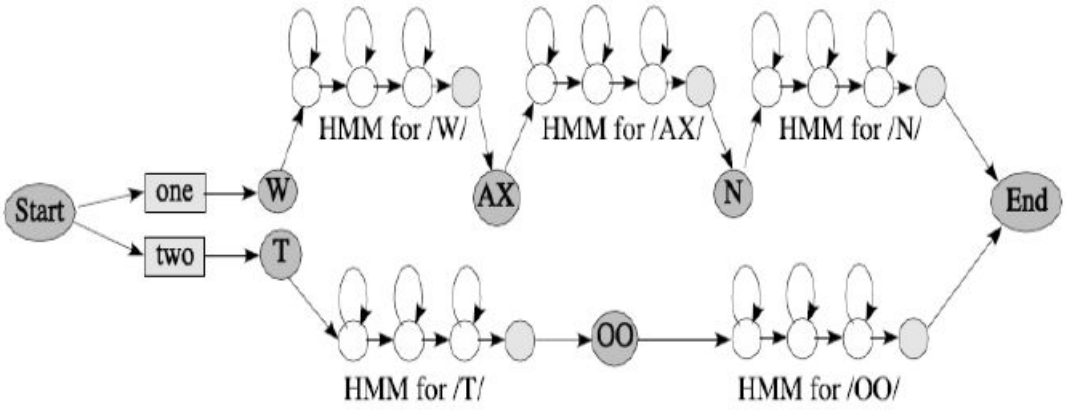

FIGURE *2.14* – Exemple d'un graphe de recherche, d'après [18]

qui représentent les graphes HMM comme étant une structure fixe dans la mémoire, l'implémentation des HMMs dans Sphinx 4 est un graphe d'objets orienté. Dans ce graphe, chaque nœud correspond à un état HMM et chaque lien représente une probabilité de transition d'un état à un autre dans le modèle HMM. En représentant le modèle HMM comme un graphe d'objets orienté au lieu d'une structure fixe, une implémentation du module *AcousticModel* peut facilement fournir des HMMs avec des topologies différentes.

Chaque état HMM est capable de produire un score à partir d'un ensemble de caractéristiques observées. Le calcul de ce score est effectué par l'état HMM lui-même. Le module *AcousticModel* permet aussi le partage de plusieurs composantes à tous les niveaux. Ces composantes peuvent être des mixtures gaussiennes, des matrices de transitions ou des poids attribués aux mixtures. Ces composantes peuvent être partagées par n'importe quel état HMM vers un niveau plus profond.

3) Le graphe de recherche (SearchGraph)

Même si le module linguistique peut être implémenté de différentes manières et la topologie de l'espace de recherche générée par ce module peut varier beaucoup, l'espace de recherche est représenté par un graphe de recherche (SearchGraph). La figure 2.14 montre la structure générale d'un graphe de recherche pour la reconnaissance automatique de la parole en utilisant Sphinx 4.

La figure 2.14 montre un graphe orienté dans lequel chaque nœud, nommé un état de recherche (*SearchState*) représente un état émetteur ou un état non-émetteur. Des scores peuvent être attribués aux états émetteurs lors de la réception des caractéristiques acoustiques, tandis que les états non-émetteurs sont généralement utilisés pour représenter des composantes de haut niveau linguistique tel que les mots et les phonèmes dont les scores ne sont pas directement attribués lors de la réception des caractéristiques. Les liens entre les états représentent les transitions possibles qui peuvent avoir lieu. Chacune de ces transitions est représentée par la probabilité de vraisemblance du passage d'un état à

un autre.

Construire le graphe illustré dans la figure 2.14 nécessite une connaissance à partir de plusieurs ressources. Il requière un dictionnaire qui doit mapper le mot « *one* » aux phonèmes : *W, AX et N*, et le mot « *two* » aux *T* et *OO*. Il requière aussi un modèle acoustique pour obtenir les HMMs pour les phonèmes *W, AX, N, T* et *OO*. Le graphe de recherche dispose d'une information concernant comment certains mots seront représentés. Cette information est généralement fournie par le modèle de langage.

Supposant que, dans l'exemple illustré dans la figure 2.14, la probabilité qu'une personne prononce le mot « *one* » (par exemple $p = 0.8$) est supérieure à la probabilité de prononcer le mot « *two* » (par exemple $p = 0.2$). Alors dans le graphe ci-dessus, la probabilité de transition entre le nœud en entrée et le premier nœud du HMM qui représente le phonème *W* sera 0.8, tandis que la probabilité de transition entre le nœud en entrée et le premier nœud du HMM qui représente le phonème *T* sera 0.2. Par conséquent, le chemin vers le mot « *one* » va avoir un score élevé.

Une fois le graphe de recherche est construit, la séquence des signaux de la parole paramétrés sera mappée aux différents chemins à travers le graphe pour trouver le meilleur résultat. Le meilleur résultat est généralement le chemin le moins coûteux ou le chemin qui correspond au score le plus élevé. Comme illustré dans la figure 2.14, plusieurs nœuds ont des transitions (parfois en boucle). Cela pourra mener à un grand nombre de chemins possibles à travers le graphe. Donc, trouver le meilleur chemin possible est gourmand en temps et espace mémoire. Pour cela plusieurs techniques d'élagage sont fournies comme alternative. Nous donnons plus de détails dans ce qui suit.

4) Le décodeur

Le rôle principal du décodeur est d'utiliser les caractéristiques à partir du front-end avec l'aide du graphe de recherche pour générer des hypothèses (*Result*). Le fonctionnement du décodeur est basé principalement sur un gestionnaire de recherche (*SearchManager*) et d'autres composantes dont le but est de simplifier le processus de décodage pour une application.

À chaque étape du processus de décodage, le gestionnaire de recherche crée un objet *Result* qui contient tous les chemins dont un état final non-émetteur est atteint. Pour traiter le résultat, Sphinx fournit des outils capables de produire des treillis et des scores de confiances à partir de l'objet *Result*.

Chaque implémentation du gestionnaire de recherche utilise un algorithme de jeton (token passing algorithm) [22] qui représente un objet associé à un état de recherche. Ce jeton contient des scores globaux acoustiques et linguistiques du chemin à un point donné, une référence à un état de recherche, une référence à une trame de caractéristiques en entrée et autres informations importantes. La référence à un état de recherche permet au

gestionnaire de recherche d'attacher un jeton à son état de sortie, à une unité phonétique dépendante du contexte, à une prononciation, à un mot, et un état de grammaire. Chaque hypothèse partielle se termine dans un jeton actif.

Comme illustré dans la figure 2.12, les implémentations d'un gestionnaire de recherche peut construire un ensemble de jetons actifs dans une forme de liste (*ActiveList*) à chaque étape du processus du décodage.

Le gestionnaire de recherche génère des listes de jetons à partir des jetons actifs dans le treillis de recherche en appliquant une stratégie d'élagage implémentée dans le sous module de l'élagage (*Pruner*). Le gestionnaire de recherche communique aussi avec le sous module générateur de score (*Scorer*). Il s'agit d'un module d'estimation de probabilité d'un état donné qui fournit la valeur de densité de sortie de cet état en demande. Quand le gestionnaire de recherche demande un score pour un état donné à un instant donné, le sous module *Scorer* accède au vecteur de caractéristiques à cet instant et effectue les opérations mathématiques pour calculer le score. Dans le cas d'un décodage parallèle en utilisant des modèles acoustiques parallèles, le module *Scorer* aligne l'ensemble des modèles acoustiques pour être utilisé selon le type de caractéristiques.

Le module générateur de score retient toute information qui a un rapport avec les densités de sortie d'un état donné. Pour cela, le gestionnaire de recherche n'a pas besoin de savoir quand est ce que la génération de score est effectuée en appliquant des HMMs discrets, semi-continus ou continus. En outre, la fonction de densité de probabilité de chaque état HMM est isolée selon la même manière.

2.3.2.2 L'adaptation de Sphinx à la langue Arabe

Pour pouvoir utiliser la librairie pour construire notre système d'évaluation de la prononciation Arabe, plusieurs adaptations et étapes sont requises. Il faut noter que même si le domaine de l'évaluation de la prononciation existe depuis longtemps (théoriquement), le moteur de reconnaissance automatique de la parole Sphinx ne prend pas en charge cette technique. Pour cela, nous avons implémenté un module supplémentaire, coopérant avec le moteur de reconnaissance Sphinx, pour permettre une reconnaissance et une évaluation automatique de la prononciation de la langue Arabe dans un contexte d'apprentissage de prononciation.

2.4 Partie 3 : Considération pratiques

2.4.1 La construction du modèle de langage

Il existe deux types de modèles qui décrivent une langue quelconque ; les grammaires et les modèles statistiques de langage. Les grammaires décrivent des types de langage « *simples* » pour des commandes de contrôle et elles peuvent être générées automatiquement ou manuellement.

D'autre part, il existe plusieurs façons de créer un modèle statistique de langage. Quand l'ensemble de données est suffisamment large, le toolkit de modélisation de langage de CMU (pour Carnegie Mellon University) est le plus recommandé. Quand l'ensemble de données est petit, un service web plus rapide fera l'affaire. Pour ce qui est pratique touchant les contributions de cette thèse, et pour la création d'un modèle de langage nous avons utilisé l'outil *CMULMTK* (pour Carnegie statistical Language Modeling Toolkit). Nous donnerons plus de détails concernant cela dans la section 2.4.1.2.

2.4.1.1 La construction d'une grammaire

Comme précédemment cité, les grammaires sont souvent générées manuellement sous le format *JSGF* (Java Speech Grammar Format) ; *JSGF* est une représentation textuelle des grammaires pour une utilisation dans le contexte de la reconnaissance de la parole qui a été développée par *Sun Microsystems*. *JSGF* reprend les styles et conventions de la programmation en Java en plus des notations de grammaires traditionnelles. Un exemple est montré dans la figure 2.15.

```
#JSGF V1.0;
/**
 * JSGF Grammar
 */
grammar test;
public <greet> = (sabah | masaa) (alkhayr | alnor | alward);
```

FIGURE *2.15* – Exemple de grammaire de type JSGF

2.4.1.2 La construction d'un modèle statistique de la langue Arabe

Pour avoir la possibilité de générer un modèle de langage, l'outil *CMULMTK* doit être téléchargé et installé. Cet outil représente un package contenant plusieurs librairies

implémentées en C++. Pour pouvoir se servir de ce package, nous avons utilisé comme plateforme d'exploitation le système Linux en sa version *Mandriva*. Le manuel de ce package contient plusieurs étapes à suivre pour une installation correcte.

Une fois l'installation de cet outil est finie avec succès, il faut définir les variables d'environnement comme suit :

```
export SPHINXTRAIN = /root/…/sphinxtrain
export CMUCLMTK = /root/…/cmuclmtk
```

Pour créer le modèle de langage, nous avons besoin des transcriptions des fichiers audio que nous avons pu collecter comme échantillons de la base de données. Quelques lignes de ces transcriptions sont montrées dans la figure 2.16. Ces lignes ont été extraites depuis le fichier de transcription de la base de données audio. <s> et </s> représentent le début et la fin d'une phrase ou d'un mot. Ces derniers sont souvent appelés des nœuds en contexte et il existe un nombre de certain nœuds en contexte, par exemple représentant le début et la fin des paragraphes, le début et la fin d'une phrase ou mot. Plus d'informations peuvent être trouvées dans [23].

```
<s> NAAM </s>
<s> LA </s>
<s> SABAHALKHEIR </s>
<s> ILALIKAA </s>
<s> LILATOUNSAAEDA </s>
<s> MINFADLEK </s>
<s> CHOUKRAN </s>
<s> JAMIL</s>
<s> KABIH </s>
<s> JAYID </s>
<s> RADI </s>
<s> SAGHIR </s>
<s> KABIR </s>
<s> MAFTOUH </s>
<s> MOUGHLAK </s>
<s> KARIB </s>
<s> BAAID </s>
<s> FATAA </s>
<s> MOUSIN </s>
```

FIGURE *2.16* – Le fichier de transcription

Ensuite, il faut générer un fichier vocabulaire. Ce vocabulaire représente tous les mots dont le modèle de langage Arabe à créer va couvrir. A la suite de cela, il faut aussi générer le format *ARPA* du modèle de langage en utilisant les commandes suivantes :

```
% text2idngram -vocab Arab.vocab -idngram Arab.idngram < Arab.txt
% idngram2lm -vocab_type 0 -idngram Arab.idngram -vocab \
Arab.vocab -arpa Arab.arpa
```

Finalement, il faut générer sous forme binaire (*DMP*) le modèle de langage final en utilisant les commandes suivantes :

```
sphinx_lm_convert -i Arab.arpa -o Arab.lm.DMP
```

2.4.2 La construction du dictionnaire

Le dictionnaire peut être conçu par deux manières différentes. La première c'est d'utiliser un outil web qui permet de construire automatiquement un fichier dictionnaire mais aussi un modèle de langage comme précédemment expliqué. L'extension des fichiers générés serait '.dic' pour le fichier dictionnaire et '.lm' pour le fichier modèle de langage. Vu qu'aucune ressource n'est à l'heure actuelle disponible pour la langue Arabe, nous avons opté pour la construction manuelle d'un dictionnaire qui couvre un certain nombre de mots et de petites phrases en Arabe. Par le biais de ce qui est illustré dans la figure 2.17 suivante, nous allons expliquer comment nous avons créé un fichier dictionnaire Arabe.

```
AALA              AH AA L AH
ASFAL             AA S F AH L
ANAJAIAA          AH N AH JH AH AA IH AA
AYNA              AH IY N AH
BAAID             B AH AA IY DH
BARID             B AH R IH D
BATIAA            B AH T IY AH
BIJANIB           B IH JH AH N IH B
BEDAKHEL          B IH D AH G IH L
BAHID             B AH HH IH D
BAKHS             B AH G S
BAKIR             B AH K IH R
BAAD              B AH AA D
BIKAM             B IH K AH M
CHOUKRAN          CH UH K R AH N
DAKHEL            D AH G IH L
FAWKA             F AW K AH
FATAA             F AH T AH
FARIGH            F AH R IH G
HOUNA             HH OW N AH
HOUNAK            HH OW N AH K
ILALIKAA          IY L AH L AY K AH
JAMIL             JH AH M IY L
JAYID             JH AH Y IY D
KABIR             K AH B IY R
KABIH             K AE B IY HH
KARIB             K AH R IY B
KHARIJ            G AH R IH JH
```

FIGURE *2.17* – Le fichier dictionnaire

La partie gauche du fichier 'arabdic.dic' comme montré dans la figure 2.17 représente l'ensemble des mots tandis que la partie droite du fichier représente l'ensemble des phonèmes d'un mot donné. Le délimiteur pour les mots et la séquences des phonèmes est le

caractère blanc ou une tabulation. S'il existe des mots qui sont prononcés différemment par différents locuteurs alors différentes entrées seront stockés dans le dictionnaire.

Un mot peut avoir plus de deux prononciations différentes. La liste de tous les mots dont l'application doit reconnaitre devra être pré-sauvegardée dans un fichier vocabulaire dont l'extension est souvent '.vocab' dans un ordre alphabétique. Cela fournit un bon point de départ pour construire un dictionnaire. Un autre fichier est encore nécessaire à générer qui consiste en la liste des phonèmes prise en charge par l'application. Ce fichier, dont l'extension est '.phone', contient toute la liste des phonèmes qui ont été utilisés pour construire le dictionnaire. D'autres phonèmes spéciaux sont nécessaires pour compléter la liste de phonèmes comme par exemple le silence qui est représenté par 'SIL'. La figure 2.18 montre un extrait du contenu du fichier 'Arab.phone'.

```
AA
AE
AH
AY
AW
B
CH
D
DH
EY
F
G
H
HH
```

FIGURE *2.18* – Fichier dictionnaire de la liste des phonèmes

L'application a besoin aussi de reconnaitre les sons bruits, pour cela, un fichier dont l'extension est '.filler' doit être créé. Il s'agit d'une liste de transcription de sons susceptibles de ne pas être de la parole.

Après avoir construit le modèle de langage, il ne reste à présent que la création du modèle acoustique. Dans ce qui suit, nous donnons les détails pratiques concernant comment débuter la création d'un modèle acoustique de la langue Arabe.

2.4.3 La construction du modèle acoustique

L'outil CMUSphinx [5] fournit différents modèles acoustiques de haute qualité. Il en existe pour l'Anglais des modèles acoustiques de la voix parlé via un microphone, mais aussi de la parole via des conversations téléphoniques. Ces modèles ont été optimisés pour permettre de bonnes performances en termes de résultats de la reconnaissance et pour bien s'adapter avec la plupart des applications.

En ce qui concerne les modèles acoustiques, Sphinx fournie plusieurs manières d'adap-

tation qui sont suffisante pour la plupart des cas quand plus de précision est nécessaire. Une technique d'adaptation est connue pour pouvoir fonctionner parfaitement quant à l'utilisation de différents environnement d'enregistrement (une distance proche ou loin du microphone, ou des canaux téléphoniques) ou même quand il s'agit d'un autre langage différent de celui de l'Anglais. Une adaptation, par exemple, fonctionne bien si on a rapidement besoin de rajouter un support pour quelques nouvelles langues seulement en faisant un mapping de l'ensemble des phonèmes d'un modèle acoustique fournit à un dictionnaire de phonèmes de la langue cible.

Il y a, malheureusement, des applications dont les modèles acoustiques existants ne fonctionnent pas. Cela nécessite un processus d'apprentissage de modèles.

2.4.3.1 La préparation de données

Le classifieur choisit pour l'application, représenté par les HMMs dans notre cas, va essayer d'apprendre les paramètres des modèles des unités de sons en utilisant un ensemble d'échantillons de signal de la parole ou proprement dit ; une base de données d'apprentissage. Cette base de données contient des informations nécessaires dont le but est d'extraire des statistiques à partir du signal de la parole sous forme d'un modèle acoustique.

Le classifieur à besoin dans ce cas d'être informé quelles sont les unités de sons dont les paramètres devront être appris, ou la séquence au niveau de laquelle ces unités appartiennent pour chaque échantillon de parole dans la base de données. Cette information est fournie au classifieur à travers un fichier de configuration nommé fichier de transcription, dans lequel la séquence de mots et les unités de sons qui ne représentent pas de la parole sont mentionnés exactement comme apparus dans le signal de la parole, suivi par un *tag* qui peut être utilisé pour associer cette séquence à l'échantillon de la parole correspondant.

Ensuite, le classifieur vérifie dans le dictionnaire qui va effectuer un *mapping* de chaque mot à une séquence d'unité de sons pour générer la séquence des unités de sons associée à chaque échantillon dans la base de données.

Le classifieur a besoin aussi de deux types de dictionnaire, l'un concerne le mapping des mots qui appartiennent au langage aux séquences des unités de sons, contrairement à l'autre dictionnaire qui concerne le mapping des unités de sons qui ne représentent pas de la parole à leurs unités de sons correspondantes.

2.4.3.2 La compilation des packages nécessaires

Les packages suivants nous ont été très utiles pour la création d'un modèle acoustique Arabe : *sphinxbase-0.8*, *sphinx-0.8* et *sphinxtrain-0.8*. La création et le test du modèle acoustique Arabe a été effectué sous une station Linux afin d'utiliser toutes les caracté-

ristique de l'outil Sphinx qui ne sont pas tous compatibles sous une station exécutant un système d'exploitation Windows. Pour les scripts interagissants avec Sphinx pour tester le modèle acoustique Arabe créé nous avons utilisé le langage script *ActivPerl*.

2.4.3.3 Le configuration des fichiers scripts pour la phase d'apprentissage

Pour commencer la phase d'apprentissage, tout devra s'exécuter sous le répertoire là où la base de données *wav* a été sauvegardée. Pour simplifier les choses, la commande suivante permet d'initialiser le processus d'apprentissage pour la création du modèle acoustique :

```
sphinxtrain -t arab setup
```

Cela va copier tous les fichiers de configuration nécessaires dans un sous répertoire '*etc*' et préparer la base de données pour l'apprentissage, la structure des sous répertoires créée est illustrée comme suit :

```
etc
feat
logdir
model_parameters
model_architecture
wav
```

Après l'initialisation du processus de l'apprentissage, seuls les deux répertoires '*etc*' et '*wav*' seront présents ; les autres seront créés lors de l'exécution du processus d'apprentissage.

Ensuite, le dictionnaire, la séquence de tous les phonèmes ainsi que le fichier de transcription précédemment créés doivent être placés dans le répertoire '*etc*'. Quelques fichiers de configuration présents sous le répertoire '*etc*' devront être modifié, il existe plusieurs variables mais seules quelques-unes parmi elles seront modifiées. Ces variables se trouvent dans le fichier configuration '*sphinx_train.cfg*' sous le répertoire '*etc/sphinx_train.cfg*'.

2.4.3.4 La configuration du format de la base de données audio

Par défaut, le fichier de configuration '*sphix_train.cfg*' vient avec les informations illustrées dans ce qui suit :

```
$CFG_WAVFILES_DIR = "$CFG_BASE_DIR/wav";

$CFG_WAVFILE_EXTENSION = 'sph';

$CFG_WAVFILE_TYPE = 'nist';
```

Vu que la base de données audio utilisée dans ce projet de thèse était enregistrée sous un format audio '*wav*', l'extension '*sph*' devrait être changée à '*mswav*' comme illustré ci-dessous :

```
$CFG_WAVFILES_DIR = "$CFG_BASE_DIR/wav";

$CFG_WAVFILE_EXTENSION = 'wav';

$CFG_WAVFILE_TYPE = 'mswav';
```

2.4.3.5 Configuration de la variable d'environnement

Certaines valeurs sont déjà initialisées dans le fichier de configuration '*sphinx_train.cfg*'. Les autres devront être définit selon le nom attribué au début, comme par exemple le nom du fichier dictionnaire, transcription...etc. La figure 2.19 illustre les modifications des variables d'environnement qui doivent être effectuées pour la prise en charge des fichiers dictionnaire et transcription précédemment créés.

```
# Variables used in main training of models

$CFG_DICTIONARY     = "$CFG_LIST_DIR/$CFG_DB_NAME.dic";

$CFG_RAWPHONEFILE   = "$CFG_LIST_DIR/$CFG_DB_NAME.phone";

$CFG_FILLERDICT     = "$CFG_LIST_DIR/$CFG_DB_NAME.filler";

$CFG_LISTOFFILES    = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.fileids";

$CFG_TRANSCRIPTFILE = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.transcription"
```

FIGURE *2.19* – Les variables d'environnement pour l'apprentissage du modèle acoustique

Le nombre de densités gaussiennes dépend étroitement de la taille du vocabulaire utilisé. Si l'apprentissage effectué est sur un ensemble de modèle continu pour une grande taille de vocabulaire et que les données audio sont plus de 100 heures d'enregistrement, alors le mieux c'est de choisir la valeur 32. Pour le modèle acoustique Arabe, et vu que la taille de la base de données audio utilisée n'est pas aussi importante, une valeur initialisée à 8 densités gaussiennes était suffisante, comme illustré ci-dessous :

```
$CFG_FINAL_NUM_DENSITIES = 8;
```

Cette variable pourrait prendre n'importe quelle valeur d'ordre 2 ; 4, 8, 16, 32, 64. Une autre variable est aussi trop importante dans le fichier de configuration '*sphinx_train.sfg*', il s'agit du nombre des états qui vont former le treillis de recherche pour l'apprentissage du modèle. Plus le modèle a plusieurs états, plus le signal de la parole est précisément discriminé. Mais il faut noter que si plusieurs états sont envisagés, le modèle ne va pas être assez générique pour reconnaitre avec précision le signal de la parole. Cela veut dire que le taux d'erreur généré sera trop élevé. C'est pour cela qu'il est très important d'éviter des situations qui mènent vers un sur-apprentissage des modèles. Le nombre approximatif,

TABLE *2.3* – Le nombre approximatif des états et des densités gaussiennes

Vocabulaire	Heures d'enregistrement	États	Densités	Exemple
20	5	200	8	Tidigits Digits Recognition
100	20	2000	8	RM1 Command and Control
5000	30	4000	16	WSJ1 5k Small Dictation
20000	80	4000	32	WSJ1 20k Big Dictation
60000	200	6000	16	HUB4 Broadcast News
60000	2000	12000	64	Fisher Rich Telephone Transcription

comme mentionné dans le tutoriel sur la dernière version à l'heure actuelle de l'outil Sphinx [5], des états (*Senones*) et le nombre de densité sont mentionnés dans le tableau 2.3.

Il faut noter aussi que seuls les états représentant les unités de sons dans le fichier de transcription pourront être entrainés. Cela veut dire que si la transcription effectuée n'est pas assez générique, par exemple le même mot prononcé par 1000 locuteurs 1000 fois c'est l'équivalant de dire « seulement quelques *Senones* pourront représentés ces données », peu importe sur combien d'heures les données ont été enregistrées.

Pour avoir un modèle acoustique adéquat, il faut effectuer des expérimentations avec différents paramètres, et essayer ensuite de ne garder que ceux qui donnent de meilleures précisions. Souvent, les variables qui peuvent influencer les performances du modèle acoustique sont : le nombre de *Senones* et le nombre de *mixtures gaussiennes*.

2.4.3.6 La configuration des paramètres audio de la base de données

Par défaut, les fichiers audio pris en charge par l'outil Sphinx sont de ratio de 16000 échantillons par seconde ($16Khz$). Si cela est le cas, le fichier 'feat.params' ou fichier paramètres de caractéristiques sera automatiquement généré avec les valeurs recommandées.

Dans le cas des fichiers audio avec un ratio d'échantillonnage de $8Khz$ (comme dans le cas des échantillons extraits depuis une base de données de conversation téléphonique), certaines valeurs dans le fichier de configuration 'sphinx_train.cfg' doivent être sélectionnées. Le ratio d'échantillonnage le plus bas signifie la présence d'un changement dans la fréquence des sons utilisée et le nombre des filtres utilisés pour reconnaitre la parole. Les valeurs recommandées sont illustrées dans la figure 2.20.

```
$CFG_WAVFILE_SRATE = 8000.0;

$CFG_NUM_FILT = 31; # For wideband speech it's 40, for telephone 8khz
reasonable value is 31

$CFG_LO_FILT = 200; # For telephone 8kHz speech value is 200

$CFG_HI_FILT = 3500; # For telephone 8kHz speech value is 3500
```

FIGURE *2.20* – Les valeurs recommandées pour la fréquence des sons et le nombre des filtres utilisés

2.4.3.7 La configuration des paramètres de décodage

Certaines variables dans le fichier de configuration 'sphinx_train.cfg', utilisées par le décodeur, doivent être définies par l'utilisateur et peuvent affecter les résultats de sortie du décodeur. Ces variables concernent le nom des deux types de ficher dictionnaire précédemment définit, le fichier de transcription, et un fichier qui doit contenir le nom de tous les échantillons de la base de données audio mentionné en enlevant leur extension. La figure 2.21 présente des lignes dans le fichier de configuration là où ces variables devront être définis.

```
$DEC_CFG_DICTIONARY     = "$DEC_CFG_BASE_DIR/etc/$DEC_CFG_DB_NAME.dic";

$DEC_CFG_FILLERDICT     = "$DEC_CFG_BASE_DIR/etc/$DEC_CFG_DB_NAME.filler";

$DEC_CFG_LISTOFFILES   =
      "$DEC_CFG_BASE_DIR/etc/${DEC_CFG_DB_NAME}_test.fileids";

$DEC_CFG_TRANSCRIPTFILE="$DEC_CFG_BASE_DIR/etc/${DEC_CFG_DB_NAME}_test.transcr
iption";

$DEC_CFG_RESULT_DIR     = "$DEC_CFG_BASE_DIR/result";

$DEC_CFG_LANGUAGEMODEL_DIR = "$DEC_CFG_BASE_DIR/etc";

$DEC_CFG_LANGUAGEMODEL  = "$DEC_CFG_LANGUAGEMODEL_DIR/Arab.lm.DMP";
```

FIGURE *2.21* – L'initialisation de la variable dictionnaire et transcription

Il faut définir le nom des fichiers nécessaires pour la phase de l'apprentissage d'une sorte à ce qu'ils commencent tous par le même préfixe.

2.4.4 L'apprentissage

Tout d'abord, avant de commencer la phase de l'apprentissage, le décodeur doit travailler sur des fichiers de caractéristiques et non pas des fichiers sons *wav*. Pour cela, une étape d'extraction de caractéristiques est nécessaire, ensuite le décodage aura lieu sur ces fichiers de caractéristique dont l'extension est '*mfc*'. L'outil *sphinx_base* permet l'extraction des caractéristiques des fichiers *wav* de la base de données en utilisant la commande

suivante via l'exécutable '*sphinx_fe*' :

```
$SPHINX/bin/sphinxfe -c arab train.fileids -di wav/ -do feats/ -ei mswav -mswav
yes -eo mfc
```

Une fois les fichiers de caractéristiques sont extraits, un autre fichier doit être créé ; '*feat.ctl*' qui consiste en les noms des fichiers de caractéristique mentionnés sans leurs extensions. Une fois le fichier '*feat.ctl*' est créé, le décodage pourra être lancé pour la génération du modèle acoustique en utilisant la commande suivante via l'exécutable '*sphinx3_decode*' :

```
$SPHINX/bin/sphinx3_decode -hmm models/hmm/ -lm models/lm/arab_train.lm.DMP -
dict models/lm/arab.dic -fdict models/lm/arab.filler -ctl arab_train.fileids-
logfn log.txt -hyp out.txt -cepdir feats/
```

Après cela, le décodeur exécute les différents stages nécessaires pour la création du modèle acoustique. Cela va prendre quelques minutes pour la phase de l'apprentissage. Dans le cas d'une base de données audio de taille importante, l'apprentissage pourra prendre un temps considérable.

Durant l'exécution des différentes étapes, le premier stage représente le stage le plus important qui va vérifier si la configuration des variables initiales a été effectuée correctement ou non, ainsi que la consistance des données. La figure 2.22 illustre l'état d'exécution durant le processus du décodage.

```
Baum welch starting for 2 Gaussian(s), iteration: 3 (1 of 1)

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Normalization for iteration: 3

Current Overall Likelihood Per Frame = 16.7360332231956

Convergence Ratio = 0.167477903890013

Baum welch starting for 2 Gaussian(s), iteration: 4 (1 of 1)

0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%

Normalization for iteration: 4
```

FIGURE *2.22* – L'exécution pendant l'apprentissage du modèle acoustique

La figure 2.23 montre une liste de scripts représentant toutes les étapes nécessaires pour la création du modèle acoustique.

```
perl scripts_pl/000.comp_feat/slave_feat.pl

perl scripts_pl/00.verify/verify_all.pl

perl scripts_pl/10.vector_quantize/slave.VQ.pl

perl scripts_pl/20.ci_hmm/slave_convg.pl

perl scripts_pl/30.cd_hmm_untied/slave_convg.pl

perl scripts_pl/40.buildtrees/slave.treebuilder.pl

perl scripts_pl/45.prunetree/slave-state-tying.pl

perl scripts_pl/50.cd_hmm_tied/slave_convg.pl

perl scripts_pl/90.deleted_interpolation/deleted_interpolation.pl
```

FIGURE *2.23* – Les scripts nécessaires pour l'apprentissage du modèle acoustique

Pendant la création du modèle acoustique, plusieurs scripts Perl sous le répertoire '*.../scripts_pl*' seront exécutés l'un après l'autre. Ces scripts sont numérotés de 00∗ à 99∗. Ils seront exécutés dans un ordre chronologique.

Après l'exécution de chaque script, plusieurs répertoires vont apparaitre, qui contiennent des fichiers générés au cours du processus de l'apprentissage. Un de ces fichiers important généré est celui qui a l'extension HTML, nommé '*arab.html*'. Ce dernier contient, après la fin du décodage, un rapport des résultats de l'exécution de tous les scripts illustrés dans la figure 2.24.

```
MODULE: 000 Computing feature from audio files  (2014-01-21 21:55)
Extracting features from segments starting at (part 1 of 1)
sphinx fe Log File
completed
Extracting features from segments starting at (part 1 of 1)
sphinx fe Log File
completed
Feature extraction is done

MODULE: 00 verify training files        (2014-01-21 21:55)
Phase 1: Checking to see if the dict and filler dict agrees with the phonelist
file.
Found 103 words using 35 phones
passed
Phase 2: Checking to make sure there are not duplicate entries in the
dictionary
passed
Phase 3: Check general format for the fileids file; utterance length (must be
positive); files exist
passed
Phase 4: Checking number of lines in the transcript file should match lines in
fileids file
passed
Phase 5: Determine amount of training data, see if n_tied_states seems
reasonable.
Estimated Total Hours Training: 1.28290555555556
This is a small amount of data, no comment at this time
WARNING
Phase 6: Checking that all the words in the transcript are in the dictionary
Words in dictionary: 100
Words in filler dictionary: 3
passed
Phase 7: Checking that all the phones in the transcript are in the phonelist,
and all phones in the phonelist appear at least once
Passed
…
```

FIGURE *2.24* – Rapport de résultats d'exécution des scripts

Il faut noter aussi que lors de l'exécution des scripts 00∗ à 99∗, plusieurs ensembles de modèles acoustiques seront générés et chacun d'entre eux pourra être utilisé lors de la reconnaissance de la parole. Il faut dire aussi que seules quelques étapes sont nécessaires pour la création du modèle acoustique semi-continue.

Pendant l'exécution du script '*slave_feat.pl*', des fichiers de caractéristiques seront extraits. Le système ne travaille pas directement sur des signaux acoustiques. Les signaux de la parole seront transformés d'abord en une séquence de vecteurs de caractéristiques, qui sont utilisé à la place des échantillons *wav* de la base de donnée audio. Le script '*make_feats.pl*' va calculer, pour chaque prononciation, les coefficients *MFCC*.

Une fois le script '*20.ci_hmm/slave_convg.pl*' est exécuté, l'apprentissage des modèles indépendant du contexte (*Context-Independent*) pour les unités phonétique dans le dictionnaire précédemment crée sera effectué.

Pour le script '*30.cd_hmm_untied/slave_conveg.pl*', l'apprentissage des modèles dépendants du contexte (*triphone Context-Dependent*) pour les unités phonétique, avec des états qui ne sont pas directement liés, sera effectué. Ces modèles sont nécessaires pour la

construction des arbres de décision pour lier les états.

Le script '*40.buildtrees/slave_convg.pl*' va créer des arbres de décision pour chaque état de chaque unité phonétique. Le script '*45.prunetree/slave-state-tying.pl*' consiste en l'élagage des arbres de décision pour lier les états.

Le script '*50.cd_hmm_tied/slave_convg.pl*' consiste en l'apprentissage des modèles acoustiques finaux dépendants du contexte (*triphone*). L'apprentissage effectué sur les modèles dépendants du contexte est réalisé en plusieurs stages. Cela débute par une seule gaussienne par état *hmm*, suivi par 2 gaussiennes par état *hmm* et ainsi de suite jusqu'à ce que le nombre de gaussienne choisi par état sera exécuté.
Le script '50.cd_hmm_tied/slave_convg.pl' va automatiquement effectuer l'apprentissage de tous les modèles dépendent du contexte intermédiaire.

À la fin de chaque stage, les modèles sont prêts à être utilisés. Il est possible aussi d'effectuer une reconnaissance pendant que l'apprentissage des modèles est en cours d'exécution ; à condition que le stage générant ces modèles soit terminé avec succès.

2.4.5 Le test du modèle acoustique conçu

Il est primordial de tester la qualité de la base de données d'apprentissage pour sélectionner les meilleurs paramètres, comprendre comment l'application fonctionne et optimiser ainsi les performances. Afin de rendre cela faisable, une étape de décodage ou de test est nécessaire. Cela représente le dernier stage du processus de création d'un modèle acoustique en utilisant l'outil Sphinx. Toutefois, il est toujours faisable de recommencer le test via la commande suivante :

```
sphinxtrain -s decode run
```

Cette commande débutera le processus de décodage en utilisant le modèle acoustique conçu ainsi que le modèle de langage configuré dans le fichier de configuration '*sphinx_train.cfg*'. La figure 2.25 montre le début de l'exécution de la dernière étape test du modèle acoustique conçu.

```
MODULE: DECODE Decoding using models previously trained

        Decoding 130 segments starting at 0 (part 1 of 1)

0%
```

FIGURE *2.25* – Début de l'exécution de la dernière étape test du modèle acoustique conçu

Quand le processus de la reconnaissance est terminé, le script calcule le taux d'erreur au niveau mot (WER ; word error rate) et le taux d'erreur au niveau phrase (SER ; sentence

error rate). Plus petit le taux obtenu serait, plus pertinent le modèle acoustique serait.

```
MODULE: DECODE Decoding using models previously trained  (2014-01-21 21:58)


Decoding 116 segments starting at 0 (part 1 of 1)

pocketsphinx_batch Log File

completed

Aligning results to find error rate


SENTENCE ERROR: 2.6% (3/116) WORD ERROR RATE: 3.5% (4/116)
```

```
…

bidaya  (03-REC0119-041219)

bidaya  (03-REC0119-041219)

Words: 1 Correct: 1 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy
= 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

nihaya  (03-REC0119-041222)

nihaya  (03-REC0119-041222)

Words: 1 Correct: 1 Errors: 0 Percent correct = 100.00% Error = 0.00% Accuracy
= 100.00%

Insertions: 0 Deletions: 0 Substitutions: 0

TOTAL Words: 116 Correct: 113 Errors: 4

TOTAL Percent correct = 97.41% Error = 3.45% Accuracy = 96.55%

TOTAL Insertions: 1 Deletions: 0 Substitutions: 3
```

FIGURE *2.26* – Journal de résultats de décodage

Plus de détails concernant le décodage, comme par exemple l'alignement avec la transcription de référence, la vitesse et le résultat de décodage de chaque fichier *wav* de la base de données, sont dans un fichier journal récupéré après l'exécution : '*arab.align*'. La figure 2.26 montre un extrait du contenu de ce fichier journal.

2.5 Conclusion

Ce chapitre a été consacré à l'introduction de la reconnaissance automatique de la parole ainsi que l'outil de modélisation le plus utilisé dans ses applications qui est : Les modèles de Markov cachés. Nous avons montré, à travers ce chapitre, la puissance de

l'outil Sphinx à rendre possible l'implémentation d'un outil qui permet une évaluation de la prononciation. Dans ce qui suit, nous allons introduire les systèmes d'évaluation automatique de la prononciation.

Les systèmes d'évaluation de la prononciation

3.1 Introduction

L'apprentissage de la langue assisté par ordinateur (*CALL* pour Computer Assisted Language Learning) est une forme d'apprentissage de langues basé sur l'ordinateur qui porte deux caractéristiques importantes : apprentissage bidirectionnel et apprentissage individualisé.

La philosophie du CALL permet aux étudiants d'apprendre les langues tous seuls en utilisant des leçons interactifs (bidirectionnels) et individuels comme elle peut renforcer ce qui a été apprit dans les classes.

Dans les années précédentes, les systèmes d'enseignement de langue assisté par ordinateur CALL ont été principalement basés sur le traitement naturel des langues (NLP : Natural Language Processing) comme par exemple en incluant des composantes basées sur la grammaire et le vocabulaire. Heureusement, les avancés dans le domaine de la reconnaissance automatique de la parole ont contribué au développement des systèmes d'enseignement de prononciation assisté par ordinateur CAPT (Computer Assisted Pronunciation Teaching) et permettent l'évaluation automatique de la prononciation. Cette évaluation est souvent fournie par un *feedback* sous forme d'une mesure ou un score. Différentes mesures ont été proposées pour évaluer, quantitativement, la qualité de la prononciation de l'apprenant où de mesurer la maitrise de la parole. Le présent chapitre est dédié à la présentation des systèmes *CAPT* et *CALL* ainsi que les mesures les plus utilisées dans le domaine de l'évaluation de la prononciation. Nous y soulevant aussi, un des problèmes inhérents à cette discipline qui est la variabilité des évaluations des experts qui constitue un frein à une évaluation automatique consensuelle.

3.2 L'apprentissage des langues assisté par ordinateur

les chercheurs ont investi l'utilisation des ordinateurs pour l'apprentissage des langues depuis 1960 [24]. La recherche dans les domaines impliquant les systèmes *CALL* a explosé durant cette dernière décennie. La recherche dans le domaine *CALL* peut être regroupée dans deux classes : une recherche focalisée sur les systèmes et une recherche focalisée sur les techniques implémentées au niveau de ces systèmes.

Les systèmes CALL sont sous plusieurs formes et avec de différentes configurations. D'une manière simple, ces systèmes peuvent avoir une architecture sous forme de pages web, des fichiers audio dont le but est d'entendre/imiter la voix, des programmes multimédia statistiques. D'une manière un peu plus compliquée, ces systèmes ont un moteur de reconnaissance automatique de la parole, synthèse de la parole, et des environnements

TABLE *3.1* – Comparaison entre un tuteur humain et un système informatique d'après [25]

	Ordinateur	Tuteur humain
Disponibilité	24/24, 7/7	Limité
Attention à l'apprenant	100%	Faible sauf dans le cas d'un seul étudiant
Coût	Ordinateur avec une connexion Internet	Chère dans le cas d'un seul étudiant
Niveau de stress	Difficulté pour ajuster le microphone, erreurs de reconnaissance	Embarras pour s'exprimer devant une classe
Prononciation/ Intonation	Parole de synthèse, peut paraitre naturelle	naturelle

interactives en *3D* dont le but est d'enseigner des normes culturelles et linguistiques.

Les systèmes CALL modernes visent à être des environnements d'apprentissage de langue plus riches qui incluent une meilleure qualité audio et graphique et avec des *feedbacks* informatifs. Le contenu des leçons n'est pas forcement statique, et est généré aléatoirement ou d'une manière à ce qu'il soit adapté, en réponse aux actions de l'apprenant. Plusieurs systèmes utilisent quelques formes de reconnaissance automatique de la parole, la synthèse de la parole, la compréhension du langage naturel ou la génération du langage naturel. Comparé à un tuteur humain, un système informatique offre de nombreux avantages (voir tableau 3.1).

Les systèmes de dialogue peuvent être utilisés pour créer des environnements dynamiques dans lesquels les apprenants peuvent avoir l'accès à des conversations dynamiques et naturelles d'une manière objective. Au lieu d'être limités à des phrases ou des scripts spécifiques à suivre pendant l'apprentissage d'une langue quelconque, les apprenants peuvent garder les conversations qui sont mises à leur disposition dans des sessions pratiques. Comme la reconnaissance automatique de la parole est loin d'être parfaite, il y a une tension constante dans les systèmes de dialogues entre permettre le choix libre des conversations et limiter suffisamment le domaine pour maintenir une performance acceptable. Les systèmes de dialogue adoptent des stratégies différentes pour une balance appropriée.

On retiendra que depuis les premiers jours, le *CALL* a été développé dans une relation symbiotique entre le développement de la technologie et de la pédagogie.

3.3 L'apprentissage de la prononciation assisté par ordinateur *CAPT*

Les systèmes CAPTs ont été spécifiquement conçus pour évaluer et améliorer la prononciation. Un système *CAPT* comprend deux composantes principales : un module de la reconnaissance automatique de la parole et un module d'évaluation. En effet, pour attribuer un score de prononciation à l'apprenant ; une étape de reconnaissance est requise.

La première étape dans un système de reconnaissance automatique de la parole consiste en l'extraction des caractéristiques du signal acoustique de la parole. Pour des considérations pratiques, le flux de la parole est segmenté en trames temporelles. Ensuite, chaque trame est traitée indépendamment comme étant un fichier audio. À partir de chaque trame, un ensemble de coefficients acoustiques seront extraits qui sont souvent les *MFCC* (Mel Frequency Cepstral Coefficients). Finalement, et pour chaque fichier audio, une collection de vecteurs acoustiques seront associés.

Dans un système CAPT, la représentation acoustique du signal de la parole en entrée est alignée aux *HMMs* (un ou plusieurs modèle HMMs) qui représentent le texte affiché (voir la figure 3.1) en utilisant le mode alignement forcé (*force-alignment*). Le système produit, à la fin du processus, un score dont le but est de vérifier le degré de la déviation de la prononciation effectuée par rapport à la prononciation correcte.

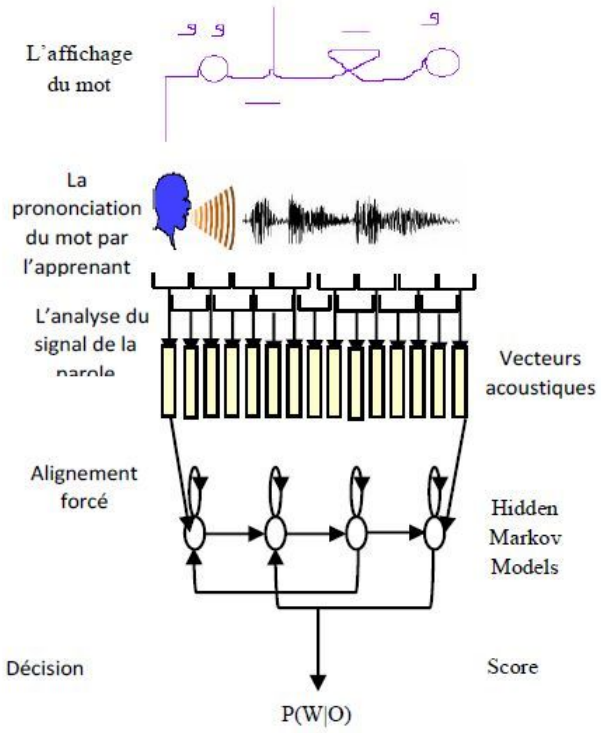

FIGURE *3.1* – Étapes de l'évaluation de la prononciation

Sur la base des sorties du système de reconnaissance, la phase de l'évaluation de la prononciation peut débuter.

Le processus d'évaluation de la prononciation peut être résumé comme l'a suggéré *Neumeyer* dans [26] en trois étapes principales :

1. Génération des segments phonétiques en utilisant un système de reconnaissance de la parole basé sur les HMM,
2. Calcul de scores automatiques pour les différents segments phonétiques en comparant la parole de l'apprenant à celle de locuteurs qui sont natifs de la langue,
3. L'étalonnage de ces scores qui inclut leur réglage et éventuellement la combinaison de certains d'entre-eux.

Le but étant que l'évaluation résultante soit aussi proche que possible de celle d'un expert humain. Pour réaliser cela, il est impératif de récolter des données d'évaluation qui comprennent l'appréciation des experts. Nous allons dans ce qui suit présenter les scores les plus utilisées dans le contexte de l'évaluation de la prononciation et nous aborderons un peu plus loin le problème de corrélation entre les différentes évaluations des experts humains.

3.4 Les différents scores automatique utilisés en l'évaluation de la prononciation

Dans un système CAPT, le texte à prononcer est affiché à l'apprenant (son, mot, une phrase, etc.) ; ainsi, le système « sait » ce que l'apprenant doit prononcer. Ainsi, la tâche de reconnaissance est simplifié puisque le signal entrant est analysé et la représentation acoustique active obtenue est alignée avec les modèles de référence du mot affiché (ou groupe de mots). L'étape de l'alignement forcé du signal en entrée avec le HMM fournit un score qui montre à quel point le discours entrant est ressemblant aux modèles de référence.

Les premières tentatives qui utilisèrent la technologie *RAP* pour évaluer la prononciation ont été réalisées par *Bernstein* et al. dans [27] au cours du développement du système *SRI*, un outil qui vise à enseigner l'anglais à des étudiants japonais. Dans cette première tentative, le score utilisé est une distance entre les caractéristiques séparatrices d'états (SFS : State Feature Separator) de modèles de références HMM et la prononciation à évaluer. Ces états ont été soigneusement identifiés pour des populations distinctes de locuteurs Japonais et Américains.

Dans ce premier travail, les scores automatiques étaient étroitement liés au texte à prononcer, des études ultérieures ont tenté de proposer des mesures qui sont indépendantes

du texte. Les premiers résultats ont utilisé les sorties des *HMM* dans le processus de reconnaissance. Ces sorties étaient : le score de la durée du segment et sa probabilité par rapport à ceux de référence. De plus, la plupart des études considèrent le phone comme ce segment.

Plus tard, d'autres scores dérivés ont été utilisés [28]. La plupart de ces scores sont une généralisation ou une normalisation des deux premiers scores tels que le taux de la parole (Speech Rate) ou de la log-vraisemblance moyenne (Average Log-Likelihood).

Dans ce qui suit nous allons présenter la plupart de ces scores en les regroupant en fonction de leur origine.

3.4.1 Les scores basés temps

Des locuteurs avec une prononciation correcte parlent souvent plus rapidement que les débutants [29]. Ainsi, un ensemble de scores correspondants à la durée de la parole ont été examinés. Tout d'abord, la durée du discours (*TDS* : Time Duration of Speech) est la durée totale du temps nécessaire pour produire le discours ; ce score est considéré par *Cucchiarini et al.* [30] comme étant le meilleur pour évaluer la prononciation. Tandis que la durée des pauses (TDP : Time Duration of Pauses) représente la durée totale des pauses internes produites dans le discours. La durée de discours sans pauses (TDS-wp) est la durée totale du temps nécessaire pour produire le discours sans les pauses internes.

3.4.1.1 Le ratio de la parole (*ROS* : Rate of Speech)

Les locuteurs qui ont une prononciation correcte parlent souvent plus rapidement que les apprenants débutants. Donc, le ratio de la parole peut être utilisé comme un score prédictif pour mesurer le degré de la validité de la prononciation (*pronunciation correctness*). Ce score peut être calculé en divisant le nombre total des phonèmes par la durée prise pour produire ces phonèmes [26]. Les phonèmes silencieux ne sont pas négligés durant le calcul du score ROS.

3.4.1.2 Le ratio de l'articulation (*ROA* : Rate of Articulation)

Durant le calcul du score *ROA* [26], le nombre total de phonèmes (ou unités de sons) produits dans un échantillon de parole est divisé par la durée nécessaire prise pour produire ces phonèmes. Contrairement au calcul du ratio de la parole, les pauses ou les phonèmes silencieux sont négligés durant le calcul de ce score. Comme les apprenants débutants (ou les locuteurs non natifs) ont tendance d'avoir un ratio d'articulation plus bas par rapport aux locuteurs natifs, le ratio de l'articulation peut être aussi considéré comme une bonne mesure ou score pour déterminer le degré de la validité de la prononciation.

3.4.1.3 La durée d'un phonème

Comme précédemment mentionné, les locuteurs experts (ou natifs) prononcent les mots plus rapidement et d'une façon meilleure que les apprenants débutants. On peut avancer que les mesures qui sont basées sur la durée de chaque phonème vont permettre une comparaison dans le but d'estimer quels sont les phonèmes qui ont été prolongés pendant la prononciation. Ce score, pour pouvoir être calculé, requiert la durée en trame du $i^{ème}$ phonème à partir de l'alignement de *Viterbi* [29]. Les phonèmes silencieux (qui ne contiennent aucune donnée) sont négligés pendant le calcul de ce score. Pour obtenir le score durée d'un phonème correspondant, la probabilité logarithmique de la durée du phonème est calculée en utilisant une distribution de durée de ce phonème.

3.4.2 Les scores basée vraisemblance

Les scores basés sur la vraisemblance sont basés sur le résultat de l'alignement forcé du signal en entrée avec le modèle du mot, qui se traduit par une probabilité à postériori.

3.4.2.1 Le logarithme de vraisemblance

Il est supposé qu'avec des modèles *HMM* entrainés sur une base de donnée audio native, le logarithme de vraisemblance de données audio, déterminé par le biais de l'algorithme de Viterbi HMM, est une bonne mesure pour déterminer le degré de similarité entre la parole native et la parole non native quand les modèles HMMs sont entrainés sur un corpus de données natives [26]. Pour chaque phrase, une segmentation phonétique est obtenue avec le logarithme de vraisemblance de chaque segment de phonème. Si t_i est le temps initial du i^{th} segment phonétique, le logarithme de vraisemblance total de ce segment peut être calculé, en utilisant les HMMs, par l'équation (3.1) :

$$LL_i = \sum_{t=t_i}^{t_{i+1}-1} log(p(S_t|S_{t-1})p(X_t|S_t)) \tag{3.1}$$

où X_t est le vecteur spectral observé, et S_t est l'état HMM qui correspond au temps t, $p(S_t|S_{t-1})$ est la probabilité de transition HMM et $p(X_t|S_t)$ représente la distribution de sortie d'un état HMM S_t [26].

Pour un certain niveau de divergence entre le signal de la parole en entrée et les modèles de référence, le logarithme de vraisemblance dépend de la longueur du mot (si on considère le logarithme de vraisemblance de chaque phonème du mot), la moyenne globale du logarithme de vraisemblance (*GLL* : Global average Log Likelihood) a été introduite dans [31]. Ce dernier peut être calculé par l'application de l'équation (3.2) suivante :

$$GLL = \frac{\sum_{i=1}^{N} LL_i}{\sum_{i=1}^{N} d_i} \tag{3.2}$$

Dans ce cas, le degré de vraisemblance des phonèmes longs a tendance de dominer le score GLL. Même si les phonèmes courts peuvent avoir un effet perceptuel important, avec une petite durée, le degré de divergence entre eux peut être caché par celui des phonèmes longs. Pour pallier à ce problème, la moyenne locale du logarithme de vraisemblance (LLL : Local average Log Likelihood) est utilisée. Ce score peut être calculé par l'équation (3.3) suivante :

$$LLL = \frac{1}{N} \sum_{i=1}^{N} \frac{LL_i}{d_i} \tag{3.3}$$

Dans [26], les auteurs ont montré à travers leur étude comparative que la corrélation entre les scores acoustiques basés sur le logarithme de vraisemblance et les scores attribués par les experts humains peut être améliorée si ces deux derniers scores sont normalisés en se basant sur une estimation du degré de ressemblance entre les caractéristiques spectrales du signal de la parole et l'apprentissage des données venant des locuteurs natifs.

3.4.2.2 Le GOP (Goodness of Pronunciation)

Le score GOP [32] [33] est un score dérivé à partir du logarithme de vraisemblance obtenu de la phase de décodage HMM. L'algorithme proposé par [32] [33] calcule le ratio du logarithme de vraisemblance qui, une fois le phonème est prononcé, correspond au phonème qui doit être prononcé. Au niveau du calcul de ce score, deux modes de reconnaissance sont appliqués lors de la réception de la parole de l'apprenant : l'une est une reconnaissance en mode alignement forcé (reconnaissance de mot exacte ou phrase plus une segmentation), l'autre est en mode libre (reconnaissance de phonèmes ; pas besoin d'une segmentation). Un score GOP pour une réalisation d'un phonème spécifique est ensuite déterminé (voir figure 3.2) en prenant la différence entre la probabilité logarithmique obtenue durant la phase de reconnaissance en mode alignement forcé et celle obtenue durant la phase de reconnaissance en mode libre. Quand un score GOP est calculé, une valeur seuil doit être appliquée afin de rejeter les phonèmes qui ont été mal prononcés. Le choix de cette valeur dépend du niveau de la précision requis [33].

3.4.3 Autres scores

D'autres scores dérivés peuvent être pris en considération, nous citons entre autres :

Les pauses : ce score représente le nombre de pauses qui ont été effectuées avant la

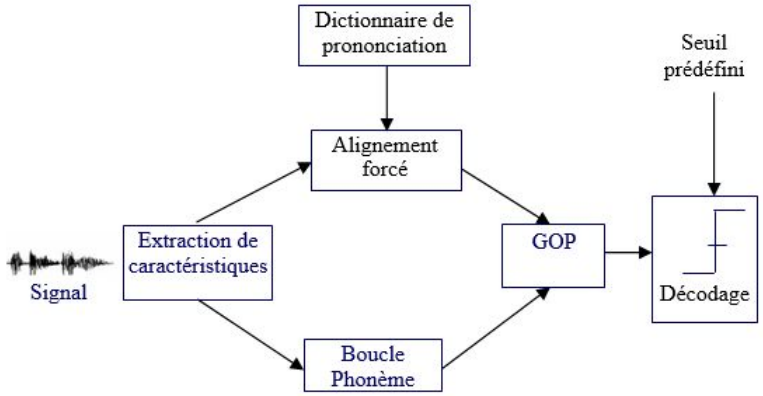

FIGURE *3.2* – Calcul de la mesure GOP

fin de la prononciation.

GLL et LLL sans pauses (GLL-wp et LLL-wp) : ces deux scores sont calculés de la même manière que les scores GLL et LLL précédemment expliqués sauf que cette fois-ci les pauses ou les phonèmes silencieux sont négligés.

Le ratio de la non parole (RONS : rate of non speech) : ce score peut être calculé en divisant le nombre de pauses détectées par la durée totale requise pour produire ces pauses. Plus cette valeur sera basse, moins la chance que la parole est effectuée avec aisance serait.

Le ratio des phonèmes par temps (PTR : phonation per time ratio) : ce score peut être déterminé en divisant la durée total requise pour produire la parole en incluant les phonèmes silencieux par la durée total de la parole en négligeant les phonèmes silencieux.

Dans le tableau 3.2, nous résumons les scores utilisés ainsi que les principaux travaux où ils ont été introduits.

Dans ce qui suit, nous allons tenter de résumer, non exhaustivement, différentes approches et systèmes réels qui existent à l'heure actuelle dans le domaine de l'évaluation automatique de la prononciation pour pouvoir se positionner par rapport à ces travaux.

3.5 Les travaux connexes

Bernstein et al. [27] ; ont été parmi les premiers chercheurs qui ont utilisés les résultats, sous forme de score, de l'application du processus de reconnaissance automatique de la parole basé sur les HMMs pour évaluer la prononciation. Dans ce cas, le score de la durée du segment de la parole ainsi que le logarithme de vraisemblance correspondant ont été utilisés comme résultats de classification. Les chercheurs dans [27] ont utilisés

TABLE *3.2* – Résumé des scores pour l'évaluation de la prononciation

Auteurs	Score automatique	Système (logiciel)	Langage cible
[27]	SFS distances	SRI	Anglais
[34]	- Durée d'un segment - Vraisemblance d'un segment	SRI	-
[28]	- Log likelihood - Score basé temps - Erreur de classification d'un phonème - Duré d'un segment	VILTS	Français
[29]	- Concordance spectrale - Durée d'un phonème - Durée d'un mot - Vitesse de la parole	EduSpeak	Espagnol
[35]	Likelihood Ratio	-	Allemand
[33]	Goodness of Pronunciation	PLASER	Anglais
[36]	Goodness of Pronunciation	PLASER	Anglais
[37]	probabilité log likelihood des phonèmes	AZELLA	Anglais

la technologie de la reconnaissance automatique de la parole pour attribuer un score à la prononciation des étudiants Japonais voulant apprendre la prononciation en Anglais (via un corpus de collection des conversations téléphoniques). Le système proposé invite l'apprenant à prononcer le texte affiché (approche dépendante de texte : *Text-dependent*). De ce fait, l'approche ne pourra pas être généralisée vu que de nouveaux essais peuvent avoir lieu, chose qui requiert une mise à jour régulière du corpus de donnée.

Plus tard, *Franco* et son équipe ont fait plusieurs expérimentations pour déterminer une collection de scores qui permettent de mimer le jugement de l'expert humain [29] [28] [26]. Leur recherche effectuée était dans le cadre du développement d'un système nommé : *EduSpeak*. Ce système avait pour but de réaliser des segmentations en phonème du signal de la parole. Ces segmentations ont été utilisées par la suite pour produire des scores de la prononciation à la fin de chaque session d'apprentissage. Pour générer des scores fiables à un niveau phonétique, des scores issus d'une classification des segments et des scores de la durée de chaque segment de la parole ont été utilisés.

Le système *PLASER* (Pronunciation Learning via Automatic SpEech Recognition) ; un système d'apprentissage de la prononciation via la technologie de la reconnaissance automatique de la parole est un autre outil multimédia qui a pour but d'assurer l'apprentissage de la prononciation correct de l'Anglais au apprenants Chinois dont la langue d'origine est la cantonaise Chinois [36]. Les modèles acoustiques utilisés comme référence pour la comparaison ont été construits en utilisant le corpus de données commercialisé

TABLE *3.3* – Exemples de systèmes CAPT

Authors	System (software)	Mother language	Target language
[27]	SRI	Japanese	English
[28]	VILTS	American-English	French
[38]	EduSpeak	-	Spanish
[35]	-	-	Dutch
[36]	PLASER	-	English
[39]	-	Greek	English
[40]	-	Malay	Arabic
[41]	-	Malay (independent)	Arabic
[37]	AZELLA	-	English

TIMIT avec un ensemble de phrases extraites de ce corpus prononcées par des locuteurs d'origine Anglais-Américains. Deux types d'exercices ont été proposés : des exercices à pair minimal et de exercices de mots. À la fin de chaque évaluation de prononciation le système renvoie à l'apprenant un feedback informatif qui est basé sur le GOP (Goodness Of Pronunciation) en colorant les phonèmes qui ont été mal prononcés.

Concernant la langue Arabe, récemment, les auteurs de [41] ont cités que « la recherche sur le traitement de la parole concernant la langue Arabe vient de commencer, pour cela il n'existe aucune recherche ou système trouvé pour quantifier la prononciation de la langue Arabe sous forme de scores. Alors, des études sont requises ». Dans [41] les auteurs ont présentés un travail qui avait pour but de fournir un outil d'aide aux enseignants Malaisiens pour apprendre la langue Arabe rapidement afin d'améliorer leur prononciation, en se focalisant sur une compréhension basée sur le couple entendre/parler. Le système construit est basé sur les modèles HMM. Au niveau de la phase d'attribution des scores à la prononciation, les prononciations ont été notées de 1 à 4 par les experts humains, tandis que le système basé-HMM attribuait des notes, sous forme de scores, aux prononciations en se basant sur le logarithme de vraisemblance fourni comme résultat de la phase de décodage HMM durant la reconnaissance automatique de la parole. Le système proposé a achevé une précision de 87.61% où la phase de l'apprentissage a été effectuée sur des données natives et non natives.

Le tableau récapitulatif 3.3 illustre quelques systèmes réels d'évaluation automatique de la prononciation de différentes langues cibles.

Les auteurs dans [42] ont effectué quelques expérimentations qui touche la détection des erreurs de prononciation en Arabe. Le but était de fournir un système basé sur l'utilisation de la technologie de la reconnaissance automatique de la parole qui détecte les erreurs de prononciation des apprenants Arabe.

D'autres systèmes qui se positionnent dans le contexte de l'apprentissage des langues,

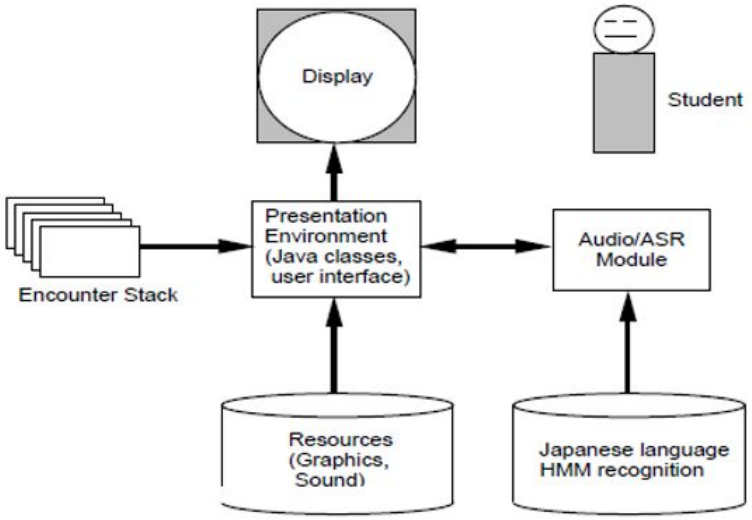

FIGURE *3.3* – Diagramme du système Subarashii, d'après [44]

proposent des approches novatrices, nous retrouvons :

Subarashii [43] [44] qui est un système expérimental d'apprentissage de la langue Japonaise parlée basé sur le dialogue. Subarashii prône une stratégie pédagogique qui stipule que cet apprentissage est mieux perçu s'il se fait dans un contexte applicatif. Ainsi, Subarashii est conçu pour comprendre ce que dit l'apprenant en Japonais (dans un contexte restreint) et il répond en Japonais parlé et non sous forme écrite. Dans ce système (voir figure 3.3) précurseur dans les systèmes interactifs pour l'enseignement des langues parlées (en anglais ISLE Interactive Spoken Language Education), on cherche à relier l'enseignant dans les tâches répétitives auprès des apprenants et on réserve à l'enseignant le suivi de tâches plus subtiles et plus créatives. Subarashii offre à l'apprenant un ensemble de situations qu'un étudiant visitant le Japon peut vivre et construit autour de cela des rencontres avec les Japonais ; les rencontres sont agencées sous forme d'un jeu d'aventure. L'interaction est ainsi guidée par le but du dialogue et les réponses de l'apprenant sont alors contraintes. Le système de reconnaissance accepte ou rejette un mot selon des seuils prédéfinis.

Bien que les différents modules soient écrits en Java, la partie réservée aux entrée/sortie audio fût écrite en C.

Dans [45], les auteurs présentent leur système *TLCTS* (pour Tactical Language and cultural Training System) qu'ils ont fait évoluer à partir du système *TLTS* [45] [46] [47] [48] vers une dimension culturelle du langage. C'est l'exemple riche d'un système multimédia pour l'apprentissage de langue. L'apprenant est submergé dans un monde virtuel en 3D. Chaque cours est conçu sous la forme d'un jeu et est basé sur un scénario associé à la mission du jeu où l'apprenant joue un rôle dans le contexte culturel du langage cible. Notons que le système a été essentiellement développé pour un usage militaire, en agissant

FIGURE *3.4* – Exemple d'un dialogue dans une mission

avec des acteurs dans l'environnement en utilisant la parole et des communications non verbales. Ainsi, la figure 3.4 nous montre une capture d'écran où le langage cible est le « *Dari* » ; une variété du la langue Persan, parlée en Afghanistan.

La transcription de la conversation apparaît en haut de l'écran. La reconnaissance de la parole est effectuée en utilisant la toolkit HTK « Hidden markov models ToolKit » augmenté avec modèles à chaine de bruit afin de capturer les erreurs de prononciation. Différents types d'erreurs sont recensés (grammaticale, sémantique, etc.) et pris en charge par le système de reconnaissance.

Toujours dans la même optique, *Raux et Eskenazi* [49] proposent un système d'apprentissage de langue basé sur le dialogue dans des situations réelles où on essaie d'immerger l'apprenant, telles que la réservation d'un billet d'avion. Trois sous tâches essentielles dans ce système sont : La compréhension du langage parlé, la gestion de la conversation et la génération du dialogue. En particulier, la tâche de compréhension est réalisée en deux phases : D'abord, la parole émise par l'apprenant est transcrite en une suite de mots par le système de RAP, ensuite un module de compréhension du langage naturel analyse l'hypothèse de la transcription en une représentation sémantique interne sur laquelle le système peut effectuer un raisonnement. Le système de RAP est basé sur CMU. Les auteurs soulignent en particulier, la difficulté de reconnaître des mots en se basant sur les modèles acoustiques construit à partir d'enregistrements de locuteurs natifs du langage. Pour résoudre ce problème les concepteurs du système ont prévu d'incorporer des heures d'enregistrements d'Anglais issu de locuteurs Indiens, Japonais, Allemands et Chinois.

Ceci devrait réduire les erreurs de reconnaissance en prenant en considération les différents accents. Au côté des modèles acoustiques, les concepteurs ont prévu d'adapter le module linguistique pour prendre en charge certaines erreurs de concordance syntaxique ou autre. Dans un contexte de dialogue libre, le système propose un feedback correctif à l'apprenant, comme l'illustre la figure 3.5.

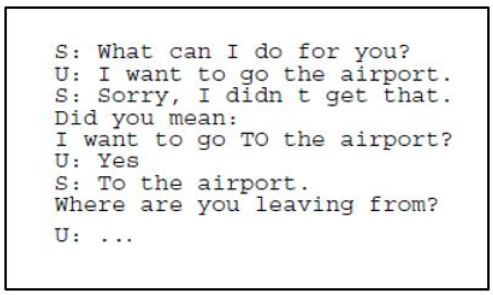

FIGURE *3.5* – Exemple de dialogue avec feedback correctif

Chao et al. [25] ont créé un jeu de traduction basé web pour apprendre la langue Chinoise avec des exercices répétitifs pour acquérir la grammaire et le vocabulaire. La figure 3.6 montre une capture d'écran d'une session de traduction.

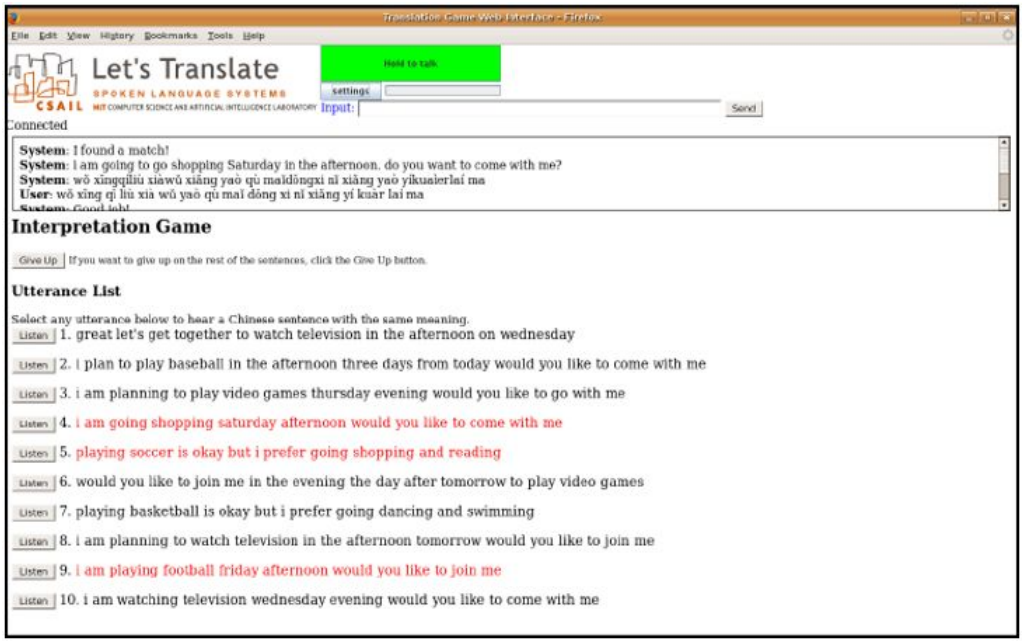

FIGURE *3.6* – Session de traduction

Le système nommé *DISCO* (Development and Integration of Speech technology into COurseware for language learning) [50] est un système Hollandais conçu principalement pour fournir des feedbacks sur la prononciation, la morphologie et la syntaxe. Ce système exploite les erreurs communes de morphologie et de syntaxe commises par les apprenants

de langue non maternelle Néerlandais. Le système DISCO dirige les dialogues en élicitant des réponses très consistantes aux questions ; il utilise un processus à deux étapes pour reconnaitre la parole. Dans une première étape il détermine le contenu de la réponse de l'apprenant en augmentant le *FST* (Finite State Transducer) du modèle de langage. Dans une seconde étape, il analyse cette réponse pour vérifier si elle est correcte ou non.

Le *SayBot Player* [51] est un système pour enseigner la langue Anglaise aux locuteurs d'origine Chinoise. Le système comprend deux composantes : un système de reconnaissance de la parole et un module d'évaluation de la prononciation. Ce système maintient un flux de dialogue similaire à celui des enseignants en utilisant une architecture FST comme dans le système DISCO. Les scores attribués à la prononciation sont fournies en utilisant les HMMs et plus spécifiquement le score logarithme de vraisemblance et les mesures qui sont basées sur la durée de la parole. Les erreurs commises durant le dialogue ont été classées en quatre catégories : correct (tous les mots sont corrects et le score de prononciation est bien), erreur prédéfinie (le score de prononciation est bien, mais la phrase est reconnue parmi un ensemble d'erreurs prédéfinis), mal prononciation (les mots reconnus où la prononciation était mauvaise) et général (le système n'arrive pas à comprendre ce que l'apprenant essaye de dire).

3.6 Approches pour l'évaluation de la prononciation

Le but du processus d'évaluation est de fournir une note ou une appréciation à l'apprenant, qu'un expert humain lui aurait attribuée. Pour cela, deux approches sont possibles : L'approche par estimation et l'approche par classification [52].

La première approche peut être vue comme un problème de prédiction, où on essaye d'estimer la valeur du score qu'un expert humain aurait donné en utilisant un ensemble de variables de prédiction [52]. En général, cette estimation est faite grâce à une fonction de régression non linéaire.

La seconde approche permet de passer des scores de la machine à une appréciation fournie par l'expert humain, en définissant un ensemble de N classes. N étant l'échelle de notation. Chaque prononciation sera alors classée comme appartenant à l'une des N classes, où les classes sont les valeurs discrètes de notes attribuées par les experts.

De plus, les scores précédemment définis peuvent être combinés de diverses façons. Un exemple pratique peut être trouvé dans [53], où les auteurs supposent que la note finale peut être prédite grâce à un sous ensemble de scores. Pour cela, ils ont utilisé une fonction linéaire de régression, des réseaux de neurones et des arbres de régression.

3.6.1 Les approches basées sur la classification

Les auteurs de [54] ont développé des classifieurs basés sur l'utilisation des *LDA* (Linear Dynamic Analysis) et les arbres de décision pour les trois erreurs de prononciation les plus célèbres de l'Allemand commises par les locuteurs non natifs. Les auteurs ont examiné les propriétés acoustiques des erreurs de prononciation en question afin d'en extraire les caractéristiques acoustiques qui ont été utilisées pour l'entrainement des classifieurs et qui sont capables d'identifier les phonèmes mal prononcés par les locuteurs non natifs.

Une autre approche alternative a été proposée dans [55] au niveau de laquelle les informations articulatoires ont été utilisées pour améliorer la détection automatique des erreurs typique de la prononciation au niveau phonème commises par les locuteurs non natifs. Pour cela, une nouvelle version des HMMs, adaptée pour l'évaluation automatique de la prononciation, a été présentée. Les informations articulatoires concernant les caractéristiques qui ont été extraites depuis les résultats de la reconnaissance de concaténation articulatoire sur 8 flux représentatifs et par le calcul des scores de confiances basé sur l'articulation multidimensionnelle.

Dans [56], des règles d'erreurs de prononciation ont été définies et regroupées dans un arbre de décision pour une classification. Un seuil a été attribué pour chaque classe dans l'arbre de décision ce qui a permis à l'algorithme de réaliser des résultats de détection prometteurs.

Les auteurs de [57] ont modélisé le problème en utilisant les SVM et des modèles d'espace de prononciation pour améliorer la performance de la détection des erreurs de prononciation. En gros, chaque phonème a été modélisé par plusieurs modèles acoustiques en parallèles pour représenter les variations de la prononciation d'un phonème donné.

Les auteurs de [58] ont présenté une méthode quant à l'absence d'une base de données dédiée aux systèmes d'enseignement de prononciation assisté par ordinateur CAPT. Cette méthode consiste à déterminer automatiquement les seuils du célèbre algorithme de classification des phonèmes GOP (voir la section 3.4.2.2 pour plus de détails concernant le GOP) selon leurs degré d'exactitude. La distribution de deux scores pour une prononciation erronée, obtenue pour chaque phonème, a été calculée en insérant des erreurs contrôlées dans le dictionnaire qui contient la transcription des prononciations de telle sorte que chaque phonème est aléatoirement remplacé par un phonème du même groupe. Ces groupes des phonèmes ont été obtenus par une classification phonémique en utilisant les arbres de régression. Après l'obtention des deux distributions de chacun des deux scores calculés, la mesure *ERR* (Eqaul Error Rate) de chaque pair de distribution a été calculée et utilisée comme un seuil pour chaque phonème.

La recherche de *Minematsu et al.* [59] propose une approche complètement différente en modélisant les sons de la prononciation comme étant des distributions dans l'espace

de fréquence relative à d'autres distributions de sons au niveau langage. Cela a été pris en considération par *Jakobson* [60] qui a fait valoir que l'étude des sons d'un langage doit prendre en charge la structure du système sons dans son ensemble. La structure définie par *Minematsu et al.* a été ensuite utilisée pour définir une métrique de distorsion qui mesure la différence entre les structures phonétiques de deux ensembles de locuteurs : des Américains natifs en Anglais et des apprenants Japonais de la langue Anglaise. Cet indicateur a montré sa consistance en montrant une corrélation avec l'évaluation de la maitrise de la prononciation, et cette corrélation se maintient même dans le cas où le modèle acoustique des locuteurs non natifs est comparé à plusieurs modèles acoustiques des locuteurs natifs.

Les auteurs de [61] ont combinés les scores obtenus par l'application des logarithmes des probabilités des *HMM* et les scores *GMM* en utilisant une régression non linéaire afin de rendre la fonction d'attribution de scores proche du jugement d'un expert en prononciation. Dans ce travail, les probabilités logarithmiques ne sont pas directement utilisées par la fonction d'attribution de scores. Les probabilités logarithmiques sont utilisées pour classer l'ordre des syllabes prononcées correctement contre 410 autres syllabes de la langue Chinoise. Le classement des syllabes est ensuite utilisé pour calculer le score de chaque syllabe. Les scores obtenus par le biais des *GMM* sont utilisés de la même manière. Ensuite, une régression non linéaire est utilisée pour optimiser des différents paramètres dans le but de combiner ces scores pour en garder un qui sera proche au jugement de l'expert en prononciation.

Une approche décrite dans [62] utilise des probabilités log-postérieur extraites par l'application d'un alignement forcé avec les HMM pour classifier la qualité des syllabes en utilisant les séparateurs à vaste marge SVM. Les résultats de la classification sur un grand nombre de syllabes produit un score final sur la capacité de la prononciation d'un locuteur. Ce score est corrélé avec le corpus de scores Putonghua Shuiping Kaoshi '*PSK*' qui représente un corpus des locuteurs Chinois de différents dialectes.

Un autre exemple de méthode d'attribution de score qui n'utilise pas explicitement les caractéristiques extraites à partir des HMM peut être trouvé dans [63]. Les auteurs ont pu trouver une corrélation positive entre les mesures d'élagages des syllabes par seconde, le ratio de la différence entre le nombre total des syllabes et les syllabes inutiles par rapport à la durée total et le ratio des syllabes non accentuées par rapport aux syllabes accentuées. Un aspect relatif à cette étude est que les auteurs ont pris soin de recueillir des évaluations des enseignants qui ont été formées spécifiquement dans le cadre de *CEFR* (Commom European Framework of Reference) pour évaluer la prononciation. Cela comprenait de nombreux éléments spécifiques d'évaluation de volume, de pitch du son, de la qualité des voyelles, de la qualité des consonnes, épenthèse, élision, le stress, le stress de la phrase, le rythme, l'intonation, débit de parole, la fluidité, lieu de pauses et la fréquence des pauses.

Neumeyer et al. [28] ont examiné la prononciation de la langue Française parlée par des locuteurs d'origine des états unis. Dans cette étude, les chercheurs ont enregistré des échantillons de parole spontanée prononcés par 100 locuteurs Français natifs et 100 locuteurs Américains. Ils ont étudié 4 méthodes différentes pour attribuer un score à la prononciation selon deux niveaux : niveau locuteur et niveau phrase. Les corrélations ont été calculées entre les scores, qui comprenaient le logarithme de vraisemblance HMM -la classification des segments-, la durée de chaque segment, les scores liés à la durée qui sont déterminés par différents algorithmes et les scores attribués par les évaluateurs humains. Initialement, ils ont trouvé que la normalisation des scores générés par les HMM (les probabilités de vraisemblance et postérieurs) n'a pas une bonne corrélation avec les évaluations humaines sur une échelle de 1 à 5 (1 était non natif et 5 pour natif). Tous ces scores, à l'exception de ceux qui sont basés sur la durée, ont été calculés à ce qu'ils pensaient être des corrélations inacceptables tant au niveau phrase et le niveau locuteur. Les auteurs ont pu, dans une autre référence [64], améliorer la corrélation des scores calculés à l'aide des HMM au niveau locuteur en utilisant la normalisation des scores qui représentent les probabilités log-postérieurs au lieux des scores basés sur le logarithme de vraisemblance.

Dans d'autres travaux comme [65] [52] [66], les chercheurs ont concentrés leurs efforts sur le niveau phrase et locuteur pour l'évaluation de la prononciation en attribuant des scores a certains phonèmes. Les auteurs ont introduits une nouvelle méthodologie pour la détection des erreurs de prononciation avec laquelle ils ont comparés la probabilité log-postérieur résultant d'une méthode appliquée aux modèles natifs avec une approche caractérisée par un modèle dual dans laquelle un modèle de phonème représente la prononciation correcte et l'autre représente la mauvaise prononciation.

3.6.2 Autres approches

Dans [67] une approche pour la détection des erreurs de prononciation a été développée. Cette approche est basée sur l'utilisation des connaissances linguistiques. Ces dernières ont été obtenues grâce aux erreurs de prononciation les plus communs des locuteurs non natifs et l'espace de prononciation conçu par l'utilisation des vecteurs des log-probabilités postérieur. Un classifieur SVM a été ensuite appliqué pour détecter les erreurs de prononciation des apprenants de la langue cible Mandarin Chinois.

Les auteurs de [68] ont pu inclure l'intelligibilité des apprenants d'une langue cible dans un système de diagnostic des erreurs de prononciation. Les auteurs ont pu construire un algorithme probabiliste afin d'extraire l'intelligibilité à partir du ratio d'erreurs et aussi définir une fonction de priorité d'erreur dont le but était d'indiquer quelle erreur influence le plus l'intelligibilité.

3.7 La détection des erreurs de prononciation dans des systèmes réels

Plusieurs systèmes CAPT ont été développés pour améliorer l'apprentissage et la maitrise d'une langue étrangère L2. Parmi ces systèmes *FLUENCY* [69] qui utilise le système de la reconnaissance automatique Sphinx 2 pour détecter les erreurs de prononciation de L2. Le but de ce système est d'aider les locuteurs non natifs à améliorer leur prononciation en pratiquant et interagissant avec le système. Les erreurs de prononciation détectées sont ensuite analysées et un feedback est retourné à l'utilisateur. Pour chaque phonème/mot prononcé par l'utilisateur, *FLUENCY* effectue une évaluation, en vérifiant la durée par rapport à une base de référence, si le phonème était plus court ou plus long par rapport au modèle de référence. Le but de *FLENCY* ne se limite pas seulement à localiser l'erreur de prononciation ; il met à disposition de l'apprenant la possibilité d'entendre le phonème et des instructions sur comment bien placer les articulations pour améliorer la performance. Le système Sphinx 2 a été aussi utilisé pour mesurer les informations prosodiques et détecter les erreurs phonétiques commises par des locuteurs non natifs de la langue Anglaise. Cette recherche a été utilisée pour créer un prototype de tuteur de langage [70] qui était basée sur 5 principes articulés par : la production de grandes quantités de parole, l'exposition a de nombreux exemples de la parole natif, l'accent au début sur des facteurs prosodiques et faciliter les taches quant à un environnement d'apprentissage. Une clé importante dans le système était l'utilisation des techniques de sollicitation dans le but de prédire les phrases qui pourraient être utilisées pour une reconnaissance basée sur un alignement forcé, contrairement à d'autres systèmes [71].

Les auteurs de [72] ont utilisés la reconnaissance discrète des mots basée sur les modèles (la reconnaissance des modèles) pour évaluer les apprenants des langues Espagnole et Chinoise. Une analyse segmentale a été effectuée pour identifier les erreurs de prononciation des phonèmes spécifiques. Ceux-ci ont été ensuite utilisés pour créer un système permettant de pondérer l'importance de différentes erreurs. Finalement, une interface basée application jeux a été ajoutée [73] dans le but de fournir un feedback sur la qualité globale de la prononciation. Un aspect intéressant de cette recherche est que la reconnaissance basée sur les HMM a été comparée avec une méthode de modélisation. Les auteurs ont trouvé que, tandis que le module de reconnaissance HMM avait de meilleurs résultats de reconnaissance, la reconnaissance des modèles arrivait à bien distinguer entre les paires minimales.

Une approche a été proposée par [74] qui consiste à combiner les résultats d'un alignement forcé de la langue Anglaise prononcé par des apprenants Koréens, avec les transcriptions manuelles phonétiques des phonéticiens experts. Une analyse phonologique détaillée a été effectuée pour obtenir un ensemble de règles d'augmentation qui modélisaient les

prononciations communes effectuées par les étudiants. Ces règles marquaient les erreurs de prononciation phonétique dans un contenu parlé (phrases, mots...) et déclenchaient des messages sous forme de feedback aux étudiants. Cette approche a été ensuite étendue pas *Harrison et al.* [75].

Une nouvelle approche proposée dans [76] [77] combine les probabilités log-postérieurs des trames sonores, les probabilités log-postérieurs des phonèmes et le score de classification des formants résultant de l'extraction des caractéristiques des images en utilisant la fonction *Gabor* pour évaluer la qualité des voyelles de la langue Mandarin. Trois techniques ont été expérimentées pour combiner les scores : une régression linéaire dans le but d'approximer un score d'un expert humain, une estimation de probabilité conjointe et un réseau de neurone. Le réseau de neurone, en utilisant ces trois derniers caractéristiques, a achevé une corrélation élevée, en comparant avec les probabilités log-postérieurs, avec le score attribué par l'expert en prononciation.

Dans [57] les SVMs ont été utilisés pour détecter les erreurs de prononciation effectuées au niveau des phonèmes en utilisant les probabilités log-vraisemblance produites par un treillis HMM. Ensuite, un ratio dépendant des phonèmes a été fixé pour balancer la précision et le rappel des erreurs de prononciation. Contrairement à la plupart des méthodes HMMs qui utilisent les GMMs pour modéliser les prononciations des phonèmes, cette recherche utilise un modèle appelé « le modèle espace de prononciation » (Pronunciation Space Model : PSM). Les auteurs ont été motivés par l'observation que plusieurs substitutions de phonèmes ne sont pas des substitutions complètes d'un phonème par un autre, mais c'étaient des substitutions d'un phonème partiellement changé par un son qui peut ne pas apparaitre dans la langue cible.

Les auteurs de [78] ont concentré leurs efforts sur l'utilisation de plus d'une voix native afin de trouver le « meilleur locuteur » (le locuteur dont la prononciation est la plus proche possible par rapport au modèle de référence) qui pourrait être imité par l'utilisateur dans le but d'améliorer ses compétences de prononciation.

Le système *ISLE* [79] est fondé sur un large ensemble d'exercices questions-réponses dans lesquelles la réponse de l'utilisateur est construite à partir d'un petit ensemble prédéfini. Il inclut le moteur de la reconnaissance automatique de la parole qui reçoit la parole dans le but de localiser les erreurs de prononciation. Pour chaque phonème ou un mot complet, un score de confiance est calculé à la base du logarithme de vraisemblance du chemin reconnu, la probabilité du meilleur état émetteur du modèle. Le phonème/mot examiné est ensuite considéré comme étant mal prononcé si son score de confiance ne vérifie pas un certain seuil.

Les auteurs de [80] ont introduits un système nommé *HAFSS*. Il s'agit d'un logiciel commercialisé dont le but est d'enseigner la récitation correcte du *Coran*. Ce système

était dédié aux locuteurs non natifs pour apprendre la prononciation correcte en Arabe pour réciter le Coran. Les objectifs principaux de ce système étaient : enseigner la récitation correcte du Coran, évaluer la qualité de la récitation des apprenants, produire des feedbacks à la fin de chaque session d'apprentissage afin d'aider les apprenants à localiser les lettres qui ont été mal prononcés. Le système *HAFSS* se compose de 6 composantes principales : les modèles HMMs de vérification, l'adaptation du locuteur, le générateur des hypothèses de prononciation, l'analyse du score de confiance, l'analyse de la durée des phonèmes et à la fin se trouve le générateur de feedbacks. Le score de confiance attribué à la prononciation selon *HAFFS* est basé sur le logarithme de vraisemblance. Lorsque le système détecte des erreurs de prononciation avec un faible score de confiance il réagit avec des réponses alternatives : omettre la déclaration de l'erreur (éviter de décourager l'apprenant et assurer la continuité de l'apprentissage), inviter l'apprenant à répéter la prononciation, signaler l'existence d'une erreur non identifiée et inviter l'apprenant à répéter la prononciation (ce qui est bien pour les apprenants experts), signaler l'erreur de prononciation la plus probable.

Dans [81] les auteurs ont présenté le système *IELS* (Interactiv English Learning System) ; un système d'apprentissage de la prononciation assisté par ordinateur destiné à apprendre l'Anglais pour des apprenants Chinois dont la langue maternelle est le Mandarin. Le système fournit des feedbacks concernant les mauvaises prononciations des phonèmes, mots, stress lexical et un score sur la qualité globale de la prononciation de l'apprenant. Le système se base sur une architecture client/serveur au niveau de laquelle le client fournit une interface pour l'utilisateur et des fonctions d'entrées/sorties audio. Quant au serveur, il prend en charge le traitement de la parole, y compris la reconnaissance de la parole basée sur les HMMs, la détection du stress (stress detection) basée sur les SVMs, et l'association d'un score à la prononciation. En effet, le système IELS est conçu à la base d'une architecture client/serveur selon les raisons suivantes : la reconnaissance automatique de la parole est coûteuse en terme de calcul et demande/occupe, par conséquent, beaucoup de mémoire/processeur. Pour ce faire, les auteurs dans [81] ont déployé un serveur afin de manipuler le calcul du traitement de la parole, et un client pour les entrées/sorties et pour ce qui est de l'affichage.

Les auteurs dans [82] ont étudié l'utilisation d'un modèle statistique de la durée de phonème pour permettre la séparation des énoncés intacts (corrects) de ceux qui sont endommagés (erronés) dans un système CAPT. Le système CAPT proposé ; *CHELSEA* ; effectue un alignement forcé entre l'énoncé en entrée et la transcription canonique du texte que l'apprenant est invité à prononcer. Cette transcription est obtenue grâce à un dictionnaire de recherche (lookup dictionnary). Les énoncés corrects ou intacts contiennent un contenu parlé qui correspond au texte à prononcer. Pour ces énoncés, le système proposé dans [82] effectue une analyse phonétique détaillée et génère des feedbacks correctifs pour

mettre en valeur l'occurrence des erreurs phonétiques. Selon *CHELSEA* les énoncés corrompus (endommagés) proviennent de la non-maitrise, des enregistrements tronqués, ou d'un contenu parlé qui ne correspond pas au texte à prononcer (le texte que l'apprenant est invité à prononcer). Pour ces raisons le feedback approprié est d'inviter l'apprenant de refaire l'enregistrement encore une fois. Selon [82] un mécanisme de filtrage pour les énoncés intacts en entrée est développé par le biais d'une modélisation de la durée de phone.

3.8 Évaluation de la prononciation par des experts humains

L'un des problèmes que l'on observe lorsque nous souhaitons évaluer une prononciation est la disparité entre les notes des experts. Or, si nous souhaitons de plus automatiser cette évaluation, nous avons besoin de ces appréciations d'experts pour calibrer nos scores. Ce point a été largement abordé dans une étude dans [35]. Le but principal de cette étude est de faire le point sur l'important rôle que jouent les jugements des experts humains dans le domaine de l'évaluation automatique de la prononciation et la disparité de leurs notes.

3.8.1 Problèmes liés à l'évaluation pas les experts humains

En ce qui est du développement automatique des instruments pour le test de langue il est vite apparu que pour certaine compétences l'automatisation sera plus facile que pour d'autres. Généralement, quatre compétences sont requises à base de quelques dimensions comme : le mode (oral vs écrit) et la direction (réceptive vs productive). Depuis que dans le test des compétences réceptives, il est possible d'utiliser des réponses qui sont faciles à noter, développer des tests automatiques pour ces compétences est devenue une tâche faisable. Pour les compétences productives, d'un autre côté, les tests automatiques sont difficiles à développer à cause de la nature des entrées. Ainsi, dans le cas de la parole, la direction et le mode se conspirent pour effectuer le test automatique.

En plus de ces difficultés, plusieurs méthodes pour l'évaluation de certaines compétences orales comme la prononciation ont été proposées [27] [28] [64]. La plupart de ces systèmes utilisent les technologies récentes qui sont principalement basées sur la reconnaissance automatique de la parole. Mais il semble primordial pour n'importe quel système, qui a l'intention de tester ou améliorer la prononciation, de se référer à quelques standards qui sont basés sur les jugements des experts humains, une importance qui ne peut pas être sous-estimée, comme les scores qui sont attribués par les experts humains représentent ce que les techniques d'évaluation automatique tentent de produire.

La recherche sur l'évaluation de la prononciation a révélé que les scores globaux de la qualité de la prononciation pourront être affectés par la variance des caractéristiques de la parole [83]. La parole non-native peut dévier de la parole native selon divers aspects tel que la maitrise, la structure syllabique, l'accent, l'intonation et la qualité segmentale. Quand les experts natifs sont invités à attribuer des scores à la prononciation des locuteurs non natifs, leurs scores sont souvent affectés par plus que ces aspects. La recherche sur la relation entre l'évaluation sous forme de scores des locuteurs natifs attribués aux locuteurs non natifs et la déviance par rapport aux différents aspects de la qualité de la parole a révélé que chaque région affecte le score complet à différents niveaux [83].

Cela suggère que l'évaluation globale de la qualité de prononciation sous forme de score attribué par les experts humains a une structure complexe [35], chose qui pourrait être une problématique quand de tels scores sont utilisés comme une référence pour des mesures automatiquement générées sur la qualité de la prononciation, car on ne sait pas exactement l'aspect visé des scores attribués par les experts humains. Pour cela, quelques indices doivent être pris en considération pour développer une application qui soit un peu plus proche du jugement de l'expert humain, comme : qu'est-ce que les experts ont l'intention d'évaluer, et surtout quelle serait l'influence de leur jugements. Pour faire face à cette situation, il semble important que plusieurs classements spécifiques de la qualité de la prononciation doivent être collectés avec des classements globaux pour une meilleure compréhension de l'évaluation de la prononciation par les experts humais.

D'autres problèmes avec les scores de prononciation attribués par les experts humains ont été relevés par [28] [64] sur le fait de ne pas prendre en considération des segments de sons spécifiques (comme les sons des essais ou les sons rares). Dans cette étude, il était demandé aux experts humains d'assigner un score global sur la qualité de la prononciation pour chaque ensemble de phrases prononcées par chaque locuteur (évaluation au niveau phrase). Les scores de toutes les phrases, pour chaque locuteur, ont été pondérés afin d'obtenir un score global de locuteur (évaluation au niveau locuteur). Même si cette procédure peut s'avérer logique à une première vue, il y avait bien quelques problèmes avec cette méthode [35].

Les scores assignés par un expert humain à différentes phrases prononcées par un locuteur et ceux assignés au niveau du même locuteur peuvent être différents comme étant une fonction segmentale [84] [35]. Par exemple, si des sons de type rares sont présents au niveau d'une prononciation d'une phrase, le score attribué pour cette phrase pourrait être considérablement inférieur à ceux attribués à d'autres phrases prononcées par le même locuteur et qui ne contiennent pas ce type de sons. Vu la possibilité de la présence des sons stigmatisant, les scores de prononciation collectés au niveau locuteur pourront être bien inférieur à ceux qui résulteraient en moyennant les différentes phrases prononcées par le même locuteur. En d'autres termes, le score moyen ne serait, peut-être, pas en mesure de

refléter l'effet des sons rares au même degré de ceux prononcés auquel un score global au niveau locuteur a été attribué. Cela conduit vers la suggestion que si les chercheurs se sont intéressés aux scores de prononciation au niveau locuteurs, ils devraient avoir l'aide des experts humains à savoir écouter les fragments de sons qui contiennent tout l'ensemble des phonèmes de la langue dont l'évaluation de la prononciation est requise [35], ce qui sera le cas, dans ce qui suit, concernant l'évaluation de la prononciation en Arabe.

3.8.2 Autour de l'évaluation de la prononciation en Arabe par les experts

Avant de débuter nos expérimentations relatives à l'évaluation automatique de la prononciation en Arabe, nous avons souhaité reprendre une partie du protocole de test réalisé dans les travaux de [35], mais dans le cadre de la langue Arabe. En effet, nous allons investiguer la corrélation intra et inter-experts sur un ensemble de mots et de phrases. Pour cela nous avons demandé à des experts d'évaluer un ensemble de prononciation de mots en Arabe. Les locuteurs impliqués dans cette expérimentation sont au nombre neuf (9) avec des niveaux de maîtrise de la langue différents. Le but de cette expérimentation c'est d'avoir l'aide des experts humains pour évaluer une prononciation et ainsi de vérifier l'habilitée d'une approche automatique d'évaluation à avoir le comportement d'un expert humain pour juger la prononciation.

Un groupe de trois experts humains a été choisi pour cette expérimentation. Il était demandé aux experts d'entendre attentivement les enregistrements qui leur avaient été présentés afin de pouvoir, par la suite, attribuer une note qui sera comprise entre 0 et 10. Cet intervalle a été choisi pour faciliter la compréhension du feedback quant à l'application de l'évaluation automatique. En plus de la note comprise entre 0 et 10, une appréciation est rajoutée à la fin pour que le feedback informatif soit simple et compréhensible. Le processus d'évaluation par les experts humains a été répété trois fois chacun dans une journée séparée. Le but ici était de pouvoir mesurer la relation entre les notes attribués par les experts humains et ce selon deux points : la relation de l'évaluation inter-experts et la relation de l'évaluation intra-experts.

L'appréciation attribuée à une prononciation peut appartenir à une parmi ces trois classe : *bonne*, *moyenne* ou *mauvaise* prononciation selon la note attribuée par les experts humains. Les coefficients de corrélation entre les évaluations des experts humains sont ensuite déterminés pour vérifier le degré de similitude.

Les deux tableaux 3.4 et 3.5 suivants résument la moyenne des coefficients de corrélation des notes des experts humains obtenus pour les neuf locuteurs sur un ensemble de seize (16) mots prononcés en Arabe. Le tableau 3.4 montre les résultats de corrélation intra-experts des notes attribuées. Le but ici est de vérifier si une réévaluation d'un expert

TABLE *3.4* – Les coefficients de corrélation intra-experts obtenus

	Corr. Intra Expert 1	Corr. Intra Expert 2	Corr. Intra Expert 3
locuteur 1	-0.337052649	0.447619784	0
locuteur 2	0.491322081	0.897313616	0.547790352
locuteur 3	-0.264485942	0.760609054	0.583819382
locuteur 4	0.845058842	0.902232359	0.722748487
locuteur 5	0.937301353	0.909637228	0.94725552
locuteur 6	0.639064442	0.452910814	0.532692181
locuteur 7	0.751028958	0.764758128	0.340326592
locuteur 8	0.941682305	0.855748918	0.667156154
locuteur 9	0.915035672	0.903274001	0.809770235
Moyenne	0.546550563	0.766011545	0.572395434

TABLE *3.5* – Les coefficients de corrélation inter-experts obtenus

	Corr. Inter Experts 1,2	Corr. Inter Experts 1,3	Corr. Inter Experts 2,3
locuteur 1	-0.106576139	-0.359210604	-0.118678166
locuteur 2	0.421448457	0.436398128	0.60298195
locuteur 3	-0.263346218	-0.263824377	0.319728417
locuteur 4	0.708530024	0.596984534	0.876408243
locuteur 5	0.715385047	0.732045876	0.732045876
locuteur 6	0.547722558	0.992183564	0.616509059
locuteur 7	0.716335319	0.692977479	0.630669772
locuteur 8	0.892899388	0.857436983	0.809018693
locuteur 9	0.463048825	0.500307138	0.801809802
Moyenne	0.455049696	0.465033191	0.600088697

sera approximativement la même de la précédente. Il est très important d'analyser la manière avec laquelle les experts humains évaluent la prononciation afin de s'assurer qu'une méthode d'évaluation automatique de la prononciation adoptée sera en mesure de fournir à l'apprenant des feedbacks informatifs fiable à propos de sa qualité globale de prononciation. Comme on peut le constater dans le tableau suivant, la meilleure corrélation des notes attribuées était celle de l'expert 2 (une corrélation de 0.76).

Pour les corrélations inter-experts obtenues, comme mentionné dans le tableau 3.5, il faut dire que même en la présence de différences entre la manière dont les experts humains ont jugé la prononciation, on voit que la corrélation entre les notes attribuées par les différents experts est assez similaires. Globalement, on peut dire que la fiabilité de l'évaluation des experts humains est satisfaisante vu que le nombre d'échantillons évalué n'était pas si grand, même dans le cas de l'évaluation intra-experts (jusqu'à 0.6 pour la corrélation entre l'expert 2 et l'expert 3).

Le but de cette expérimentation est que la méthode d'évaluation de la prononciation implémentée doit avoir le jugement le plus proche de celui de l'expert, et c'est pour cela

que les appréciations attribuées par la méthode d'évaluation de la prononciation qui sera présentée dans ce qui suit, doivent être corrélée avec celles des experts humains pour mettre le point sur l'importance de faire référence aux jugements des experts en prononciation pour avoir un outil d'évaluation automatique de la prononciation performant.

3.9 Conclusion

Nous avons étudié, dans le présent chapitre, l'apprentissage de la prononciation basé sur la reconnaissance automatique de la parole. Nous avons commencé par introduire le *CALL* et spécifiquement les systèmes *CAPT*. Nous avons montré, à travers l'état de l'art, qu'un système *CAPT* est une composante incontournable lorsqu'il s'agit de l'acquisition des règles correctes en ce qui concerne la prononciation. Nous avons menés aussi une étude sur les différents scores ou mesures automatiques utilisés dans le cadre de l'évaluation de la prononciation. Ces mesures ont été appliquées de différentes manières dans le but d'attribuer des scores aux segments de la parole pour pouvoir déterminer une éventuelle différence entre la parole prononcée et un modèle de référence. Certains travaux de recherches ont utilisé aussi ces mesures pour déterminer le degré de la maitrise de la parole dans le cas de l'apprentissage de langue.

Dans les chapitres suivants nous allons détailler notre contribution qui touche l'évaluation automatique de la prononciation en Arabe. Nous essayerons aussi d'expliquer comment les différentes mesures proposées dans la littérature ont été appliquées à l'évaluation automatique de la prononciation en Arabe.

Nous allons aussi dans les deux derniers chapitres de cette thèse présenter nos contributions qui visent parmi leurs objectifs de limiter les effets de variabilité dans les évaluations des experts.

Étude comparative entre les différents scores en évaluation de la prononciation

4.1 Introduction

Dans ce chapitre, nous nous intéressons à la comparaison des différents scores précédemment cités dans un contexte d'évaluation de la prononciation en Arabe et en Anglais.

En effet, pour généraliser nos résultats, nous avons choisi de comparer les scores précédents dans le cadre de deux langues différentes : Arabe et Anglais.

L'Arabe et l'Anglais sont de deux familles de langues différentes, sémitiques et germaniques, respectivement. Parce qu'ils descendent de familles de langues différentes, l'Arabe et l'Anglais ont de nombreuses différences dans leurs grammaires individuelles. La grammaire d'une langue comprend ses attributs phonétiques, et il y a beaucoup de différences phonétiques entre les langues Arabe et Anglais.

Les difficultés dans l'apprentissage de la prononciation des deux langues sont différentes. Pour l'apprentissage de l'anglais par l'algérien, l'une des principales difficultés est la maîtrise de la tension ; qui peut changer le sens d'un mot. Alors que pour l'apprentissage d'un algérien de l'Arabe standard (ceci est valable pour les apprenants venant d'autres horizons) est la maîtrise des sons qui sont propres à l'Arabe.

Notre application a été développée en utilisant la boîte à outils CMU Sphinx [18]. Pour construire un système de reconnaissance automatique de la parole deux composants sont nécessaires : modèles acoustiques et modèle de langage. Dans cette recherche, les modèles acoustiques sont construits sur la base de la représentation MFCC de phonèmes.

4.2 Le moteur de reconnaissance de la langue Arabe construit

Notre application a été développée en utilisant plusieurs versions de l'outil sphinx présenté dans le premier chapitre. Comme précédemment mentionné, pour construire un moteur de reconnaissance automatique de la parole, deux composantes principales sont nécessaires : un modèle acoustique et un modèle de langage. Dans cette thèse, les modèles acoustiques sont construits en se basant sur les MFCC pour la représentation des phonèmes.

Le corpus réservé pour l'apprentissage inclut un ensemble d'une centaine de mots conçu pour détecter les problèmes de prononciation les plus connus chez les jeunes écoliers. L'ensemble de donnée d'apprentissage comprend quinze (15) locuteurs qui ont prononcé cent(100) mots. Pour la phase de test, on considère un autre ensemble comprenant huit (8) locuteurs (différents de ceux qui ont participé dans la phase d'apprentissage). Ces huit locuteurs ont été invités à prononcer à haute voix une liste de mot en Arabe moderne standard (*MSA* : Modern Standard Arabic). Pour construire les modèles acoustiques, nous

TABLE *4.1* – Nombre de *grams* utilisés pour créer le modèle de langage Arabe

1-gram	2-gram	3-gram
2851	5694	8342

avons utilisé comme unité de base des phonèmes dépendants du contexte (Context Dependant Phoneme). Chaque phonème est représenté par trois états HMMs et des densités gaussiennes d'observation attachés à chaque état. Le CMU toolkit [21] (*Cambridge statistical language modeling*) a été utilisé pour construire le modèle de langage Arabe. Cet outil vient comme une entité supplémentaire avec le moteur de la reconnaissance automatique de la parole Sphinx. Les données textuelles utilisées pour construire ce modèle de langage consistent en un dictionnaire qui contient une liste de 100 mots. Le tableau 4.1 résume les paramètres utilisés qui ont mené à la construction d'un modèle de langage Arabe *3-gram*.

4.3 L'ensemble de données

4.3.1 La collection des échantillons wav pour l'apprentissage

Les échantillons son enregistrés durant notre projet de thèse sont un mélange de mots isolés et de la parole continue effectués par 15 jeunes Algériens. Les échantillons de mots ont été enregistrés en utilisant un microphone de haute qualité pour s'assurer de la clarté de la voix. Chaque fichier audio a été paramétré à $16kHz$ avec $16bitPCM$ par trame audio. Ce format représente le format nécessaire compatible avec l'outil de construction du modèle acoustique Sphinx.

Le corpus de texte (dictionnaire) utilisé dans cette expérimentation consiste en 100 mots en Arabe standard. L'ensemble de mots a été divisé en deux listes. La première liste contient 54 mots, et la deuxième liste contient 46 mots. Le but de l'utilisation de cette distribution est de faciliter la phase de la collection des enregistrements wav, en expliquant aux jeunes qui ont contribué à la phase de collection de données que la première liste des mots à prononcer sera une liste de mots adjectifs, tandis que la deuxième liste sera une liste de mot décrivant des noms d'endroits, noms de membre de famille, couleurs, noms d'animaux et quelques information concernant le climat.

Le nombre total d'enregistrement collecté consiste en 2748 fichiers wav. Le tableau 4.2 illustre la distribution des locuteurs, le nombre de mots lus par chaque locuteur et le nombre total des mots lus.

Pour construire les modèles acoustiques, nous avons utilisé comme unité de base des phonèmes dépendants du contexte (Context Dependant Phoneme). Chaque phonème est représenté par trois états HMMs et des densités gaussiennes d'observation attachés à

TABLE *4.2* – L'ensemble de données utilisé pour la création du modèle acoustique Arabe

	Locuteurs	Genre	Liste de 54 mots (nombre de fois lu)	Liste de 46 mots (nombre de fois lu)	Nombre total de mots lu
	*Loc.*1 → 12	3F et 9M	12	0	648
	*Loc.*02	M	2	2	200
	*Loc.*03	M	2	2	200
	*Loc.*05	M	3	3	300
	*Loc.*08	M	5	5	500
	*Loc.*09	M	2	2	200
	*Loc.*11	M	2	2	200
	*Loc.*13	M	2	2	200
	*Loc.*14	M	1	1	100
	*Loc.*15	M	2	2	200
Total locuteur par mot	15 Locuteurs	3F et 12M	33	21	2748

chaque état. Le tableau 4.3 illustre la définition des paramètres utilisés pour la création du modèle Acoustique Arabe.

TABLE *4.3* – Les paramètres du modèle acoustique utilisés

n_base	n_tri	n_state_map	n_tied_state	n_tied_ci_state	n_tied_tmat
35	2772	11228	905	105	35

où :

- n_base : représente le nombre des phonèmes utilisés pour la phase d'apprentissage du modèle acoustique,
- n_tri : est le nombre de triphones,
- n_state_map : est le nombre total des états HMMs qui représente tous les phonèmes qui forment le corpus textuel utilisé dans le dictionnaire de mots Arabes incluant les états émetteurs et les états non-émetteurs,
- n_tied_state : est le nombre total de la base de données des phonèmes juste après le partage des états (*state-sharing*),
- n_tied_tmat : est le nombre total des matrices de transition pour un ensemble de modèles donné.

Toutes ces informations sont automatiquement générées après la configuration de la base de données juste avant de commencer la phase d'apprentissage.

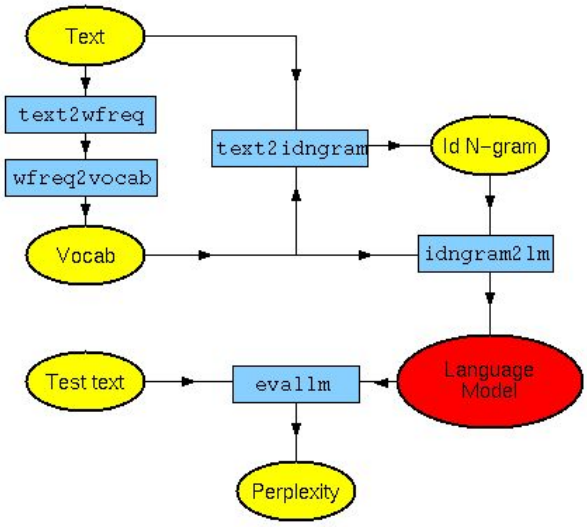

FIGURE *4.1* – L'architecture générale de l'outil de création du modèle de langage CMU-LMTK

4.3.2 Le modèle de langage

L'outil CMU-LMTK [21] qui est un outil supplémentaire dans le package sphinx a été utilisé pour la création du modèle de langage Arabe qui couvre la liste des 100 mots précédemment cités. Le processus à suivre pour créer un modèle de langage peut être résumé en 5 étapes : calculer le nombre d'unigram pour chaque mot, convertir le nombre d'unigram d'un mot en un vocabulaire, générer un identifiant *n-gram* (ID N-gram) binaire du corpus textuel d'apprentissage en se basant sur ce vocabulaire, convertir l'identifiant *n-gram* en un format binaire d'un modèle de langage et finalement calculer la perplexité du modèle de langage. La figure 4.1 illustre le déroulement du processus de la création d'un modèle de langage d'une langue donnée.

4.3.3 La collection d'échantillons de test

Pour la phase de test, on considère un autre ensemble comprenant huit (8) autre locuteurs (différents de ceux utilisés dans la phase d'apprentissage). Ces huit locuteurs ont été invités à prononcer à haute voix une liste de mot en Arabe moderne standard (MSA : modern standard Arabic). La figure 4.2 illustre un exemple de session de reconnaissance sous la plateforme Sphinx 4.

Durant une phase préliminaire de test de notre moteur de reconnaissance Arabe, le système a fourni les performances illustrées dans le tableau 4.4 sur une collection de 116 mots prononcés en Arabe.

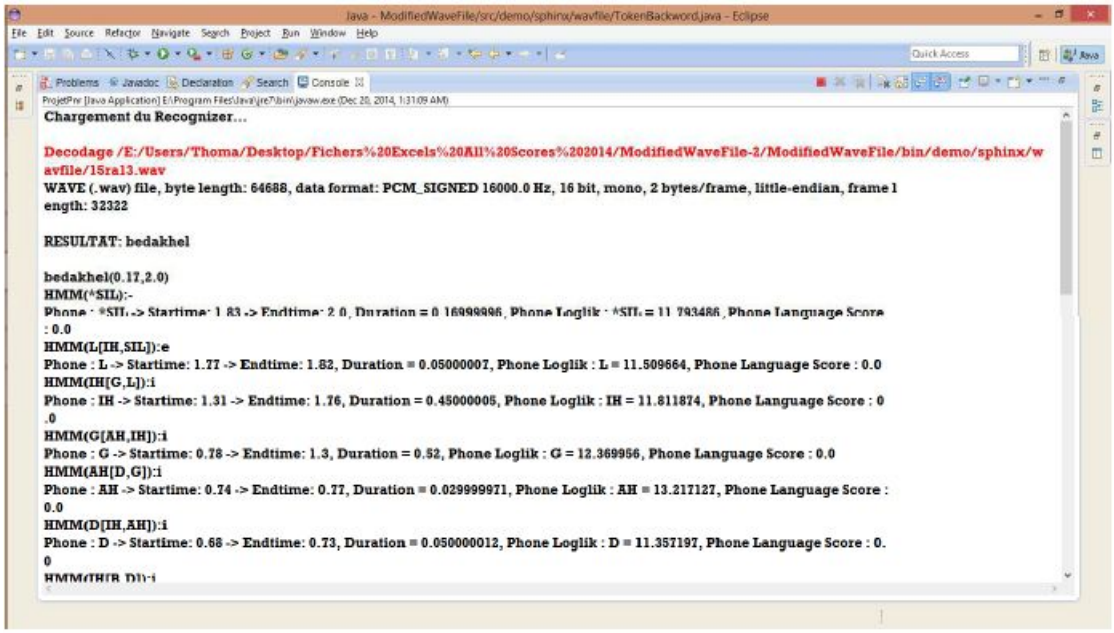

FIGURE *4.2* – Capture d'écran correspondant à une session de reconnaissance

TABLE *4.4* – Les performances de la reconnaissance de la parole

Total de mots	116	Suppressions	0
Correctes	113	Insertions	1
Pourcentage correct	97.41%	Substitutions	3
Erreur	3.45%	Word Error Rate	3.5%
Précision	96.55%	Sentence Error Rate	2.6%

4.4 Expérimentation et résultats de l'évaluation

Une fois le moteur de la reconnaissance est construit, testé et approuvé, nous avons effectué une collection d'expérimentations dans le but de signaler les scores qui sont les mieux adaptés pour la langue Arabe dans un contexte d'attribution de score à la prononciation. Pour ce but, une attention particulière a été prise pour la collection de données de test. La base de données expérimentale réservée pour le test inclut huit locuteurs. Chacun d'entre eux prononce une collection de 16 mots. Nous avons choisi attentivement des élèves avec différentes niveaux de compétence de lecture.

Le jugement des experts humains nous fournit un classement des mots prononcés par les 8 lecteurs. Cela va nous permettre de déterminer une collection de mesures pour évaluer les performances de chaque score et de fournir une interprétation plus précise des résultats.

4.4.1 Les mesures de l'évaluation

Pour évaluer les performances de la phase d'attribution des scores à la prononciation, différentes mesures ont été calculées :

1. Vrai positif (CA : Correct Acceptance) : représente les mots qui ont été bien prononcés et jugés corrects,
2. Faux positif (CR : Correct Rejection) : représente les mots qui ont été mal prononcés et jugés incorrects,
3. Vrais négatif (FA : False Acceptance) : représente les mots qui ont été mal prononcés et jugés corrects,
4. Faux négatif (FR : False Rrejection) : représente les mots qui ont été bien prononcés et jugés incorrects.
5. Précision : le score de précision (SA : Scoring Accuracy) peur être calculé en appliquant l'équation (4.1) suivante :

$$SA = (\frac{CA + CR}{CA + CR + FA + FR}) * 100 \tag{4.1}$$

Tous les scores calculés concernant l'évaluation de la prononciation sont extraits à partir des résultats obtenus durant le processus de la reconnaissance automatique de la langue Arabe.

4.4.2 Résultats obtenus pour la langue Arabe

Après la reconnaissance basée HMMs en mode alignement forcé du signal de la parole en entré, les résultats obtenus de la phase de test des 8 locuteurs sont résumés dans le tableau 4.5. Le tableau 4.5 est ordonné selon la précision obtenue pour chaque score de prononciation. La section "Les différents scores automatique utilisés en l'évaluation de la prononciation" donne plus de détails concernant comment ces scores peuvent être déterminés.

4.4.2.1 Les mesures basées sur la durée

Le tableau 4.5 suivant montre que les scores basés sur la durée sont les meilleurs scores en termes de séparation entre une bonne et une mauvaise prononciation. Le meilleur score en termes de précision est la durée des pauses TDP suivie par la durée totale de la parole TDS. Nous allons les considérer dans ce qui suit avec plus de détails (voir figure 4.3 et 4.4).

TABLE *4.5* – Les performances des scores calculés pour la langue Arabe

Score	CA	CR	FA	FR	SA
TDP	88.54%	90.63%	9.38%	11.46%	89.58%
TDS	87.50%	90.63%	9.38%	12.50%	89.06%
ROS	88.54%	81.25%	18.75%	11.46%	84.90%
GLL	87.50%	81.25%	18.75%	12.50%	84.38%
$Pauses$	89.58%	71.88%	28.13%	10.42%	80.73%
$LLL - wp$	80.21%	81.25%	18.75%	19.79%	80.73%
PTR	90.63%	65.63%	34.38%	9.38%	78.13%
$RONS$	75.00%	62.50%	37.50%	25.00%	68.75%
GOP	66.66%	68.75%	31.25%	33.33%	67.70%
ROA	88.54%	34.38%	65.63%	11.46%	61.46%
$GLL - wp$	72.92%	28.13%	71.88%	27.08%	50.52%
$TDS - wp$	21.88%	75.00%	25.00%	78.13%	48.44%
LLL	60.42%	34.38%	65.63%	39.58%	47.40%

Les figures 4.3 et 4.4 sont presque similaires. Ces deux figures montrent la différence entre les mots en se basant sur les scores TDP et TDS pour chaque locuteur. Les deux figures montrent clairement les tendances des mauvais lecteurs ayant des scores dont la valeur est un peu trop élevée. Ces sores montrent que la prononciation du locuteur 7 est la plus mauvaise tandis que celle du locuteur 1 est la meilleure. Cette évaluation est confirmée par les experts. La tendance du mauvais lecteur d'avoir la durée la plus longue est particulièrement constatée pour les mots qui sont longs. On considère les mots les plus longs et les mots les plus courts en terme de la moyenne du score TDS par rapport à tous les locuteurs. Les résultats obtenus sont illustrés dans la figure 4.5.

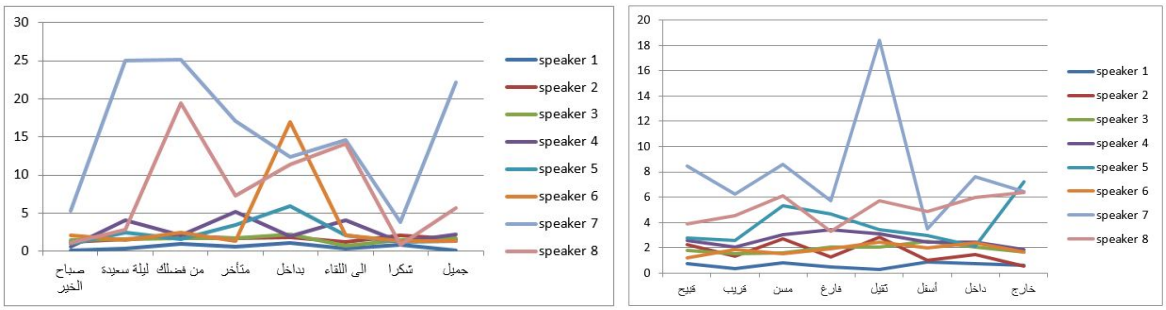

FIGURE *4.3* – TDP des mots pour chaque locuteur

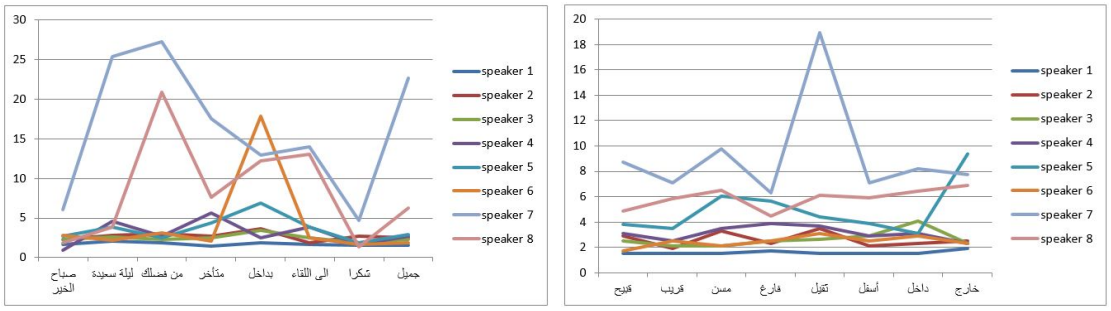

FIGURE *4.4* – *TDS* des mots pour chaque locuteur

La figure 4.5 montre que la différence entre les locuteurs est devenue plus importante lorsque le mot est long. Les locuteurs 7 et 8 semblent être les plus mauvais en termes de qualité de prononciation. Donc, de tels scores peuvent être utilisés pour évaluer une longue séquence de phonèmes.

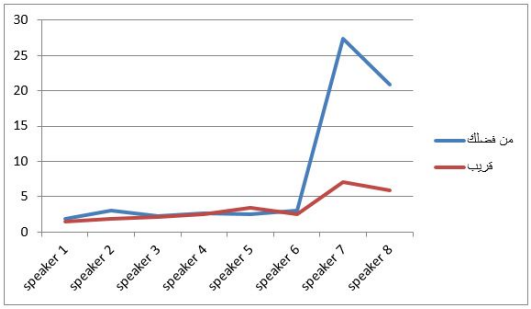

FIGURE *4.5* – Le score *TDS* obtenu pour les mots courts et longs

Même si ces deux mesures, très proches par définition, donnent de meilleurs résultats, une observation plus attentive montre qu'ils donnent plus d'informations à propos des compétences de lecture des apprenants que sur la prononciation d'un mot (ou d'une phrase). Si on considère le mot /بِدآخِل/ qui n'est pas très commun, et en se référant à la figure 4.4, le graphe montre que le score *TDS* de plusieurs locuteurs est élevé. On note aussi que parmi eux il y a de bon locuteurs (comme le locuteur 5) qui a fait des pauses avant de commencer la lecture. Ainsi, pendant que le score *TDP* semble être le meilleur score qui permet la distinction entre ce qui a été bien prononcé et ce que ne l'été pas, le score *TDS* reste le mieux placé vu qu'il ne néglige ni la parole ni les pauses ou les phonèmes silencieux.

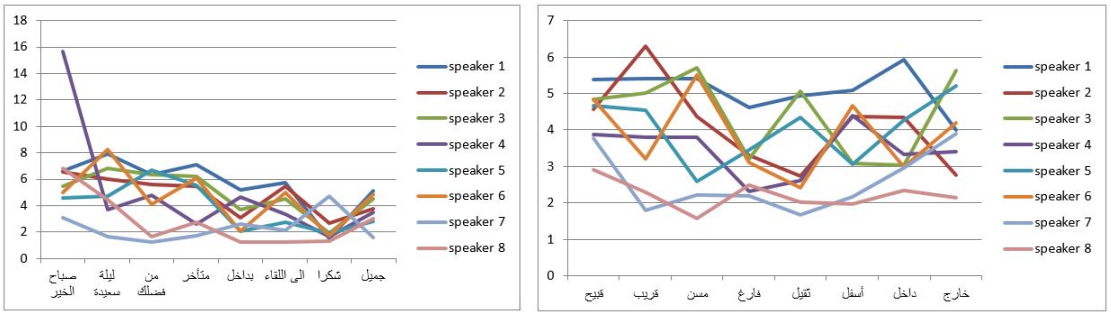

FIGURE *4.6* – *GLL* des mots pour chaque locuteur

D'un autre côté, le score PTR détient la meilleure précision quant aux faux négatifs (la valeur la plus basse par rapport aux autres scores). Cela est particulièrement souhaitable dans le cas des jeunes écoliers qui ont besoin d'être encouragés dans leurs processus d'apprentissage de la prononciation.

4.4.2.2 Les mesures basées sur le logarithme de vraisemblance

Nous considérons maintenant les meilleurs scores basés sur la vraisemblance. Le logarithme de vraisemblance nous informe d'une distance approximative de la prononciation en entrée par rapport à son modèle de référence. Selon le tableau 4.5, le score : moyenne globale du logarithme de vraisemblance (GLL) est le meilleur par rapport aux autres qui appartiennent à cette famille de scores.

Le score GLL permet l'évaluation de la prononciation des mots en fournissant un ratio de vrai positif compétitif. Il permet aussi une évaluation fiable concernant les performances globales du locuteur. En effet, le locuteur 5 (figure 4.6) avait de bonnes performances, il avait aussi une bonne articulation selon l'avis des experts, même si il a commis quelques erreurs de prononciation au niveau du mot /اَلَى اَلّقـآء/ et /بِـدآخِـل/. Ces erreurs de prononciation sont très lisibles sur la figure 4.6.

D'un autre côté, le score GOP n'a pas donné de bonnes performances. Cela est sûrement dû à la phase de la reconnaissance en mode libre qui requiert une modélisation plus précise des phonèmes.

TABLE *4.6* – Les performances des scores calculés pour la langue Anglais

Score	CA	CR	FA	FR	SA
TDP	83.33%	87.50%	12.50%	16.67%	85.42%
$RONS$	83.33%	79.17%	20.83%	16.67%	81.25%
ROS	83.33%	70.83%	29.17%	16.67%	77.08%
TDS	50.00%	100%	0%	50.00%	75.00%
GOP	83.33%	62.50%	37.50%	16.67%	72.92%
GLL	100%	45.83%	54.17%	0%	72.92%
$LLL - wp$	100%	45.83%	54.17%	0%	72.92%
LLL	100%	41.67%	58.33%	0%	70.83%
PTR	83.33%	54.17%	45.83%	16.67%	68.75%
ROA	100%	25%	75%	0%	62.50%
$GLL - wp$	50%	62.50%	37.50%	50%	56.25%
$TDS - wp$	50.00%	58.33%	41.67%	50%	54.17%
$Pauses$	100%	0%	100%	0%	50%

4.4.3 Résultats obtenus pour l'Anglais

Pour pouvoir apporter une appréciation des scores testés dans le contexte de l'Arabe, nous avons souhaité testé ces mêmes scores dans le cadre de l'Anglais. Pour cela, nous avons utilisé les modèles acoustiques des phonèmes qui sont offerts par Sphinx. Pour mener les expérimentations en évaluation de la prononciation pour l'Anglais, nous avons collecté les enregistrements de dix (10) locuteurs Algériens qui prononcèrent cinq (5) mots en Anglais. Le tableau 4.6 résume les résultats de tous les scores obtenus pour ces prononciations.

Pour l'Anglais, il apparaît clairement que le GOP donne de bons résultats. Toutefois, En ce qui est des scores relatifs à la durée, nous remarquons (figure 4.7) que le TDS se maintient toujours en bonne position pour montrer les « mauvais » locuteurs.

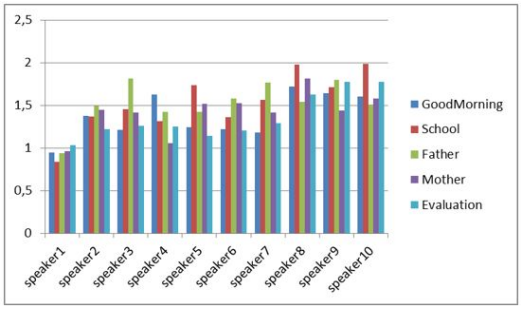

FIGURE *4.7* – TDS des mots prononcés en Anglais pour chaque locuteur

De même, la tendance est toujours accentuée pour les mots les plus longs tels que « Evaluation » dans ce cas. Nous allons considérer maintenant les meilleurs scores basés sur le logarithme de vraisemblance (figure 4.8). Selon le tableau 4.6, le score moyenne globale

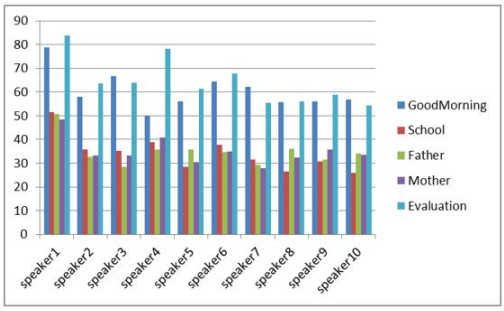

FIGURE *4.8* – *GLL* des mots prononcés en Anglais pour chaque locuteur

du logarithme de vraisemblance GLL est le meilleur score. Même si le score GOP donne de meilleurs résultats, le score GLL reste compétitif car le ratio de fausse alarme est à zéro.

4.5 Conclusion

Dans ce chapitre, nous avons présenté nos expérimentations dont le but était de construire un système basé sur la technologie de la reconnaissance automatique de la parole qui peut évaluer la prononciation des jeunes écoliers Arabe en appliquant les techniques $CAPT$.

Deux groupes de scores automatiquement générés ont été calculés pour évaluer les compétences de lecture des apprenants. Nous avons retenu parmi eux deux scores. Premièrement le TDS semble être une bonne mesure d'évaluation de différentes performances des apprenants. En effet, l'apprenant qui a des difficultés de lecture a tendance de produire plusieurs pauses durant la prononciation en la rendant trop longue par rapport à la norme standard.

Le deuxième score est le GLL qui permet de bien distinguer entre la bonne et la mauvaise prononciation d'une séquence de phonèmes (mot ou phrase) en terme de vrai positif. Il peut être aussi une bonne mesure pour évaluer les différentes performances d'un lecteur sur une collection de mots. Ces deux scores peuvent fournir des *feedbacks* informatifs fiables pour les jeunes écoliers débutant l'apprentissage de la prononciation de la langue Arabe.

Approche statistique de décision pour l'évaluation de la prononciation en Arabe

5.1 Introduction

A l'heure actuelle, peu nombreux sont les travaux qui impliquent la langue Arabe comme une langue cible dans le but d'enseigner la prononciation correcte. Dans [85] les auteurs ont présentés le système *VAT* (Versant Arabic Test). Il s'agit d'un test complètement automatique pour attribuer des scores à la prononciation de la langue Arabe standard moderne (MSA ; Modern Standard Arabic). Le but de ce système est de faciliter le couplet écouter/parler. Le test proposé fournit un calcul de quatre scores selon plusieurs dimensions concernant la maîtrise des phrases, le vocabulaire, l'articulation et les scores de prononciation. Ces scores sont obtenus en utilisant un moteur de reconnaissance automatique de la parole basé sur les HMMs.

L'objectif du travail présenté dans [42] est d'aider les jeunes écoliers Algériens, qui ont des difficultés de prononciation de la langue Arabe, à améliorer leur compétence de prononciation à travers l'utilisation de la technologie de la reconnaissance automatique de la parole. Le système proposé est basé sur des méthodes de détection des erreurs de prononciation dont différents scores, basés sur la durée et la vraisemblance, ont été calculés pour pouvoir mesurer quantitativement la mauvaise prononciation du mot. À travers les résultats, les auteurs ont conclu que le score GLL surpasse les autres scores proposés.

Le but de la contribution que nous allons présenter dans le présent chapitre est la détection des difficultés de lecture chez des jeunes écoliers Algériens. Nous nous focalisons particulièrement sur la décision à propos de la prononciation effectuée si elle est bonne ou mauvaise. Pour cela, on considère la réponse à cette question comme étant un problème de classification et une approche statistique de décision sera proposée sous forme d'un module de décision. Cette approche va nous permettre de poursuivre l'investigation en ce qui concerne la prononciation de chaque phonème dans un mot ou dans une phrase.

5.2 L'architecture du système proposé

Dans un système CAPT, le texte à prononcer est présenté à l'apprenant sous forme audio, texte écrit, phrase,...etc. Donc le système « sait » ce que doit prononcer l'apprenant. Pour cela, la phase de la reconnaissance est simplifiée dès l'analyse du signal de parole en entrée. La représentation acoustique obtenue est ensuite alignée aux modèles de référence du texte affiché. L'alignement effectué est un alignement en mode forcé. L'étape de décodage HMM en mode alignement forcé fournit un score qui mesure la distance entre le signal de la parole en entré et les modèles de références. La sortie ou le résultat est un ensemble de deux scores : la durée du phonème ainsi que son logarithme de vraisemblance. Ensuite, d'autres scores seront déterminés (dérivés) à partir de ces deux derniers scores.

Pour la suite de notre proposition, on assume que la séquence des phonèmes d'un mot

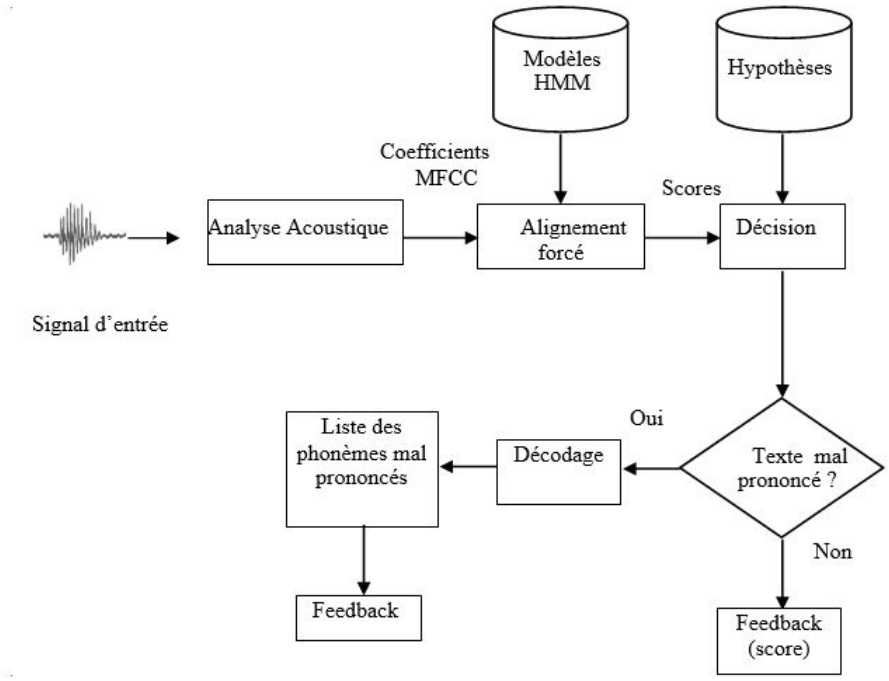

FIGURE *5.1* – L'architecture du système proposé, d'après [86]

donné obtenue suit une loi de distribution normal. Une fois les scores sont déterminés, une décision basée sur des statistiques sera effectuée. On suppose qu'une hypothèse H_0 est l'hypothèse qui signifie que le mot a été bien prononcé ; et qu'une hypothèse H_1 est son alternative i.e. le mot a été mal prononcé. Les scores déterminés sont comparés à une valeur lue à partir de la table de *Student*. Cette valeur représente un seuil qui, par rapport à sa position dans le tableau, l'hypothèse H_0 sera acceptée ou rejetée.

5.2.1 L'architecture du système

La figure 5.1 illustre l'architecture générale du système proposée. Le moteur de la reconnaissance automatique de la parole nous fournit deux scores de bases : l'un est basé sur la vraisemblance et l'autre sur la durée des segments. Nous parlons ici du score logarithme de vraisemblance et la durée de chaque phonème extrait.

5.2.2 Les scores calculés

Dans ce qui suit, nous allons présenter les scores calculés et générés par le système de l'évaluation de la prononciation proposé.

5.2.2.1 Le logarithme de vraisemblance

La section 3.4.2.1 donne plus de détails concernant la détermination du score logarithme de vraisemblance. Avant de commencer le processus de l'évaluation automatique

de la prononciation, une normalisation du logarithme de vraisemblance est effectuée qui consiste en l'utilisation d'une fonction sigmoïdale qui peut être calculée par l'équation (5.1) suivante :

$$sigmoid(LL_i) = \frac{\alpha}{e^{-\beta LL_{p_i}}} \tag{5.1}$$

où LL_i représente le score du logarithme de vraisemblance du $i^{ème}$ phonème, et α et β sont deux paramètres empiriquement déterminés. Le but de l'utilisation de cette fonction sigmoïdale est de réduire l'intervalle du logarithme de vraisemblance vu qu'il est extrait après le décodage HMM en valeur négative et très grande. Dans ce qui suit, quand le symbole LL_i se présente on suppose qu'il est déjà sous sa forme sigmoïdale.

5.2.2.2 La durée des phonèmes

Le score qui représente la durée du phonème Di est déterminé en appliquant l'algorithme de *Viterbi* implémenté dans le décodeur HMM [42]. Pour obtenir la durée D d'un phonème i, on soustrait l'instant du début t_i de l'instant de fin t_{i+1} (ou l'instant du début du phonème suivant) du phonème prononcé afin d'obtenir la durée exacte en utilisant l'équation (5.2) correspondante :

$$D_i = t_{i+1} - t_i \tag{5.2}$$

5.2.2.3 Le score DNLL (Duration Normalized Log Likelihood)

Le score $DNLL$ d'un phonème i (P_i) est obtenu en divisant le score logarithme de vraisemblance du phonème correspondant par sa durée en utilisant l'équation (5.3) suivante :

$$DNLL(P_i) = \frac{LL_i}{D_i} \tag{5.3}$$

5.3 Proposition pour l'évaluation de la prononciation en Arabe

Les modèles des mots utilisés ont été construits en utilisant les HMMs. Il est supposé que le système proposé se base sur le calcul des probabilités de vraisemblance pour détecter les erreurs de prononciation au niveau mot. En se basant sur cette information ainsi que les modèles associés un feedback correctif est renvoyé à l'apprenant.

Le système de la reconnaissance de la parole produit un score qui indique de combien la prononciation en entrée était proche par rapport à un modèle de référence.

L'étape suivante est de vérifier si la prononciation en entrée est considérée comme correcte en se basant sur ce score. Souvent, un seuil est prédéfinit. Si le score déterminé est plus grand que le seuil prédéfinit, alors la prononciation sera acceptée comme étant correcte (et vice-versa). Au lieu de définir ce seuil empiriquement après plusieurs tentatives ; nous proposons de définir ce seuil en utilisant un test statistique, en l'occurrence : le test de *Student*.

5.3.1 Le test de *Student*

Le test de *Student* (ou le *t-test*) vérifie quand est ce que la moyenne de deux groupes sont statistiquement différents. Le test de *Student* représente une bonne solution pour les problèmes associés à une inférence basé sur des échantillons non trop volumineux. Il permet d'affirmer si deux ensembles quelconques sont vraiment différents même s'ils contiennent des redondances.

Dans un premier lieu, on a besoin de construire une hypothèse nulle H_0 sur laquelle les expérimentations vont être effectuées. Quand on teste une hypothèse nulle ; qui signifie que la moyenne de l'échantillon est égale à une valeur spécifiée μ_0, on utilise la statistique suivante :

$$t = \frac{\overline{X} - \mu_0}{\frac{\sigma}{\sqrt{n}}} \tag{5.4}$$

où $\overline{X}$ est la moyenne de l'échantillon, σ est l'écart-type de l'échantillon tandis que n est la taille de l'échantillon. Le degré de liberté utilisé dans ce teste est $n - 1$.

Une fois le score t est calculé, une valeur p peut être déterminée à partir de la *table de distribution de Student*. Si la valeur de t dépasse la valeur tabulée cela veut dire que la moyenne est différente par rapport au niveau de signification p choisi (d'habitude cette valeur peut prendre 0.01, 0.05 ou 0.10) ; ainsi l'hypothèse nulle H_0 sera rejetée en faveur de l'hypothèse alternative H_1.

5.3.2 Le test de *Student* appliqué à l'évaluation de la prononciation

Comme précédemment mentionné, on assume que les scores de la séquence des phonèmes obtenus suivent une loi de distribution normale, tant dit que la décision d'accepter ou rejeter une nouvelle prononciation sera basée sur le test de *Student*.

Pour chaque phonème Arabe, un modèle HMM est construit. Les mots préparés pour être prononcés par les apprenants et évalués par le système sont considérés comme une séquence de phonèmes connue. Nos expérimentations avaient pour but de vérifier si le signal

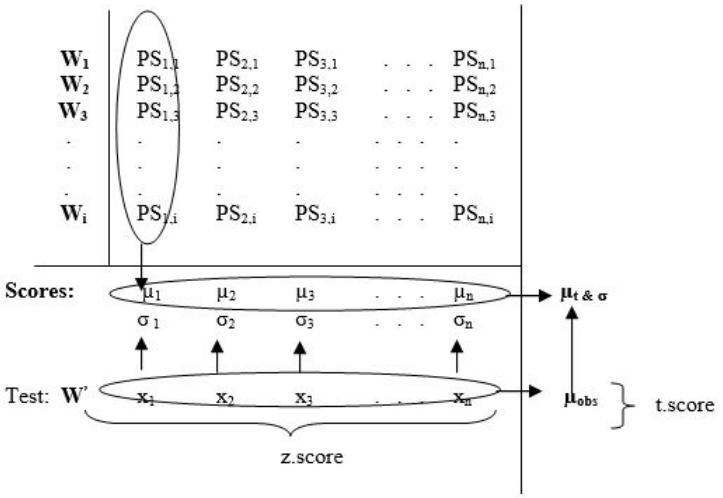

FIGURE *5.2* – Illustration de *t.score* et *z.score*, d'après [86]

de la parole en entrée est significativement diffèrent par rapport aux bonnes prononciations d'un mot donné (i.e. par rapport aux modèles construits). À partir des réalisations de ces mots (les prononciations) la moyenne des scores de la séquence des phonèmes extraite sera déterminée. Cette moyenne représente la moyenne théorique μ_t. concernant la prononciation en entrée qui va être analysé par le système de l'évaluation proposé ; une moyenne observée μ_{obs} des scores de la séquence de phonèmes correspondante sera déterminée.

Ainsi, l'hypothèse $H_0(\mu_{obs} = \mu_t)$ signifie que la prononciation à évaluer n'est pas significativement différente en comparant avec le modèle de référence. L'hypothèse H_1 confirme la présence d'une différence ; cela veut dire que la prononciation est significativement différente par rapport au modèle de référence.

Considérons la figure 5.2, où les assomptions suivantes peuvent en être extraites : pour un mot donné W, il peut y avoir i réalisations ou essais de prononciation. Chacun de ces essais est analysé selon les modèles des références des phonèmes. Pour cela, chacun de ces mots sera représenté par une séquence de scores comme suit :

$$W_i = PS_{i,1}, PS_{i,2}, ..., PS_{i,n} \tag{5.5}$$

où $PS_{i,n}$ est le score du $n^{ème}$ phonème appartenant à la $i^{ème}$ réalisation du mot W. Au niveau mot, la moyenne et l'écart-type de chaque score de phonème dans la séquence est calculé. μ_j est la moyenne du phonème j sur toutes réalisations d'un mot W. σ_j est l'écart-type associé. Les ensembles de μ_j et σ_j représentent le modèle de référence du mot. La moyenne et l'écart-type sont calculés en considérant chacun des trois scores : la durée, le logarithme de vraisemblance ainsi que le logarithme de vraisemblance normalisé $DNLL$.

Pour une prononciation à évaluer (W') ; W' est alignée en mode forcé à la séquence de phonèmes et une séquence de scores est récupérée. Selon le $t-test$ (ou le test de *Student*), deux autres scores de prononciation sont calculés : le $z.score$ et le $t.score$. Pour une évaluation au niveau phonème, le $z.score$ est déterminé pour chaque catégorie de phonème appartenant au mot. Quant au niveau mot, l'évaluation de la prononciation est automatiquement effectuée en calculant le $t.score$ en utilisant l'équation (5.6) suivante :

$$t.score(W_i) = \frac{|\mu_{obs} - \mu_t|}{\sqrt{\frac{\widehat{\sigma^2}}{N}}} \tag{5.6}$$

où le score μ_{obs} est la moyenne observée de la séquence de phonèmes, les scores μ_t et $\widehat{\sigma^2}$ sont ; respectivement, la moyenne théorique et l'écart-type de l'échantillon estimé de la séquence de phonèmes déterminée à partir des références du mot, N est le nombre de phonèmes.

Au niveau phonème, l'évaluation de la prononciation est effectuée en calculant le score $z.score$ donné par la formule (5.7) suivante :

$$z.score(P_j) = \frac{|X_j - \mu_j|}{\sigma_j} \tag{5.7}$$

Où P_j est le $j^{ème}$ phonème au niveau du mot à évaluer, X_j : le score obtenu du phonème correspondant, μ_j et σ_j sont respectivement la moyenne et l'écart-type de la séquence de phonèmes déterminée à partir des mots de référence.

Pour accepter ou rejeter une prononciation, au niveau mot, comme étant correcte ou fausse, le $t.score(W')$ est comparé à une valeur p extraite depuis la table la de loi de *Student* (ou la table $t-test$). Au niveau phonème, le $z.score$ est comparé à une valeur p extraite depuis la *table de la loi normale*.

5.3.3 Le niveau de signification d'une prononciation

Un niveau de signification, lorsqu'on a besoin 'évaluer une prononciation, représente un poids qui fait référence à une région d'acceptation ou une région de rejet. Les niveaux de signification de la prononciation sont les limites supérieures et inférieures fixées. Ils sont utilisés pour déterminer la proportion des phonèmes appartenant à chaque mot qui va se trouver dans les marges des limites imposées. Ces limites représentent des seuils qui sont lus à partir de la table des probabilités de distribution normale pour l'évaluation au niveau phonémique, et de la table des probabilités de distribution *Student* pour l'évaluation au niveau mot.

La région d'acceptation de la prononciation est la région qui correspond à une parole

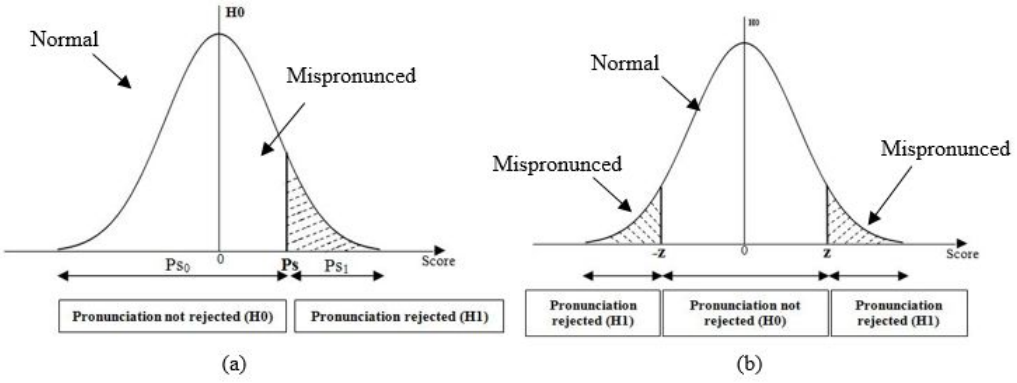

FIGURE *5.3* – La région d'acceptation/rejet de la prononciation, d'après [86]

normale dont aucune erreur de prononciation n'est signalée et vice-versa.

5.3.3.1 Test de prononciation unilatéral/bilatéral

Un test de prononciation unilatéral place (pondère) la valeur complète du niveau de signification de la prononciation sur soit la limite supérieure soit la limite inférieure comme illustré dans la figure 5.3(a). La région hachurée représente la région qui ne correspond pas au niveau de signification de la prononciation. Un test de prononciation bilatéral place une moitié du niveau de signification de la prononciation sur la limite supérieure. L'autre moitié est placée sur la limite inférieure comme montré dans la figure 5.3(b). Les deux régions hachurées représentent les régions qui ne correspondent pas au niveau de signification de la prononciation. Dans cette contribution, on considère seulement le test bilatéral.

5.3.4 Un exemple illustratif

Nous allons maintenant étudier une prononciation d'un mot à évaluer avec un degré de liberté égale à 7 (le nombre de phonèmes de ce mot étant 8). Nous supposons que le $t.score$ calculé au niveau mot avait la valeur 0.251. Si on considère un niveau de signification de prononciation égal à 0.2 ($p = 1 - 0.2$) ; la valeur de p lu à partir de la table de *Student* est 0.263. Cela veut dire que la prononciation du mot sera acceptée vu que le $t.score$ obtenu n'a pas dépassé la limite imposé ($0.251 < 0.263$). Si le niveau de la signification de la prononciation était 0.1 la prononciation du mot sera rejetée car la valeur de p serait 0.130 (voir tableau 5.1).

Prenant maintenant un cas où l'évaluation de la prononciation au niveau phonémique est nécessaire. Si on considère que le $z.score$ du premier élément de la séquence phonétique est $z.score(P_1) = 0.31$, la valeur de p correspondante, lu automatiquement à partir de la table de la loi normale, sera 0.6217. Cette valeur sera ensuite comparée au niveau

TABLE *5.1* – La table de *Student*

P / v	0.90	0.80	0.70	0.60
1	0.158	0.325	0.510	0.727
2	0.142	0.289	0.445	0.617
3	0.137	0.277	0.424	0.584
4	0.134	0.271	0.414	0.569
5	0.132	0.267	0.408	0.559
6	0.131	0.265	0.404	0.553
7	0.130	0.263	0.402	0.549
8	0.130	0.262	0.399	0.546

TABLE *5.2* – La table de la loi normale

z	0.00	0.01	0.02
0.0	0.5000	0.5040	0.5080
0.1	0.5398	0.5438	0.5478
0.2	0.5793	0.5832	0.5871
0.3	0.6179	0.6217	0.6255
0.4	0.6554	0.6591	0.6628
0.5	0.6915	0.6950	0.6985

de signification de la prononciation choisit. Si la valeur de p est inférieure ou égal au niveau de signification alors la prononciation du phonème était correcte, autrement elle sera automatiquement rejetée (voir tableau 5.2).

5.4 Expérimentation et résultats

Pour tester la méthode de l'évaluation de la prononciation en Arabe proposée, une base de données d'enregistrement wav a été préparée. Les enregistrements wav ont été effectués par des locuteurs qui ont une bonne maitrise de prononciation et d'autres qui ont des difficultés de prononciation. Les locuteurs sont invités à prononcer 16 mots en Arabe, distribués sur 80 essais où 48 parmi ces essais sont bien prononcés tandis que les autres 32 essais contiennent des erreurs de prononciation.

Comme précédemment mentionné, trois scores ont été utilisés afin de décider si la prononciation est correcte ou mal réalisée (acceptée ou rejetée). Les 16 mots prononcés par les locuteurs ont été regroupés selon le nombre de phonèmes dont le but est d'avoir leur degré de liberté (DLL) correspondant. Le tableau 5.3 suivant illustre la distribution des mots ainsi que leur degré de liberté.

TABLE *5.3* – Nombre de mots regroupés par leur degré de liberté

DLL	12	8	7	6	5	4
Mots	1	2	1	2	1	9

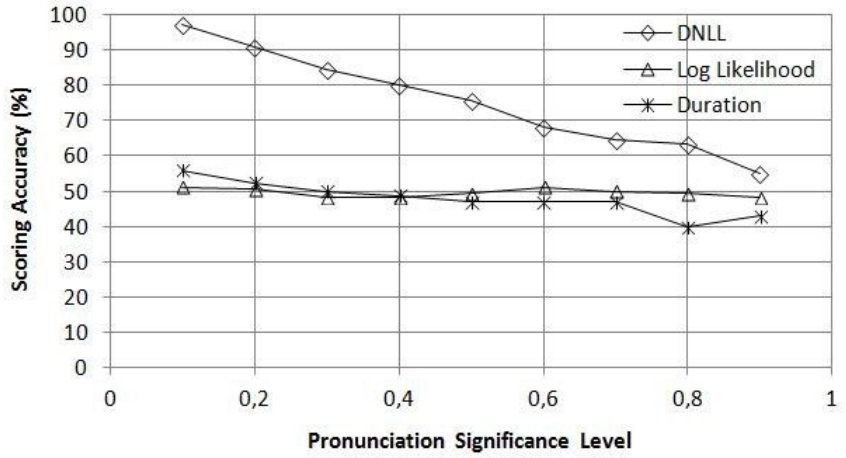

FIGURE *5.4* – Le score de précision *Vs.* Niveaux de signification de prononciation pour $DNLL$, logarithme de vraisemblance et la durée, d'après [86]

Pour évaluer les performances de l'approche proposée, nous allons considérer quatre des mesures déjà présentés (voir section 4.4.1), à savoir :

1. Vrai positif (CA : Correct Acceptance)
2. Faux positif (CR : Correct Rejection)
3. Vrais négatif (FA : False Acceptance)
4. Faux négatif (FR : False Rejection)

Pour obtenir de meilleure performance, l'approche proposée doit détecter des erreurs de prononciation et en même temps elle ne doit pas rejeter les mots qui ont été bien prononcés.

Comme une étape de décodage phonétique est éventuellement déclenchée dans le cas où le la prononciation au niveau mot est erronée, la mesure CA des mots ne doit pas être significativement différente de celle calculée au niveau phonétique concernant les locuteurs qui ont une bonne maitrise de prononciation. La mesure CR calculée au niveau mot doit être significativement différente de la mesure CA calculée au niveau phonétique concernant les locuteurs qui ont des difficultés de prononciation.

Pour évaluer la performance de l'algorithme de l'évaluation de la prononciation proposé en terme de détection d'erreurs de prononciation une mesure de précision (SA ; Scoring Accuracy) a été calculé par la formule (4.1) précédemment donnée.

La performance a été évaluée pour les trois scores de prononciation : la durée, le logarithme de vraisemblance et le $DNLL$, dans le but de déterminer lequel entre eux donne le meilleur résultat. La figure 5.4 illustre les résultats trouvés en terme de précision SA des mots sous différents niveaux de signification de prononciation.

TABLE 5.4 – Distribution des mots en CA, CR, FA, FR et les résultats obtenus pour les scores $DNLL$, logarithme de vraisemblance (LL) et la durée

	$DNLL$	LL	Durée
CA	91.66%	58.33%	66.66%
CR	96.87%	31.25%	31.25%
FA	3.12%	56.25%	56.25%
FR	2.08%	29.16%	20.83%
SA	97.31%	51.19%	55.95%

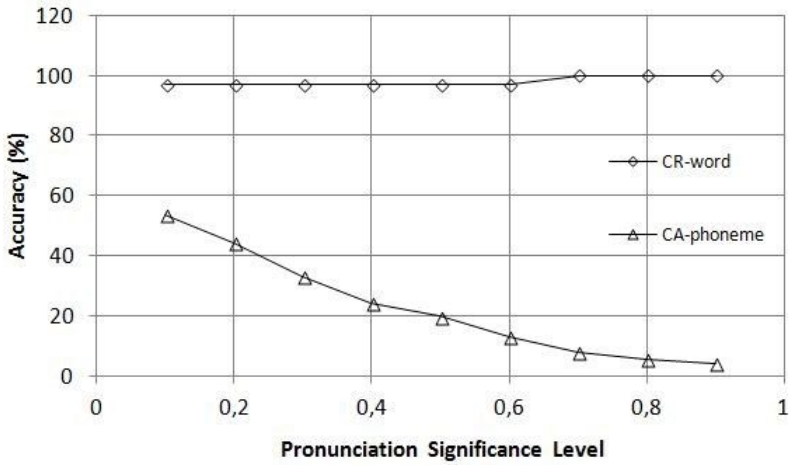

FIGURE 5.5 – CR des mots *Vs.* CA des phonèmes correspondants sous différents niveau de signification de prononciation

Comme le montre la figure 5.4, on peut en déduire que le meilleur système utilise le logarithme de vraisemblance normalisé par la durée $DNLL$ comme une mesure d'évaluation de prononciation. À un niveau de signification de prononciation de 0.1 une précision de 97.31% est observée. Pour cela, on fixe le niveau de signification de prononciation à une valeur de 0.1 pour séparer la prononciation correcte du mot de celle incorrecte et ceci est appliqué aussi quant au décodage phonétique.

Le score précision obtenu pour les autres scores de décision (durée et logarithme de vraisemblance) est synthétisé dans le tableau 5.4.

Comme précédemment mentionné, si le mot est jugé mal prononcé un décodage à niveau phonétique est nécessaire pour vérifier si la réalisation des phonèmes est proche des modèles de référence ou non. D'autre part, la figure 5.5 montre les résultats obtenus par l'application de l'approche proposée pour les locuteurs qui ont des difficultés de prononciation.

D'après la figure 5.5, il est clair que pour chaque niveau de signification de prononciation choisi, la différence entre la mesure CR des mots et la mesure CA des phonèmes correspondants est statistiquement significative ; chose qui prouve la validité de l'approche

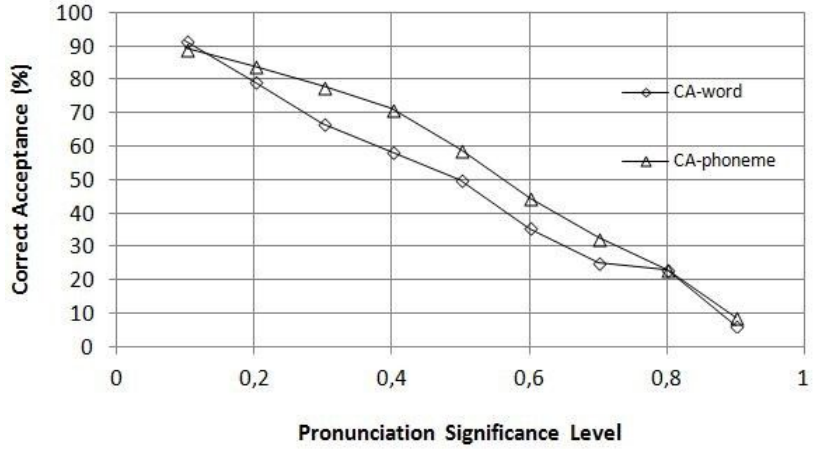

FIGURE 5.6 – CA des mots *Vs.* CA des phonèmes correspondants sous différents niveau de signification de prononciation

proposée. Pour garder une meilleure précision, un niveau de signification de prononciation de 0.1 montre que 96.87% de CR des mots a produit un taux de 53.36% de CA des phonèmes sur un total de 104 phonèmes.

Pour les mots qui ont été jugés bien prononcés, les résultats obtenus ont montré que le pourcentage des phonèmes correctement prononcés, après l'application de la phase de décodage phonétique, est corrélé au pourcentage des mots correctement acceptés comme le montre dans la figure 5.6.

On peut déduire, à partir de la figure 5.6, que la différence entre la mesure CA des mots et celle des phonèmes correspondants n'est pas grande. Pour un niveau de signification de prononciation de 0.1 un taux de 89.42% de phonèmes correctement acceptés ainsi qu'un taux de 91.66% des mots correctement acceptés ont été observés et qui sont très proches l'un à l'autre.

5.5 Conclusion

Dans ce chapitre nous avons présenté notre contribution. Il s'agit d'une approche statistique de décision pour l'évaluation automatique de la prononciation en Arabe. Le but de cette contribution est d'assister et d'aider les jeunes écoliers Algériens pendant le processus d'apprentissage de la lecture en Arabe. Cela est effectué en fournissant une évaluation précise de leur prononciation. On doit noter dans une telle situation que l'enfant ne doit pas être découragé le long du processus d'apprentissage mais il est aussi nécessaire de signaler ses erreurs de prononciation.

Cela fait de la décision concernant la prononciation de l'enfant une tache très difficile et compliquée. De plus, peu de données standardisées sont disponibles pour la langue Arabe et il faut dire qu'elles ne sont pas appliquées dans un contexte d'apprentissage

de la prononciation pour les jeunes écoliers. Ces conditions rendent la définition d'un seuil empirique très difficile pour accepter ou rejeter une prononciation. Pour pallier à ce problème, l'approche proposée dans ce chapitre pourra être de grande utilité. De plus, cette approche pourra être aussi adaptée pour prendre en charge toutes les autres langues, particulièrement celles où peu de ressources sont disponibles.

D'un autre point de vue, le travail réalisé dans cette contribution nous a permis de construire une base de données d'enregistrement wav qui va nous aider de poursuivre nos études dans ce domaine ainsi que ceux des autres chercheurs travaillant sur la langue Arabe. D'un autre côté, les résultats obtenus montrent une corrélation entre la théorie et les expérimentations. Cela encourage l'idée de combiner un outil de décision avec les HMMs pour avoir des résultats satisfaisants.

Les résultats obtenus dans des situations réels nous encouragent à poursuivre l'étude de cette approche.

Approche floue pour l'évaluation automatique de la prononciation en Arabe

6.1 Introduction : Vers une combinaison floue pour l'évaluation de la prononciation

Nous avons proposé dans le précédent chapitre une méthode de décision qui nous permet de poser la valeur d'un seuil en s'aidant d'un test statistique. Cette méthode est particulièrement adaptée dans le cas où le nombre d'échantillons représentatif d'une forme n'est pas très volumineux. Et de ce fait, peut tout à fait être adapté aux langues peu dotées de ressources. Dans la même optique, nous allons poursuivre nos propositions par une seconde alternative pour les langues peu dotées de ressources et en particulier, la langue Arabe.

D'abord, nous allons reprendre nos conclusions de l'étude comparative entre les scores machines dédiés à l'évaluation de la prononciation en Arabe, nous y avons conclu que le GLL et le TDS étaient les deux scores qui ressortent, nous allons alors les combiner pour renforcer leur pertinence. Ensuite, Rappelons-nous nos investigations quant à la corrélation ou non des notes des experts dans le contexte de la langue Arabe. Une fois ce constat posé, nous proposons une solution qui consiste en l'utilisation des ensemble flous pour poser une appréciation et ce pour être aussi proche que possible des différentes notes des experts.

Pour cela, un système d'évaluation floue de la prononciation sera présenté. Les expérimentations seront effectuées sur la prononciation en Arabe et en Anglais dans un but de généralisation de la proposition.

6.2 Combinaison floue des scores de prononciation

À l'issu du quatrième chapitre, nous avons conclu que deux scores ; GLL et TDS, peuvent être retenus comme représentatifs d'une évaluation pour une prononciation en Arabe. Nous souhaitons dans ce chapitre, réaliser une combinaison de ces deux scores. Pour cela plusieurs méthodes existent telles que par le biais des arbres de décision, des réseaux de neurones ou via une fonction auto-regressive comme cela a déjà été réalisé dans d'autres travaux dans le cadre de l'évaluation de la prononciation. Nous avons choisi dans cette partie de réaliser cette combinaison via *des règles expertes floues*. Le choix de la logique floue est motivé par notre souci d'avoir recours aux ensembles flous pour définir nos seuils qui permettent de séparer entre une bonne et une moins bonne prononciation.

Une méthode classique de décision consiste en le rejet automatique de la prononciation quand le score attribué est en dessous d'un seuil prédéfinit (figure 6.1). Mais quand une approche logique floue est appliquée pour vérifier si la prononciation est correcte ou non (figure 6.2), le système réagit avec une grande souplesse au niveau des frontière qui

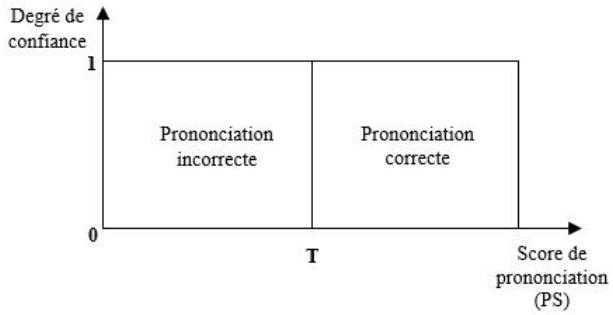

FIGURE *6.1* – Méthode classique d'évaluation de la prononciation où T représente un seuil prédéfinit

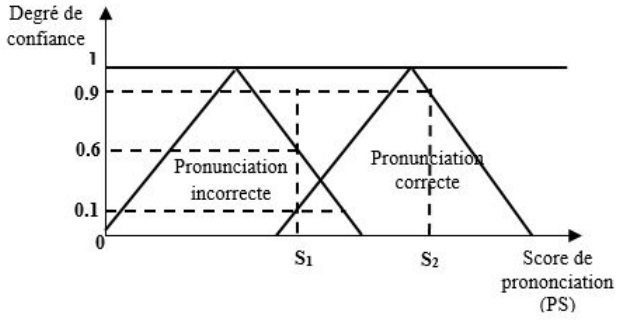

FIGURE *6.2* – Méthode d'évaluation de la prononciation basée sur la logique floue

séparent l'une de l'autre. On parle de fonction d'appartenance, en effet, une prononciation peut être à la fois bonne et mauvaise avec différents degrés d'appartenance.

La figure 6.1 illustre le cas où un seuil est appliqué pour vérifier le degré de déviation entre une prononciation correcte et une prononciation mal réalisée. Concrètement, cela se traduit par une valeur unique (*le seuil*) de l'appréciation de la prononciation de l'apprenant, le jugement va appartenir à une seule région (correcte ou mauvaise prononciation) avec un degré de confiance de 0 ou 1. Mathématiquement, cela peut être représenté par l'équation suivante :

$$\mu_{Correcte}(PS) = 0$$
$$\mu_{Incorrecte}(PS) = 1$$

Mais dans le cas de l'évaluation de la prononciation basée sur la logique floue, comme montré dans la figure 6.2, la prononciation de l'apprenant peut être bonne et en même temps mauvaise selon différents degrés de confiance (seulement entre 0 et 1). Ceci représente la clé de cette méthode d'évaluation.

Dans la figure 6.2, on suppose que nous avons deux prononciations à évaluer et que ces deux prononciations ont obtenues deux scores à la suite de l'étape de reconnaissance

en mode alignement forcé, $S1$ et $S2$ respectivement. Le score $S2$ appartient à la région « prononciation correcte » avec un degré de confiance de 90% mais le score $S1$ est beaucoup plus intéressant car il appartient à la région « prononciation incorrecte » avec un degré de confiance de 60%, et avec un degré de confiance de 10% à la région « prononciation correcte ». Cela peut conduire à un rejet de la prononciation par le système ou à un feedback plus renseigné en prenant en considération les deux informations sur la région d'appartenance.

En effet, la figure 6.2 représente une fonction d'appartenance caractérisant la sortie du moteur de l'évaluation floue de la prononciation. Il s'agit de la qualité de la prononciation ou la conclusion à propos de la prononciation. La même chose est appliquée pour les entrés du moteur de l'évaluation floue de la prononciation afin de construire des fonctions d'appartenance caractérisant les entrées du système.

6.3 L'architecture du système proposé

L'architecture du système que nous proposons, illustrée par la figure 6.3, peut se résumer en trois étapes nécessaires :

1. La reconnaissance du signal de la parole, où le résultat de cette étape est un ensemble de scores automatiques (dans notre cas deux scores)
2. L'étape de *fuzzification* des entrées, qui consiste à remplacer les scores obtenus par des degrés d'appartenance aux ensembles définis relativement à chaque score
3. La combinaison floue des scores via des règles expertes
4. La *défuzzification* qui permet de retourner un score représentatif du niveau de la prononciation.

6.3.1 La fuzzification des variables d'entrée et de sortie

Des études précédentes ont montré que la moyenne globale du logarithme de vraisemblance ou GLL était une bonne mesure pour l'évaluation de la prononciation de la langue Arabe [42], ceci a été confirmé par nos résultats du quatrième chapitre qui ont montré que le GLL est une bonne mesure pour évaluer les différentes performances d'un locuteur sur une collection de mots. On relève aussi au travers des résultats trouvés dans le quatrième chapitre, que les deux scores : durée de la parole (TDS pour Time Duration of Speech) et moyenne globale du logarithme de vraisemblance (GLL pour Global average Log Likelihood) peuvent fournir des feedbacks informatifs et fiables pour les jeunes écoliers débutant l'apprentissage de la prononciation en Arabe et aussi pour l'Anglais (voir la section 4.4.2 et 4.4.3 pour plus de détails). C'est la raison pour laquelle l'entrée du

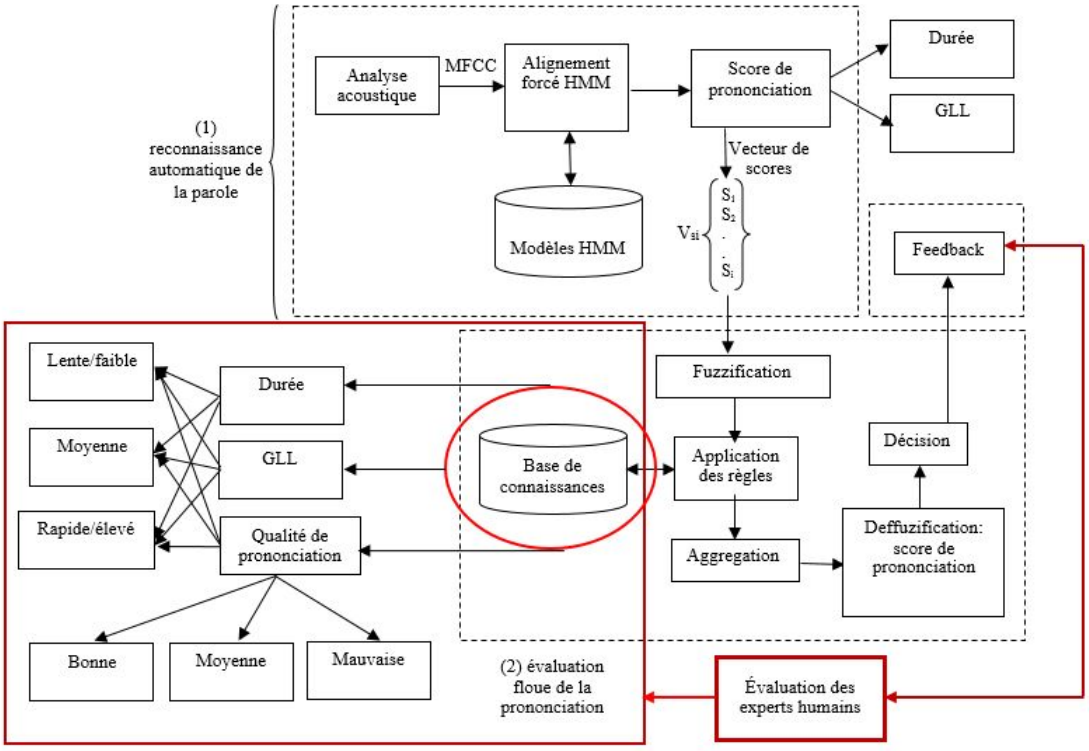

FIGURE *6.3* – L'architecture générale du système de combinaison floue des scores de l'évaluation

système d'évaluation floue de la prononciation en Arabe proposé, comme montré dans la figure 6.3, sera un vecteur de ces deux scores.

Nous supposons que la prononciation peut être classée en trois classes : bonne, moyenne et mauvaise (voir figure 6.4). De la même manière, la durée peut être longue, moyenne ou courte, le score basé sur le logarithme de vraisemblance peut être classé en petit, moyen ou élevé. Un exemple est montré dans la figure 6.4.

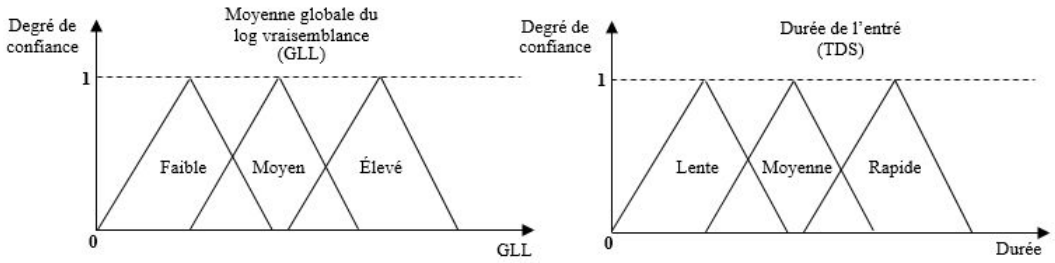

FIGURE *6.4* – Les fonctions d'appartenance définies pour les variables d'entré du système

En générale, le rôle d'un système d'enseignement de prononciation assisté par ordinateur est d'agir comme un tuteur. Ainsi, les limites des fonctions d'appartenance de la qualité de la prononciation ont été définies entre 0 et 10. Plus la valeur est petite, moins la chance que la prononciation est bien réalisée sera.

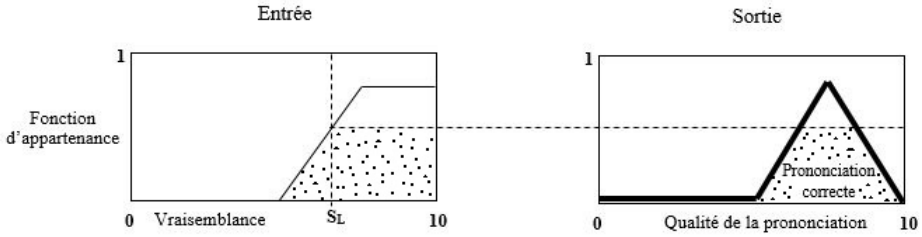

FIGURE *6.5* – Implication floue pour l'évaluation automatique de la prononciation

Une fois les fonctions d'appartenance des entrés et des sorties sont définit, le système d'évaluation floue de la prononciation proposé sera ensuite capable d'évaluer la prononciation des apprenants. Comme précédemment expliqué, l'évaluation sera représentée sous forme de feedback informatif à propos de la qualité globale de la prononciation (bonne, moyenne ou mauvaise) suivie par un score final (calculé à la base des deux entrés du moteur d'évaluation floue) nommé : le score défuzzification de prononciation DPS (Deffezzification Pronunciation Score).

6.3.2 La combinaison floue des scores

Le module de combinaison floue est la composante principale du système d'évaluation de la prononciation proposé. Il utilise les scores calculés pendant l'étape de la reconnaissance. Le but est de pouvoir ensuite les combiner en utilisant un ensemble de règles floues. Ces règles sont représentées sous forme de langage naturel comme dans le cas suivant :

$$If\ (S_1 \in A \wedge S_2 \in B) \Rightarrow (Pronunciation \in Correct \vee Pronunciation \in Wrong) \quad (6.1)$$

où S_1 et S_2 sont les scores générés à la fin de l'étape de la reconnaissance automatique de la parole. Avant cela, et en se basant sur les résultats précédemment obtenus, une base de connaissance a été construite. Cette base de connaissance consiste en un ensemble de règles. Un exemple de ces règles pourra être comme suit :

$$If\ (\textbf{log likelihood}\ based\ score\ is\ \textbf{hight}) \Rightarrow (\textbf{pronunciation}\ is\ \textbf{good}) \quad (6.2)$$

La règle précédente montre que la prononciation appartient à l'ensemble flou « good » à un certain degré de confiance qui dépend étroitement du degré d'appartenance de « log-likelihood » à l'ensemble floue prédéfinit « high ». Dans cette approche le score final de prononciation constitue en le résultat de l'étape de la défuzzification calculé (le score sur la qualité de la prononciation comme montré dans la figure 6.5).

6.3.3 Le score d'évaluation de la prononciation

Le score de défuzzification, qui représente un score final de prononciation, est déterminé en utilisant la méthode du centre de gravité (COG pour Center of Gravity). Assumant que μ est le degré d'appartenance et que le vecteur de score généré après l'alignement forcé de l'étape de la reconnaissance automatique de la parole a la forme suivante :

$$S = \{S_1, S_2, ...S_n\} \tag{6.3}$$

où S représente le vecteur de n scores dérivés. Comme précédemment mentionné, plusieurs scores peuvent être extraits à partir du moteur de la reconnaissance de la parole basé HMMs. Dans cette expérimentation, nous limiterons l'ensemble des entrés du moteur de l'évaluation floue de la prononciation à un ensemble de deux scores.

Après l'application des règles précédemment regroupées dans la base de connaissance, il ne reste qu'à prendre une décision finale à propos de la qualité de la prononciation. Pour illustrer comment le système d'évaluation floue de la prononciation réagit quant à la réception du signal de la parole, on assume que le moteur floue de décision implémenté est responsable de la génération des feedbacks en ce qui concerne la qualité de la prononciation. Si, par exemple, la durée prise pour produire de la parole est *longue*, et que la valeur du score basé sur la vraisemblance obtenue est *petite*, alors le système estime que la prononciation de l'apprenant est mauvaise. Le feedback concernant la qualité de la prononciation de l'apprenant est généré en deux formes : la première représente une des *variables linguistique* qui décrit la qualité de la prononciation (bonne, moyenne ou mauvaise). Tandis que la deuxième, elle sera la valeur du *centre de gravité* de la surface qui correspond à la combinaison des différents éléments du vecteur de scores lors de l'application des règles d'inférences.

Ce dernier est déterminé en tirant profit de la méthode du centre de gravité. En d'autres termes, le score DPS (Deffezzification Pronunciation Score) représente la conclusion à propos de l'intersection des opérations logiques (et, ou,...etc.) entre la durée de la prononciation (TDS) et ça moyenne globale du logarithme de vraisemblance correspondante GLL (le mapping des règles d'inférence). Si on prend un exemple ou la prononciation a été jugé bonne, selon les entrés du système d'évaluation floue, alors le score DPS peut être calculé en utilisant la formule (6.4) suivante :

$$DPS = COG_{Good.Pronunciation} = \frac{\sum_i \mu_{Good}(p)p}{\sum_i \mu_{Good}(p)} \tag{6.4}$$

où p est la valeur de la variable qui décrit la prononciation (bonne, moyenne ou mauvaise), $\mu_{Good}(p)$ est la fonction d'appartenance prédéfinit (ou le degré d'appartenance

correspondant) dans l'ensemble *pronunciation* pour la prononciation actuelle p à évaluer, et i représente le nombre des règles floues précédemment stockées dans la base de connaissance.

Dans cette expérimentation, nous avons utilisé des fonctions d'appartenance de type triangulaires pour représenter les classes (ou les variables linguistiques) des entrées/sorties du système d'évaluation floue de la prononciation. Si p est une variable continue (chose qui n'est pas le cas dans cette expérimentions dans le but de simplifier le calcule) alors le score DPS peut être calculé par l'équation (6.5) suivante :

$$DPS = COG_{Good.Pronunciation} = \frac{\int \mu_{Good}(p) p dp}{\int \mu_{Good}(p) dp} \tag{6.5}$$

6.4 Les résultats de l'évaluation floue de la prononciation

Dans ce qui suit, nous allons discuter et analyser les résultats de l'évaluation floue de la prononciation en Arabe et en Anglais des prononciations en se référant aux évaluations des experts précédemment effectuées.

6.4.1 La préparation de données

La technique d'évaluation floue de la prononciation a été appliquée pour l'évaluation de la prononciation en Arabe et en Anglais. Le but était de savoir si cette méthode s'adapte à l'évaluation de la prononciation de la majorité des langues ou c'est seulement pour une langue spécifique. Pour tester la méthode proposée sur la prononciation en Arabe, nous avons utilisé le modèle acoustique Arabe précédemment conçu et validé à travers les expérimentations effectuées et expliquées dans le quatrième chapitre (le tableau 4.4 illustre les résultats du test du modèle acoustique Arabe conçu). Pour le teste de la méthode proposée sur la prononciation en Anglais, nous avons utilisé le modèle acoustique WSJ (Wall Street Journal acoustical model) qui vient comme supplément avec le package de la reconnaissance automatique de la parole Sphinx 4.

Vue que, avant le teste de la méthode proposée, on disposait que de cinq échantillons de prononciation de mots en Anglais, on a gardé de même pour la prononciation en Arabe. Pour l'Anglais, 10 locuteurs ont été invités à prononcer cinq mots dont trois essais ont été réalisés pour chaque mot. Le total était de 150 mots à tester pour la prononciation en Anglais. Pour le teste de la prononciation en Arabe, 8 locuteurs ont été invités à prononcés cinq mots dont un seul essai a été réalisé pour chaque mot. Le totale était de 40 mots à tester pour l'évaluation de la prononciation en Arabe.

6.4.2 Résultats obtenus pour la prononciation en Anglais

Le tableau 6.1 résume les résultats obtenus pour le teste de la méthode de l'évaluation floue de la prononciation en Anglais de chacun des cinq mots. Il faut noter que, lors de l'évaluation de la prononciation en Anglais, la procédure d'attribution des appréciations par les experts humains n'était pas similaire à celle de la prononciation en Arabe. Vu qu'en l'absence des experts humains familiarisés avec la prononciation des phonèmes en Anglais, nous nous sommes contentés d'attribuer une appréciation générale sur la qualité globale de la prononciation. Le but ici était d'avoir la possibilité de faire une comparaison entre le score DPS attribué automatiquement par la méthode d'évaluation floue de la prononciation et l'appréciation des experts humains (bonne, moyenne ou mauvaise prononciation). À la fin, la précision de l'évaluation floue de la prononciation, par rapport à la classification des experts humains a montrée qu'un taux de 86.66% est obtenu, ce qui rend les performances du système de l'évaluation pour la prononciation en Anglais globalement acceptable.

TABLE *6.1* – Le score diffuzification de prononciation (DPS) *Vs.* évaluation des experts humains pour la prononciation en Anglais

Expert classification		1^{st} word DPS score			Average DPS	Total average
Good	speaker 1	8.625561	8.625561	8.625561	8.625560752	8.325112269
	speaker 5	7.723996	8.65645	8.306206	8.228883831	
	speaker 9	8.324414	8.309711	7.728551	8.120892224	
Medium	speaker 2	7.651174	5.428705	7.59894	6.892939715	8.029586874
	speaker 4	8.65895	8.660723	8.650656	8.656776491	
	speaker 7	8.657905	8.301324	8.657905	8.539044416	
Bad	speaker 3	4.35334	8.663787	7.793816	6.936981041	7.658667035
	speaker 6	8.655913	8.660021	8.315301	8.543745043	
	speaker 8	8.326709	8.326899	7.652684	8.102097181	
	speaker 10	8.657905	8.276887	4.220743	7.051844876	
Expert classification		2^{nd} word DPS score			Average DPS	Total average
Good	speaker 1	8.63208	7.701811	8.58593	8.306610472	5.729232435
	speaker 5	4.30443	5.784581	5.30044	5.129818861	
	speaker 9	2.5	2.558866	6.19493	3.751267973	
Medium	speaker 2	3.829108	7.55507	6.09900	5.827726957	5.395550434
	speaker 4	7.06465	4.37633	6.17169	5.870891908	
	speaker 7	4.4403	3.898513	5.125214	4.488032437	
Bad	speaker 3	4.44354	2.5	2.5	3.147847403	2.891188481
	speaker 6	4.35747	2.549747	2.5	3.135739427	
	speaker 8	2.5	2.5	2.5	2.5	
	speaker 10	2.5	2.5	3.34350	2.781167095	
Expert classification		3^{rd} word DPS score			Average DPS	Total average
Good	speaker 1	8.62578	7.545424	7.64229	7.93783412	4.99368925
	speaker 5	4.08722	2.943503	3.9324	3.654398541	
	speaker 9	2.5	2.91163	4.754867	3.388835089	
Medium	speaker 2	2.85070	4.279672	7.44065	4.85701219	3.854077355
	speaker 4	4.46747	4.685553	3.23941	4.130815391	
	speaker 7	2.72321	2.5	2.5	2.574404486	
Bad	speaker 3	4.408846	2.5	2.99673	3.301859479	3.931592163
	speaker 6	2.5	2.5	3.902383	2.967461161	
	speaker 8	3.82714	7.440655	3.70504	4.990946848	
	speaker 10	7.45289	3.445410	2.5	4.466101163	
Expert classification		4^{th} word DPS score			Average DPS	Total average
Good	speaker 1	8.619213	7.726752	7.48453	7.943501593	5.621147043
	speaker 5	2.90091	4.019687	4.38696	3.769191962	
	speaker 9	3.94739	5.736148	5.76869	5.150747573	
Medium	speaker 2	7.44065	7.369178	7.44065	7.416829603	5.237368354
	speaker 4	4.70361	4.388806	4.075006	4.389143623	
	speaker 7	3.29014	4.270547	4.157707	3.906131838	
Bad	speaker 3	2.56027	2.5	4.278952	3.113077352	3.311149474
	speaker 6	2.5	2.814858	2.5	2.604952988	
	speaker 8	4.503562	2.852512	2.784903	3.380326164	
	speaker 10	2.720977	6.234248	3.48349	4.146241391	

Expert classification		5^{th} word DPS score			Average DPS	Total average
Good	speaker 1	8.64918	8.595144	8.60195	8.615429129	7.902930387
	speaker 5	8.39034	8.398094	8.66376	8.484068573	
	speaker 9	2.5	8.663425	8.66445	6.60929346	
Medium	speaker 2	6.740609	0	8.66316	5.134592227	7.305747728
	speaker 4	8.66547	7.825711	8.66591	8.38570005	
	speaker 7	8.39809	8.394664	8.39809	8.396950908	
Bad	speaker 3	8.66504	7.825712	7.814366	8.101709037	7.519041516
	speaker 6	4.04385	8.398094	7.35706	6.59967274	
	speaker 8	7.76250	8.397604	7.394983	7.851696433	
	speaker 10	7.82571	7.396665	7.34688	7.523087851	

6.4.3 Résultats obtenus pour la prononciation en Arabe

D'une manière similaire à celle de la prononciation en Anglais, le tableau 6.2 suivant montre le résultat du teste de la méthode d'évaluation floue de la prononciation en Arabe, sauf que dans ce cas une seul réalisation de la prononciation pour chaque mot a été effectuée. Par rapport à la classification des experts humains, on peut dire que le système arrive à séparer une bonne prononciation de celle moyenne ou mauvaise. Si on prend le cas de la prononciation du quatrième mot (4^{th} word DPS score), le système d'évaluation floue de la prononciation a attribué une note de 8.03 pour les locuteurs dont leur prononciation a été jugée bonne par les experts humains, une note de 6.83 pour les locuteurs dont la prononciation a été jugée moyenne et une note de 4.24 pour les locuteurs dont la prononciation a été jugée mauvaise par le experts humains. Donc on remarque que c'était très bénéfique le fait d'implémenter un système d'évaluation de la prononciation qui se base principalement sur une analyse précédemment effectuée par des experts humains en prononciation.

À la fin, une précision de 86.66% a été obtenue, une performance globalement acceptable prenant en compte la taille des échantillons utilisée pour le teste de l'évaluation floue de la prononciation en Arabe. Ce qui rend l'analyse et l'étude de performance du système d'évaluation proposé plus pertinente c'est que ça aurait été mieux si les scores DPS attribués par le système floue d'évaluation de la prononciation ont été corrélés avec des notes attribués par les experts humains et non pas analyser ces scores par rapport aux classifications (ou appréciation) effectuées par les experts humains, et c'est ce que nous allons essayer d'illustrer dans la section suivante.

6.4.4 Résultats de corrélation entre les scores DPS et les scores des experts humains pour la prononciation en Arabe

Les mêmes résultats obtenus et mentionnés dans le tableau 6.2 concernant les scores automatique DPS attribués par la méthode d'évaluation floue de la prononciation ont

TABLE *6.2* – Le score diffuzification de prononciation (DPS) *Vs.* évaluation des experts humains (appréciation) pour la prononciation en Arabe

Expert classification		1^{st} word DPS score	Average DPS
Good	speaker1	8.596681	8.055708911
	speaker3	7.514737	
Medium	speaker2	7.511125	4.856159387
	speaker4	0	
	speaker7	7.057353	
Bad	speaker5	2.5	5.974305083
	speaker6	6.859678	
	speaker8	8.563237	
Expert classification		2^{nd} word DPS score	Average DPS
Good	speaker1	8.595022	7.778015489
	speaker3	6.961009	
Medium	speaker2	8.559524	5.303542364
	speaker4	0	
	speaker7	7.351103	
Bad	speaker5	2.5	2.5
	speaker6	2.5	
	speaker8	2.5	
Expert classification		3^{rd} word DPS score	Average DPS
Good	speaker1	8.532519	8.371854173
	speaker3	8.21119	
Medium	speaker2	7.461575	6.156949379
	speaker4	2.5	
	speaker7	8.509273	
Bad	speaker5	2.5	5.773731841
	speaker6	7.523304	
	speaker8	7.297891	
Expert classification		4^{th} word DPS score	Average DPS
Good	speaker1	8.563237	8.038090611
	speaker3	7.512944	
Medium	speaker2	7.269421	6.832666121
	speaker4	6.987808	
	speaker7	6.240769	
Bad	speaker5	2.5	4.247069359
	speaker6	7.741208	
	speaker8	2.5	
Expert classification		5^{th} word DPS score	Average DPS
Good	speaker1	8.622992	8.589372751
	speaker3	8.555753	
Medium	speaker2	6	7.729449202
	speaker4	8.596681	
	speaker7	8.591667	
Bad	speaker5	2.5	6.569503235
	speaker6	8.559524	
	speaker8	8.648986	

été comparés aux notes (et non pas aux appréciations) attribuées par les experts humains pour vérifier si un lien de corrélation existe. Pour la prononciation en Arabe, il se trouve que l'évaluation ici est plus claire par rapport aux résultats mentionnés dans le tableau 6.2. Les coefficients de corrélation obtenus entre DPS et les score attribués par les experts sont illustrés dans le tableau 6.3. Si on prend le même exemple interprété précédemment, i.e. le quatrième mot, une corrélation de 0.84 a été obtenue entre le score DPS et le troisième expert, une corrélation de 0.77 a été obtenue entre le score DPS et le deuxième expert (de même pour le premier expert), une chose qui prouve que les résultats de l'évaluation floue de la méthode proposée et un peu proche de celle des experts humains.

TABLE *6.3* – Corrélation entre les scores DPS et les notes attribuées par les experts humains

1^{st} word DPS score		Expert 1	Expert 2	Expert 3
speaker1	8.59668094	9	10	8
speaker2	7.511124986	6	6	7
speaker3	7.514736883	6	7	6
speaker4	0	5	3	5
speaker5	6.859677878	8	10	9
speaker6	7.057353176	6	9	9
speaker7	2.5	2	7	5
speaker8	8.563237372	5	7	7
Correlation DPS Vs. Experts		0.589934108	0.684909788	0.6838325
2^{nd} word DPS score		Expert 1	Expert 2	Expert 3
speaker1	8.595022399	9	10	8
speaker2	8.559523847	6	4	4
speaker3	6.961008579	6	5	6
speaker4	0	2	0	3
speaker5	2.5	6	8	8
speaker6	7.351103246	6	10	9
speaker7	2.5	0	4	3
speaker8	2.5	0	3	3
Correlation DPS Vs. Experts		0.765444785	0.621554177	0.525709206
3^{rd} word DPS score		Expert 1	Expert 2	Expert 3
speaker1	8.532518832	8	8	9
speaker2	7.461575156	4	7	5
speaker3	8.211189514	4	7	7
speaker4	2.5	5	7	4
speaker5	7.52330436	7	8	9
speaker6	8.509272982	6	10	9
speaker7	2.5	1	3	3
speaker8	7.297891163	5	10	8
Correlation DPS Vs. Experts		0.614221141	0.690508992	0.853088035

4^{th} word DPS score		Expert 1	Expert 2	Expert 3
speaker1	8.563237321	9	7	9
speaker2	7.269421126	3	8	5
speaker3	7.512943901	3	7	8
speaker4	6.9878084	5	9	7
speaker5	7.741208078	8	10	9
speaker6	6.240768839	6	10	9
speaker7	2.5	1	3	3
speaker8	2.5	1	4	3
Correlation DPS Vs. Experts		0.778627998	0.776926246	0.848005364
5^{th} word DPS score		Expert 1	Expert 2	Expert 3
speaker1	8.622992229	10	7	7
speaker2	6	5	5	4
speaker3	8.555753274	5	8	8
speaker4	8.596680933	8	7	4
speaker5	8.559523836	7	10	9
speaker6	8.591666672	6	10	9
speaker7	2.5	5	9	7
speaker8	8.648985868	4	7	7
Correlation DPS Vs. Experts		0.353167424	0.016522958	0.216131357

6.5 Conclusion

Nous avons pu constater, à travers le présent chapitre, qu'un outil d'évaluation automatique de la prononciation autonome doit fortement dépendre des évaluations des experts en prononciation pour assurer une performance acceptable. Les résultats trouvés ont été globalement acceptables, même en la présence de fausses alarmes. Cela s'explique par le fait que le modèle acoustique Arabe conçu n'est pas encore assez générique pour s'adapter bien à un tel outil et pour pouvoir contribuer ainsi à générer des résultats encore plus prometteurs.

Conclusion et perspectives

7.1 Bilan

De par les objectifs définis au début de notre travail, notre contribution dans cette thèse revêt plusieurs aspects.

D'abord, il faut rappeler que le but est la conception et la réalisation d'un outil d'évaluation de la prononciation en Arabe qui arrive à séparer entre une bonne et une mauvaise prononciation, en fournissant aux apprenants de la prononciation en Arabe un *feedback* correctif simple est compréhensible sur la qualité globale de la prononciation.

Notre première contribution a consisté en l'utilisation de l'outil Sphinx qui pour la première fois a été utilisé au niveau national dans un contexte d'apprentissage de la prononciation. En utilisant cet outil nous avons pu construire des modèles acoustiques des phonèmes Arabes. Une autre réalisation, que nous avons pu accomplir est la construction d'une base d'enregistrement d'écoliers dans un contexte d'apprentissage des sons de l'Arabe, où nous avons pris soin de leur proposer des mots ou de petites phrases à lire qui comprennent certaines des difficultés que rencontrent les écoliers dans leurs premiers apprentissages de lecture, telles que l'éllongation des voyelles.

En ce qui est du travail d'évaluation, nous avons pu récolter un ensemble d'évaluation des experts, ces données pourraient faire l'objet d'étude plus avancées sur la variabilité des évaluations *intra* et *inter* experts.

Pour nos investigations, nous les avons commencé par une étude comparative entre les scores automatiques qui existent pour l'évaluation de la prononciation, cette étude a conclu à l'émergence de deux scores et surtout elle nous a montré qu'un score bien établit dans ce domaine qui est le GOP n'est pas le meilleur pour l'Arabe ; ceci probablement en raison de la qualité des modèles acoustiques qui restent à améliorer, la combinaison d'autres scores pourraient alors apporter une solution pour des langues peu dotées de ressources comme la parole Arabe.

Quant aux propositions que nous faisons dans le cadre de notre thèse, nous mentionnons deux : D'abord l'utilisation d'un test statistique pour l'établissement du seuil d'acceptation d'une prononciation ; cette approche nous semble à la fois novatrice et prometteuse et mérite d'être poursuivie plus en avant par l'investigation d'autre test statistique. La seconde proposition tire son intérêt de la variabilité des évaluations des experts humains, elle préconise l'utilisation des ensembles flous pour permettre plus de flexibilité entre les frontières d'une bonne et d'une mauvaise prononciation.

7.2 Perspectives relatives au système de reconnaissance

Nous avons constaté et souligné dès le début de ce travail, que disposer d'un système de reconnaissance performant est un grand pas vers la conception d'un système d'évaluation de la prononciation fiable. Et si nous avons constaté dès le départ que cela n'est pas trop difficile pour une langue telle que l'Anglais pour laquelle nous disposons d'une part de bases de parole trop importante dont la base *TIMIT*, et si nous le voulons nous pouvons utiliser les modèles acoustiques des phonèmes fournis par Sphinx et qui sont construits sur la base *TIMIT*. Pour l'Arabe très peu de bases sont disponibles et même celles qui le sont, elles ne sont pas très grandes. Nous ne connaissons pas de *Benchmark* non plus ou de compétition pour l'Arabe où nous pouvons comparer les performances d'un système de reconnaissance dédié à l'Arabe. Une des premières perspectives qui doit s'inscrire dans l'immédiat est la construction de bases conséquentes pour la langue Arabe et la validation de modèles acoustiques des phonèmes qui pourront être utilisés par l'ensemble de la communauté des chercheurs et développeurs qui travaillent sur la parole Arabe.

7.3 Perspectives relatives au mode d'évaluation

En ce qui est de l'évaluation, le plus grand reste à faire car les travaux sur cette thématique sont quasi inexistants pour l'Arabe. Et tout reste à faire aussi bien sur le plan de la recherche où il faut disposer de modèle qui convient à la langue Arabe. Il faut donc repartir sur les scores d'évaluation qu'il faut resituer par rapport à l'Arabe, il faut aussi faire des travaux sur la corrélation des évaluations des experts dans un contexte de l'Arabe. Nous pourrons peut être observé des phénomènes plus ou moins accentués que pour des langues comme l'Anglais.

7.4 Perspectives relatives à l'enseignement assisté par ordinateur

Finalement, d'autres perspectives peuvent être envisagées qui relèveraient du domaine de l'enseignement assisté par ordinateur, car même si nous nous sommes intéressés dans le contexte de cette thèse aux aspects techniques de l'évaluation de la prononciation par le biais de scores automatiques, il est tout à fait opportun d'enrichir ce travail en le plaçant dans un contexte d'un environnement d'apprentissage avec toutes les études qui conviennent sur les interfaces adéquates mais aussi par la mise au point d'un profil dynamique des apprenants et tout ce que un système d'apprentissage suppose.

Bibliographie

[1] A. Farghaly and K. Shaalan, "Arabic natural language processing : Challenges and solutions," *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 8, no. 4, pp. 14 :1–14 :22, dec 2009. [Online]. Available : http://doi.acm.org/10.1145/1644879.1644881

[2] K. Necibi, H. Bahi, and T. Sari, "Speech disorders recognition using speech analysis," *Speech, Image, and Language Processing for Human Computer Interaction : Multi-Modal Advancements*, pp. 310–324, 2012.

[3] H. Bahi, "Nessr : un système neuroexpert pour la reconnaissance de la parole," *GRETSI, Saint Martin d'Hères, France*, vol. 24, no. 1, pp. 59–67, 2007.

[4] H. Frihia and H. Bahi, "Etude comparative entre les librairies de reconnaissance vocale," in *Proceedings of National Conference on Speech Processing*, Algeria, 2014, pp. 36–42.

[5] SPHINX. (2015) Sphinx toolkit. [Online]. Available : http://cmusphinx.sourceforge.net/wiki/tutorialsphinx4

[6] P. Lamere, P. Kwok, W. Walker, E. B. Gouvêa, R. Singh, B. Raj, and P. Wolf, "Design of the cmu sphinx-4 decoder." in *INTERSPEECH*. Citeseer, 2003.

[7] HTK. (2015) Htk toolkit. [Online]. Available : http://htk.eng.cam.ac.uk/

[8] S. J. Young and S. Young, *The HTK hidden Markov model toolkit : Design and philosophy*. Citeseer, 1993.

[9] K. Samudravijaya and M. Barot, "A comparison of public-domain software tools for speech recognition," in *Workshop on spoken language processing*, 2003.

[10] H. Yang, C. Oehlke, and C. Meinel, "German speech recognition : A solution for the analysis and processing of lecture recordings," in *Computer and Information Science (ICIS), 2011 IEEE/ACIS 10th International Conference on*. IEEE, 2011, pp. 201–206.

[11] J. Baker, "The dragon system–an overview," *Acoustics, speech and signal processing, IEEE transactions on*, vol. 23, no. 1, pp. 24–29, 1975.

[12] B. Lowere, "The harpy speech recognition system," *PhD thesis, Carnegie Mellon University*, 1976.

[13] J. K. Baker, "Stochastic modeling for automatic speech understanding," in *Readings in speech recognition*. Morgan Kaufmann Publishers Inc., 1990, pp. 297–307.

[14] K.-F. Lee, H.-W. Hon, and R. Reddy, "An overview of the sphinx speech recognition system," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 38, no. 1, pp. 35–45, 1990.

[15] X. Huang, F. Alleva, H.-W. Hon, M.-Y. Hwang, K.-F. Lee, and R. Rosenfeld, "The sphinx-ii speech recognition system : an overview," *Computer Speech & Language*, vol. 7, no. 2, pp. 137–148, 1993.

[16] P. Placeway, S. Chen, M. Eskenazi, U. Jain, V. Parikh, B. Raj, M. Ravishankar, R. Rosenfeld, K. Seymore, M. Siegler *et al.*, "The 1996 hub-4 sphinx-3 system," in *Proc. DARPA Speech recognition workshop*. Citeseer, 1997, pp. 85–89.

[17] M. Ravishankar, "Some results on search complexity vs accuracy," in *DARPA Speech Recognition Workshop*. Citeseer, 1997, pp. 104–107.

[18] W. Walker, P. Lamere, P. Kwok, B. Raj, R. Singh, E. Gouvea, P. Wolf, and J. Woelfel, "Sphinx-4 : A flexible open source framework for speech recognition," 2004.

[19] M. Ordowski, N. Deshmukh, A. Ganapathiraju, J. Hamaker, and J. Picone, "A public domain speech-to-text system." in *EUROSPEECH*. Citeseer, 1999.

[20] M. Mohri, "Finite-state transducers in language and speech processing," *Computational linguistics*, vol. 23, no. 2, pp. 269–311, 1997.

[21] P. Clarkson and R. Rosenfeld, "Statistical language modeling using the cmu-cambridge toolkit." in *Eurospeech*, vol. 97, 1997, pp. 2707–2710.

[22] S. J. Young, N. Russell, and J. Thornton, *Token passing : a simple conceptual model for connected speech recognition systems*. Citeseer, 1989.

[23] S. CMUCLMTK. (2015) Sphinx toolkit. [Online]. Available : http://cmusphinx.sourceforge.net/wiki/tutoriallm

[24] J. H. Underwood, *Linguistics, Computers, and the Language Teacher. A Communicative Approach.* ERIC, 1984.

[25] C.-y. Chao, S. Seneff, and C. Wang, "An interactive interpretation game for learning chinese." in *SLaTE*. Citeseer, 2007, pp. 41–44.

[26] L. Neumeyer, H. Franco, V. Digalakis, and M. Weintraub, "Automatic scoring of pronunciation quality," *Speech communication*, vol. 30, no. 2, pp. 83–93, 2000.

[27] J. Bernstein, M. Cohen, H. Murveit, D. Rtischev, and M. Weintraub, "Automatic evaluation and training in english pronunciation." in *ICSLP*, vol. 90, 1990, pp. 1185–1188.

[28] L. Neumeyer, H. Franco, M. Weintraub, and P. Price, "Automatic text-independent pronunciation scoring of foreign language student speech," in *Spoken Language, 1996.*

ICSLP 96. Proceedings., Fourth International Conference on, vol. 3. IEEE, 1996, pp. 1457–1460.

[29] H. Franco, V. Abrash, K. Precoda, H. Bratt, R. Rao, J. Butzberger, R. Rossier, and F. Cesari, "The sri eduspeak tm system : Recognition and pronunciation scoring for language learning," *Proceedings of InSTILL 2000*, pp. 123–128, 2000.

[30] C. Cucchiarini, F. De Wet, H. Strik, and L. Boves, "Assessment of dutch pronunciation by means of automatic speech recognition technology." in *ICSLP*, vol. 5. Citeseer, 1998, pp. 1739–1742.

[31] V. V. Digalakis, "Segment-based stochastic models of spectral dynamics for continuous speech recognition," 1992.

[32] S. Witt and S. Young, "Computer-assisted pronunciation teaching based on automatic speech recognition," *Language Teaching and Language Technology Groningen, The Netherlands*, 1997.

[33] S. M. Witt and S. J. Young, "Phone-level pronunciation scoring and assessment for interactive language learning," *Speech communication*, vol. 30, no. 2, pp. 95–108, 2000.

[34] J. Bernstein, J. De Jong, D. Pisoni, and B. Townshend, "Two experiments on automatic scoring of spoken language proficiency," in *Proceedings of InSTIL2000 : Integrating Speech Technology in Learning*, 2000, pp. 57–61.

[35] C. Cucchiarini, H. Strik, and L. Boves, "Different aspects of expert pronunciation quality ratings and their relation to scores produced by speech recognition algorithms," *Speech Communication*, vol. 30, no. 2, pp. 109–119, 2000.

[36] B. Mak, M. Siu, M. Ng, Y.-C. Tam, Y.-C. Chan, K.-W. Chan, K.-Y. Leung, S. Ho, F.-H. Chong, J. Wong *et al.*, "Plaser : pronunciation learning via automatic speech recognition," in *Proceedings of the HLT-NAACL 03 workshop on Building educational applications using natural language processing-Volume 2*. Association for Computational Linguistics, 2003, pp. 23–29.

[37] J. Cheng, Y. Z. D'Antilio, X. Chen, and J. Bernstein, "Automatic assessment of the speech of young english learners," *ACL 2014*, p. 12, 2014.

[38] H. Franco, H. Bratt, R. Rossier, V. R. Gadde, E. Shriberg, V. Abrash, and K. Precoda, "Eduspeak® : A speech recognition and pronunciation scoring toolkit for computer-aided language learning applications," *Language Testing*, vol. 27, no. 3, pp. 401–418, 2010.

[39] N. Moustroufas and V. Digalakis, "Automatic pronunciation evaluation of foreign speakers using unknown text," *Computer Speech & Language*, vol. 21, no. 1, pp. 219–230, 2007.

[40] H. Dahan, A. Hussin, Z. Razak, and M. Odelha, "Automatic arabic pronunciation scoring for language instruction," 2011.

[41] A. Ahmad Khan, O. Mourad, A. M. K. B. Mannan, H. B. A. M. Dahan, and M. Abushariah, "Automatic arabic pronunciation scoring for computer aided language learning," in *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on.* IEEE, 2013, pp. 1–6.

[42] K. Necibi and H. Bahi, "An arabic mispronunciation detection system by means of automatic speech recognition technology," in *The 13th International Arab Conference on Information Technoloy Proceedings*, 2012, pp. 303–308.

[43] F. Ehsani, J. Bernstein, A. Najmi, and O. Todic, "Subarashii : Japanese interactive spoken language education," in *Fifth European Conference on Speech Communication and Technology*, 1997.

[44] J. Bernstein, A. Najmi, and F. Ehsani, "Subarashii : Encounters in japanese spoken language education," *CALICO Journal*, vol. 16, no. 3, pp. 361–384, 1999.

[45] W. L. Johnson and A. Valente, "Tactical language and culture training systems : Using artificial intelligence to teach foreign languages and cultures." in *AAAI*, 2008, pp. 1632–1639.

[46] W. L. Johnson, C. Beal, A. Fowles-Winkler, U. Lauper, S. Marsella, S. Narayanan, D. Papachristou, and H. Vilhjálmsson, "Tactical language training system : An interim report," in *Intelligent Tutoring Systems.* Springer, 2004, pp. 336–345.

[47] W. L. Johnson, S. Marsella, N. Mote, H. Viljhálmsson, S. Narayanan, and S. Choi, "Tactical language training system : Supporting the rapid acquisition of foreign language and cultural skills," in *InSTIL/ICALL Symposium 2004*, 2004.

[48] W. L. Johnson, S. Marsella, and H. Vilhjalmsson, "The darwars tactical language training system," in *Proceedings of I/ITSEC*, 2004.

[49] A. Raux and M. Eskenazi, "Using task-oriented spoken dialogue systems for language learning : potential, practical applications and challenges," in *InSTIL/ICALL Symposium 2004*, 2004.

[50] C. Cucchiarini, J. Van Doremalen, and H. Strik, "Disco : development and integration of speech technology into courseware for language learning." in *INTERSPEECH*, 2008, pp. 2791–2794.

[51] S. Chevalier and Z. Cao, "Application and evaluation of speech technologies in language learning : experiments with the saybot player." in *INTERSPEECH*, 2008, pp. 2811–2814.

[52] H. Franco and L. Neumeyer, "Calibration of machine scores for pronunciation grading." in *ICSLP*, 1998.

[53] H. Franco, L. Neumeyer, V. Digalakis, and O. Ronen, "Combination of machine scores for automatic grading of pronunciation quality," *Speech Communication*, vol. 30, no. 2–3, pp. 121 – 130, 2000. [Online]. Available : http://www.sciencedirect.com/science/article/pii/S016763939900045X

[54] K. P. Truong, A. Neri, F. De Wet, C. Cucchiarini, and H. Strik, "Automatic detection of frequent pronunciation errors made by l2-learners." in *INTERSPEECH*, Lisbon, Protugal, 2005, pp. 1345–1348.

[55] J. Tepperman and S. Narayanan, "Using articulatory representations to detect segmental errors in nonnative pronunciation," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, no. 1, pp. 8–22, 2008.

[56] A. Ito, Y.-L. Lim, M. Suzuki, and S. Makino, "Pronunciation error detection method based on error rule clustering using a decision tree," 2005.

[57] S. Wei, G. Hu, Y. Hu, and R.-H. Wang, "A new method for mispronunciation detection using support vector machine based on pronunciation space models," *Speech Communication*, vol. 51, no. 10, pp. 896–905, 2009.

[58] I. Odriozola, E. Navas, I. Hernáez, I. Sainz, I. Saratxaga, J. Sánchez, and D. Erro, "Using an asr database to design a pronunciation evaluation system in basque." in *LREC*, 2012, pp. 4122–4126.

[59] N. Minematsu, "Yet another acoustic representation of speech sounds," in *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 1. IEEE, 2004, pp. I–585.

[60] R. Jakobson and L. R. Waugh, *The sound shape of language*. Walter de Gruyter, 2002.

[61] J. R. Allen, "Individualizing foreign language instruction with computers at dartmouth." *Foreign Language Annals*, vol. 5, no. 3, pp. 348–349, 1972.

[62] F. Ge, F. Pan, C. Liu, B. Dong, S.-d. Chan, X. Zhu, and Y. Yan, "An svm-based mandarin pronunciation quality assessment system," in *The Sixth International Symposium on Neural Networks (ISNN 2009)*. Springer, 2009, pp. 255–265.

[63] Y. Kondo, E. Tsutsui, and M. Nakano, "Bridging the gap between l2 research and classroom practice (2) : Evaluation of automatic scoring system for l2 speech," in *Second Language Studies : Acquisition, Learning, Education and Technology*, Tokyo, Japan, 2010, pp. 2–5.

[64] H. Franco, L. Neumeyer, Y. Kim, and O. Ronen, "Automatic pronunciation scoring for language instruction," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 2. IEEE, 1997, pp. 1471–1474.

[65] O. Ronen, L. Neumeyer, and H. Franco, "Automatic detection of mispronunciation for language instruction." in *EUROSPEECH*, 1997.

[66] H. Franco, L. Neumeyer, M. Ramos, and H. Bratt, "Automatic detection of phone-level mispronunciation for language learning." in *EUROSPEECH*, 1999.

[67] S. Xu, J. Jiang, Z. Chen, and B. Xu, "Automatic pronunciation error detection based on linguistic knowledge and pronunciation space," in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on.* IEEE, 2009, pp. 4841–4844.

[68] A. Raux and T. Kawahara, "Automatic intelligibility assessment and diagnosis of critical pronunciation errors for computer-assisted pronunciation learning." in *International Conference of Spoken Language Processing*, Denver, CO, USA, 2002, pp. 737–740.

[69] M. Eskenazi and S. Hansma, "The fluency pronunciation trainer," *Proc. Speech Technology in Language Learning*, pp. 77–80, 1998.

[70] M. Eskenazi, "Using automatic speech processing for foreign language pronunciation tutoring : Some issues and a prototype," *Language learning & technology*, vol. 2, no. 2, pp. 62–76, 1999.

[71] Y. Tsubota, M. Dantsuji, and T. Kawahara, "Practical use of autonomous english pronunciation learning system for japanese students," in *InSTIL/ICALL Symposium 2004*, 2004.

[72] J. Dalby and D. Kewley-Port, "Explicit pronunciation training using automatic speech recognition technology," *CALICO Journal*, vol. 16, no. 3, 1999.

[73] J. Dalby, D. Kewley-Port, and R. Sillings, "Language-specific pronunciation training using the hearsay system," in *STiLL-Speech Technology in Language Learning*, 1998.

[74] J.-m. Kim, C. Wang, M. Peabody, and S. Seneff, "An interactive english pronunciation dictionary for korean learners," in *proceedings of Interspeech*, 2004, pp. 1677–1680.

[75] A. M. Harrison, W.-K. Lo, X. Qian, and H. Meng, "Implementation of an extended recognition network for mispronunciation detection and diagnosis in computer-assisted pronunciation training." in *SLaTE*, 2009, pp. 45–48.

[76] F. Pan, Q. Zhao, and Y. Yan, "New machine scores and their combinations for automatic mandarin phonetic pronunciation quality assessment," in *Knowledge-Based Intelligent Information and Engineering Systems.* Springer, 2007, pp. 821–830.

[77] P. Fuping, Q. Zhao, and Y. Yan, "Mandarin vowel pronunciation quality evaluation by a novel formant classification method and its combination with traditional algorithms," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on.* IEEE, 2008, pp. 5061–5064.

[78] K. Probst, Y. Ke, and M. Eskenazi, "Enhancing foreign language tutors–in search of the golden speaker," *Speech Communication*, vol. 37, no. 3, pp. 161–173, 2002.

[79] W. Menzel, D. Herron, P. Bonaventura, and R. Morton, "Automatic detection and correction of non-native english pronunciations," *Proceedings of INSTILL*, pp. 49–56, 2000.

[80] S. M. Abdou, S. E. Hamid, M. Rashwan, A. Samir, O. Abdel-Hamid, M. Shahin, and W. Nazih, "Computer aided pronunciation learning system using speech recognition techniques," in *INTERSPEECH*, 2006.

[81] J. Chen, L. Wang, C. Li, J. Hu, and S. Li, "Iels : A computer assisted pronunciation training system for undergraduate students," in *Education Technology and Computer (ICETC), 2010 2nd International Conference on*, vol. 1. IEEE, 2010, pp. 338–342.

[82] W.-K. Lo, A. M. Harrison, and H. Meng, "Statistical phone duration modeling to filter for intact utterances in a computer-assisted pronunciation training system," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5238–5241.

[83] J. Anderson-Hsieh, R. Johnson, and K. Koehler, "The relationship between native speaker judgments of nonnative pronunciation and deviance in segmentais, prosody, and syllable structure," *Language learning*, vol. 42, no. 4, pp. 529–555, 1992.

[84] W. Labov, "The social stratification of english in new york city." Ph.D. dissertation, Columbia university., 1964.

[85] J. Cheng, J. Bernstein, U. Pado, and M. Suzuki, "Automatic assessment of spoken modern standard arabic," in *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*. Association for Computational Linguistics, 2009, pp. 1–9.

[86] K. Necibi and H. Bahi, "A statistical-based decision for arabic pronunciation assessment," *International Journal of Speech Technology*, vol. 18, no. 1, pp. 37–44, 2015. [Online]. Available : http://dx.doi.org/10.1007/s10772-014-9248-2

[87] H. Husni and Z. Jamaluddin, "A retrospective and future look at speech recognition applications in assisting children with reading disabilities," in *Proceedings of the world Congress on Engineering and Computer Science, San Francisco, Estados Unidos*. Citeseer, 2008.

Production Scientifiques

K. Necibi, H. Bahi, and T. Sari, "Speech disorders recognition using speech analysis," Speech, Image, and Language Processing for Human Computer Interaction : Multi- Modal Advancements, pp. 310–324, 2012. [Online]. Available : http://www.igi-global.com/chapter/speech-disorders-recognition-using-speech/65065

K. Necibi and H. Bahi, "An arabic mispronunciation detection system by means of automatic speech recognition technology," in The 13th International Arab Conference on Information Technoloy Proceedings, 2012, pp. 303–308. [Online]. Available : http://www.acit2k.org/ACIT/2012Proceedings/13356.pdf

K. Necibi and H. Bahi, "An ASR-based System for Arabic Mispronunciation Detection," The International Journal of Information Technology & Computer Science, vol. 6, no. 2, pp. 36-45, 2012. [Online]. Available : http://ijitcs.com/volume%206_No_2/Khaled+Necibi.php.

K. Necibi and H. Bahi, "A statistical-based decision for arabic pronunciation assessment," International Journal of Speech Technology, vol. 18, no. 1, pp. 37–44, 2015. [Online]. Available : http://dx.doi.org/10.1007/s10772-014-9248-2

ANNEXES

La dyslexie chez les jeunes écoliers

Les avancées technologiques réalisées dans les différents domaines de l'informatique ont fait que de plus en plus les professionnels de la santé cherchent à introduire cet outil aussi bien dans le diagnostic que dans la thérapie de nombreux troubles moteurs, neurologiques ou autres.

Nous voulons présenter dans cette annexe à notre thèse, un environnement où nos travaux devraient pouvoir s'intégrer, il s'agit d'un système de détection de la dyslexie. Les réflexions et les réalisations attenantes à cet environnement ont été pensées dans le cadre d'un projet national de recherche, intitulé : Détection automatique de la dyslexie chez de jeunes écoliers.

A.1 Introduction

La dyslexie est un trouble de la parole dont l'élément révélateur est la lecture. Or, la lecture est aussi l'apprentissage le plus laborieux et le plus fondamental qui est demandé au jeune élève ; il est donc évident que si cet apprentissage est déficient c'est toute la scolarité de l'enfant qui est compromise, et probablement plus tard son avenir social et professionnel.

La dyslexie est une difficulté durable de l'apprentissage du langage écrit et de l'acquisition de ses automatismes, chez des enfants intelligents, normalement scolarisés, indemnes de troubles sensoriels et de troubles psychologiques. Elle altère la capacité à identifier les mots et serait présente quel que soit l'environnement social, culturel, éducatif et pédagogique de l'enfant. *Ni les parents, ni les enseignants ne sont responsables de ce trouble spécifique d'apprentissage ; mais ils ne doivent pas l'ignorer.* On estime qu'elle touche environ 6% à 8% d'écoliers appartenant à tous les milieux sociaux avec des degrés de sévérité variables. La prise en compte des troubles individuels d'apprentissage permet de lutter efficacement contre les décrochages scolaires et l'échec en permettant aux enfants dyslexiques de ne pas perdre confiance en eux et de garder une image positive d'eux-mêmes. Il est important que les parents ne se sentent pas seuls face aux difficultés rencontrées par leur enfant et surtout qu'ils ne culpabilisent pas. Il est donc essentiel que ces enfants soient détectés le plus tôt possible afin de pouvoir organiser au mieux leur accompagnement. Il faut aussi que les professionnels (orthophonistes, médecins,...etc.), l'équipe éducative et les parents travaillent en étroite collaboration.

Au travers de ce projet, et avec la généralisation de l'outil informatique, il sera apporté un outil de détection de ce trouble à la communauté de l'éducation nationale et aux familles des jeunes élèves en difficulté scolaire.

Comme la dyslexie se caractérise par des aptitudes normales (parfois en dessus de la normale) en calcul mais par une mal-lecture due au fait que l'élève n'accède pas au sens de ce qu'il lit, l'environnement propose une détection automatique de ce trouble par un

ensemble de tests.

Le logiciel comprend deux composantes informatiques : un système de détection qui consiste en la mise en œuvre d'une batterie de test en adéquation avec les éléments révélateurs du trouble, et un système de décision qui sur la base des résultats obtenus décidera de l'existence ou de l'absence du trouble. Cet outil est un logiciel ludique et pédagogique qui permettra de cerner les difficultés de l'enfant. Notre contribution d'un système d'évaluation de la prononciation devrait alors être intégrée parmi les tests, et les scores obtenus participer au processus de décision. Nous allons dans ce qui suit expliquer un peu plus ce projet et notre positionnement.

A.2 Position du problème

La dyslexie est un trouble du langage très répandu mais tout aussi méconnu à travers le monde. En effet, les statistiques montrent que ce trouble touche près de 4% de la population d'un pays [87], mais c'est un trouble qui demeure méconnu pour beaucoup d'instituteurs et d'enseignants. Et même lorsque le trouble est décelé sa prise en charge est fort coûteuse pour les parents, et à ce titre le témoignage d'une maman canadienne est très poignant car après avoir longuement souffert avec son fils jusqu'à rencontrer une personne qui a pu mettre le doigt sur ce trouble, elle se trouve confronté aux coûts exorbitants dans les écoles spécialisés, ce qui la poussait à choisir entre un de ces deux enfants (elle avait deux enfants dyslexiques), car elle ne pouvait payer que les frais d'inscription que d'un seul de ses enfants.

Ainsi, l'introduction de l'outil informatique et à fortiori de la RAP (Reconnaissance Automatique de la Parole) dans la prise en charge de la dyslexie ne peut qu'être bénéfique à la société car en plus d'être beaucoup moins cher que la présence d'un spécialiste qui n'est pas toujours disponible mais aussi car il est connu que l'apprentissage assisté par ordinateur est beaucoup moins stressant pour un enfant qu'un milieu classique.

A.3 Éléments conceptuels du projet

La figure A.1 illustre l'architecture générale du système de détection de la dyslexie proposé. Le logiciel réalisé comprend deux composantes informatiques : un système de détection qui consiste en la mise en œuvre d'une batterie de test en adéquation avec les éléments révélateurs du trouble, et un système de décision qui sur la base des résultats obtenus décidera de l'existence ou de l'absence du trouble. Cet outil est un logiciel ludique et pédagogique qui permet de cerner les difficultés de l'enfant.

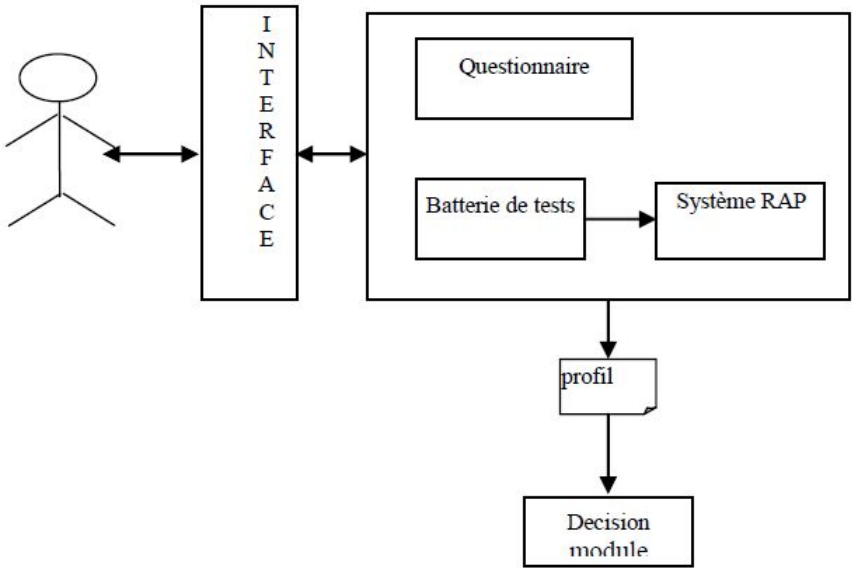

FIGURE *A.1* – Architecture du système de détection de dyslexie

A.4 Batterie de test

Le choix des tests a été fait de telle manière que les niveaux soient évolutifs, c'est-à-dire que les exercices de mathématiques, de compréhensions et de grammaire pour le premier niveau sont très faciles, par la suite, la difficulté augmente par niveau et par matière afin d'apprécier à leurs juste valeurs les aptitudes des élèves soumis aux différents tests. Les figures A.2, A.3, A.4, A.5 et A.6 montrent tels exemples.

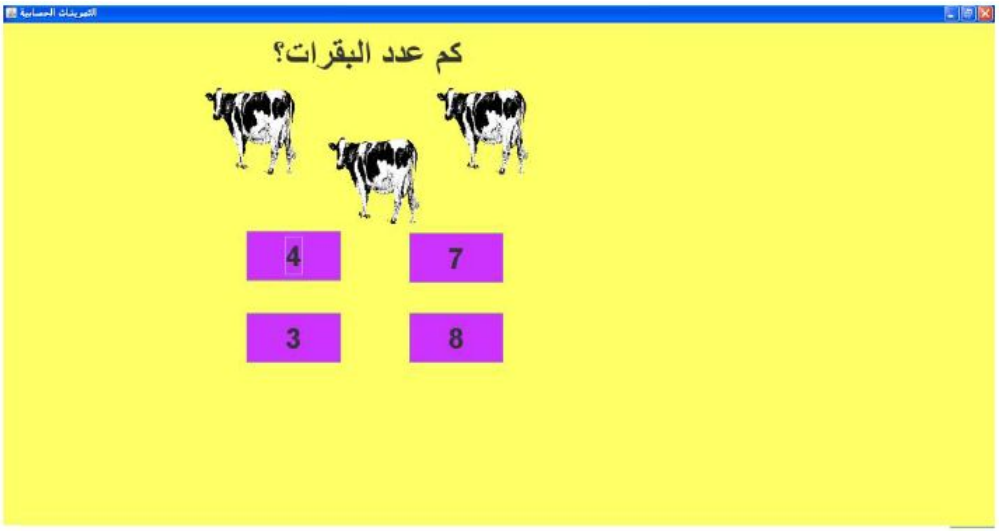

FIGURE *A.2* – Exercice de calcul (niveau 1)

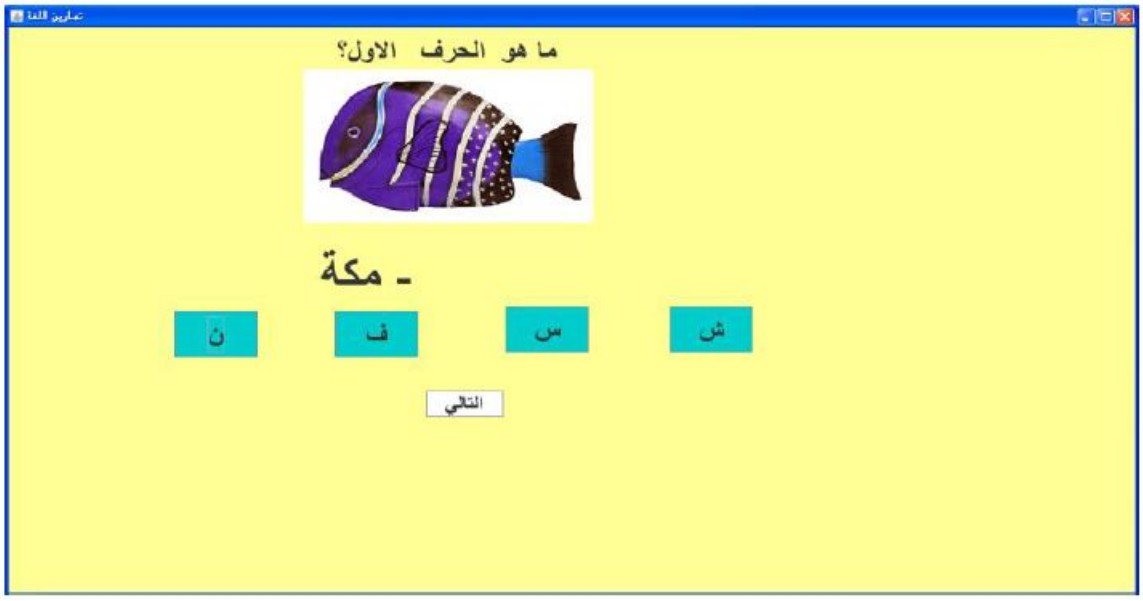

FIGURE *A.3* – Exercice de lecture (niveau 1)

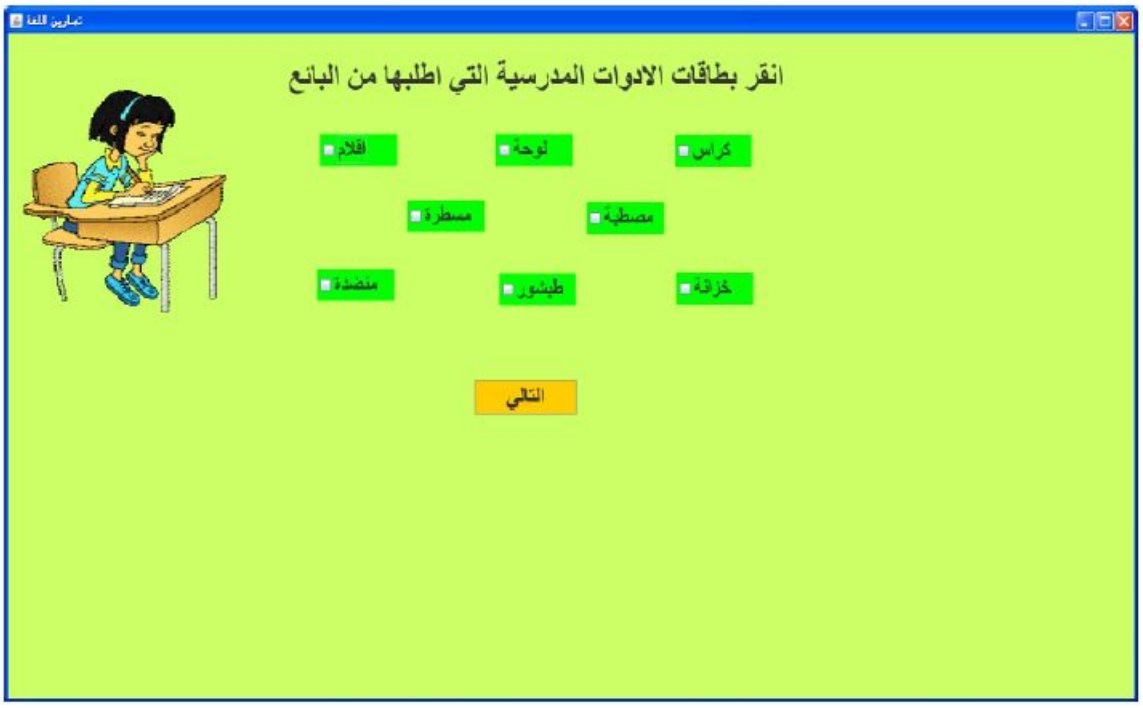

FIGURE *A.4* – Exercice de langue (niveau 2)

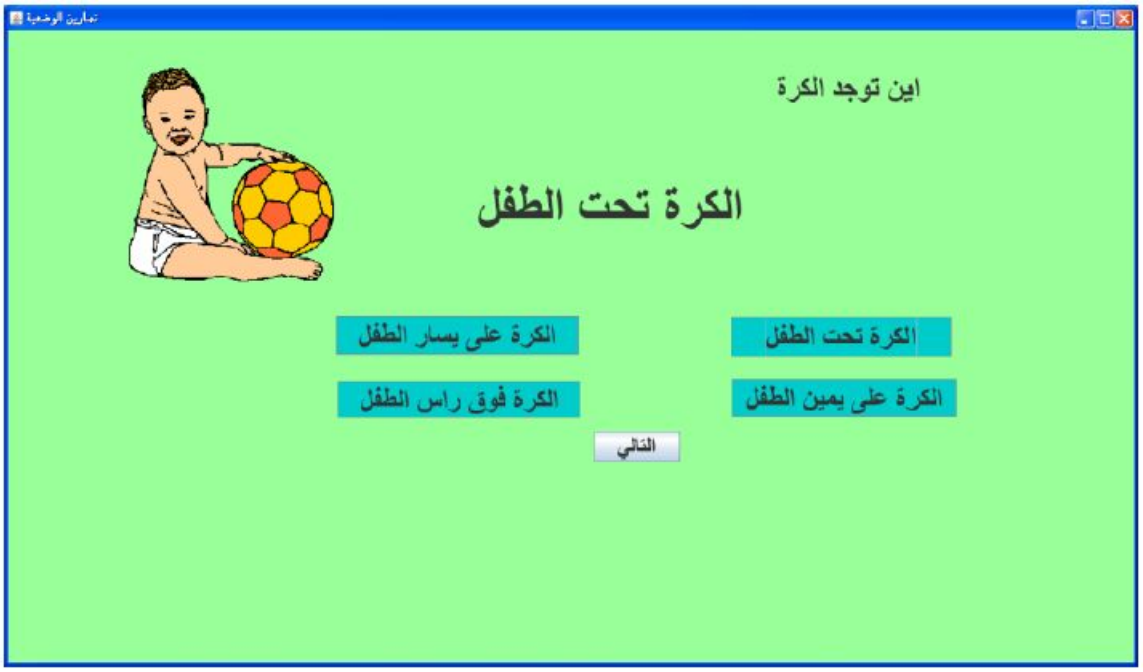

FIGURE *A.5* – Exercice de positionnement (niveau 1)

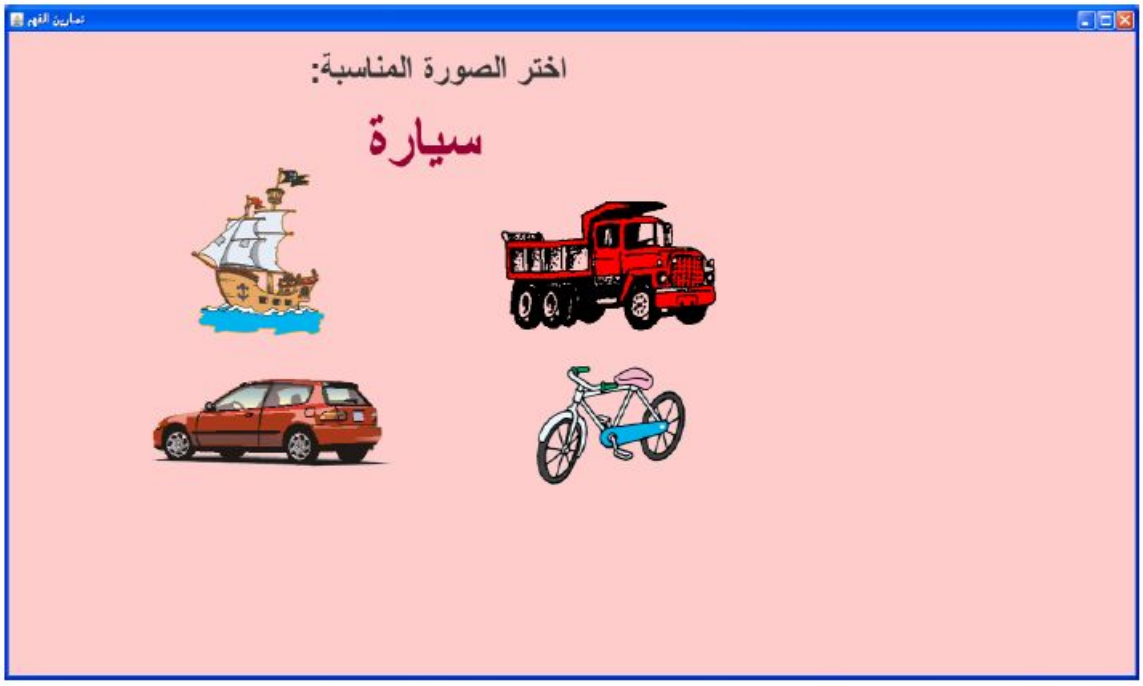

FIGURE *A.6* – Exercice de compréhension (niveau 1)

A.5 Le raisonnement basé cas (CBR ; Case Based Reasoning)

Comme nous l'avons précédemment souligné, le système est formé de deux composantes, une première partie regroupant les outils d'évaluation des aptitudes de l'enfant, comprenant en particulier le module de reconnaissance et une seconde composante qui consiste en le module de décision qui sur la base des éléments fournis par cette panoplie d'outils doit décider de l'absence ou de la présence du trouble (il peut arriver que le système n'arrive pas à décider).

Vu la complexité du profil de l'enfant, incluant parfois l'absence de certaines informations, l'absence de règles systématiques pour poser un diagnostic, les concepteurs du projet ont choisi le raisonnement à base de cas à la base du module de décision.

A.5.1 Structure d'un cas

La description du cas dans ce système est répartie sur deux dimensions. La première concerne la partie problème et contient les descripteurs issus de la phase d'évaluation. La seconde partie est la conséquence du problème, décrite en termes de diagnostic et conduite à tenir. Toutes les informations nécessaires liées à la structure d'un cas se trouvent dans la figure A.7

- Partie problème : elle est constituée du profil de l'enfant. Elle comprend la réponse aux questions relatives à la situation familiale et les aptitudes physiques de l'enfant. Elle inclut également ses réponses aux différents tests de calcul, de lecture,...etc.
- Partie conséquence : elle contient le diagnostic, quant à l'existence ou l'absence du trouble. Dans le cas d'absence de dyslexie cette partie peut contenir une éventuelle explication des difficultés que présente l'enfant.

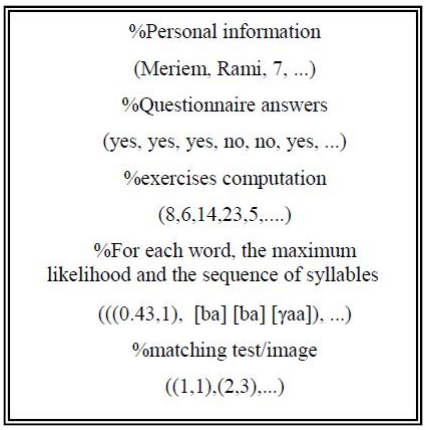

FIGURE *A.7* – Un exemple de profil (nouveau cas)

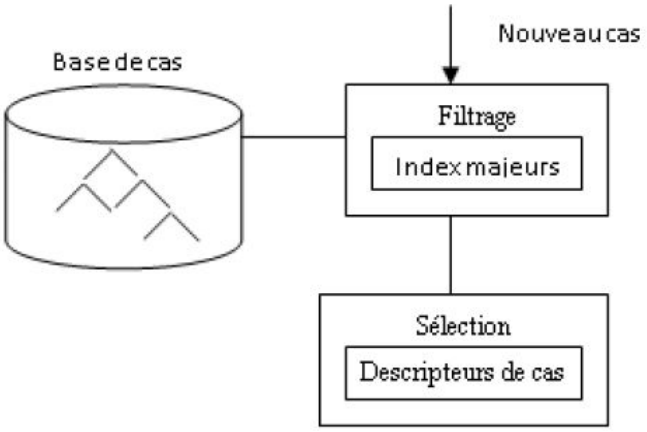

FIGURE *A.8* – Organigramme de recherche de cas similaires

A.5.2 Recherche de cas similaire

Une étape préliminaire à la recherche de cas similaire consiste en le filtrage. En guise de *filtrage* de cas, nous avons utilisé un index « majeur » (figure A.8) qui est le *niveau* de chaque test pour accélérer la recherche, et réduire le nombre de cas à comparer.

Après le filtrage, la recherche de cas similaire se poursuit par le calcul de la distance de Hamming. *La Distance de Hamming* est utilisée en télécommunication pour compter le nombre de bits altérés dans la transmission d'un message d'une longueur donnée.

A.5.3 Méthode d'adaptation

L'adaptation est une transformation de la solution pour satisfaire les exigences du nouveau contexte, mais la solution proposée est souvent inadéquate, car elle ne tient pas compte de tout le contexte (figure A.9).

Parmi les solutions adaptatives, nous retenons celles qui concernent le système de reconnaissance de la parole où les concepteurs permettent certaines permutations de sons pour poursuivre l'adaptation. Le système calcule toujours la distance d'adaptation en utilisant la même méthode utilisée dans la distance de Hamming avec les transformations effectuées par le système.

Si l'élève choisit une bonne réponse ou bien une réponse adaptée à la solution le système affecte à toutes les réponses un 0 sinon 1.

Le système calcule le nombre de 1 de chaque vecteur d'adaptation, si la valeur retenue est égale à 0, le système considère la décision et l'observation de ce cas comme un diagnostic final, sinon il choisit la décision du vecteur avec la plus petite valeur retenue.

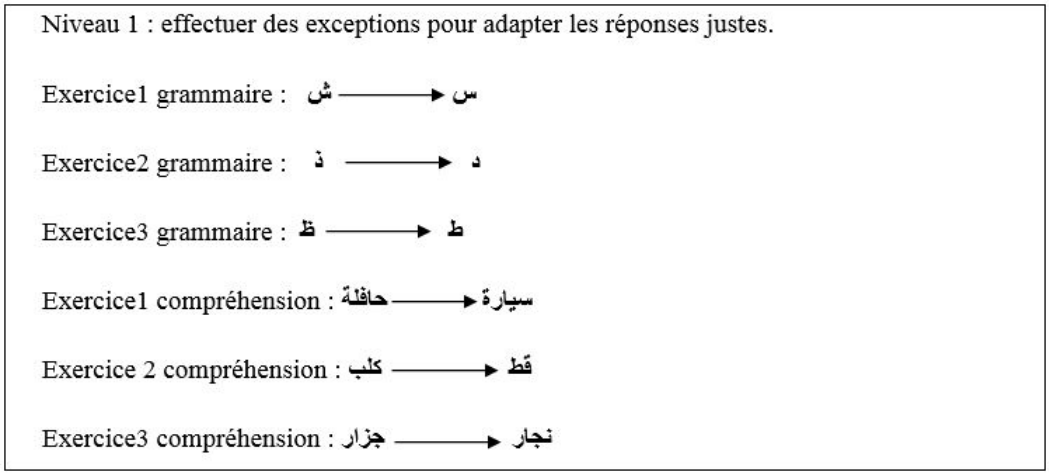

FIGURE *A.9* – L'adaptation des réponses

A.6 Conclusion

C'est ce type de projet qui a motivé notre travail, pour contribuer à l'avancement du développement de notre société d'une manière concrète et compétitive, et nous souhaitons poursuivre notre contribution par le déploiement de telles applications.