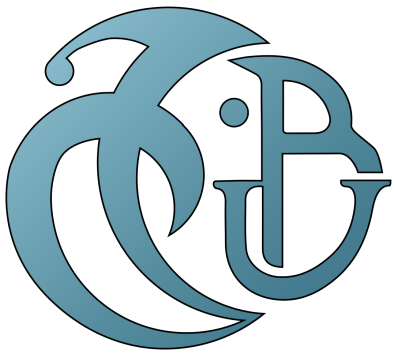
Université de Blida 1
Faculté des sciences
Département d'informatique



Mémoire de fin d'études
Pour l'obtention du diplôme de Master en Informatique
Option : Ingénierie des Logiciels

Développement d'une voix arabe synthétique open source.

Réalisé par
HEMINA Karim
HEMINNA Oussama

Encadré par
Mme.OUAHRANI L.
Dr.ZERROUKI Taha

Membres de jury
Président : CHERIF Zahar
Examinateur : FERFERA Sofiane

Présenté le 10 septembre 2020

Résumé

Ce mémoire décrit la construction d'une voix arabe synthétique à l'aide du modèle de Markov caché implémenté par le système HTS, avec Festival TTS comme module de traitement linguistique et de synthétisation. Cette voix sera accessible en open source, et elle a un large éventail d'applications dans la vie quotidienne, qui touchent différents groupes de personnes dont les aveugles et les sourds-muets. Nous définissons les scripts nécessaires pour effectuer le traitement linguistique, et nous réalisons les étapes nécessaires pour la construction de la voix par HTS. Les résultats de l'évaluation indiquent que les utilisateurs ont identifié les mots et phrases avec une moyenne de 8,97/10 et ont noté avec 7,2/10 la qualité de la voix.

Mots-clés : texte au parole, langue arabe, synthèse vocale

Abstract

This thesis describes the construction of an arabic synthesized voice based on the Hidden Markov Model (HMM) implemented in the HTS toolkit, with Festival TTS as linguistic analyzer and speech synthesizer. This voice is accessible on open source, it has a wide range of applications in daily life, affecting different groups of people including the blind and deaf. We define the necessary scripts to perform the linguistic processing, and we perform the necessary steps for the construction of the voice by HTS. The evaluation results shows that the users could identify the spoken words with an average of 8.97/10 and they noted the quality of the voice with 7.2/10.

Key words : Text to speech, arabic language, speech synthesis

ملخص

تصف المذكرة بناء صوت عربي اصطناعي باستخدام نموذج ماركوف المخفي، الذي تم إنشاءه بواسطة النظام HTS واستعمال نظام فيستفال كوحدة للمعالجة اللغوية والتوليف. سيكون هذا الصوت مفتوح المصدر, وله مجموعة واسعة من التطبيقات في الحياة اليومية التي تمس مختلف الفئات، بما في ذلك فئة المكفوفين والصم البكم. نقوم بتعريف الأدوات اللازمة لإجراء المعالجة اللغوية، و بإنجاز الخطوات الضرورية لبناء الصوت عن طريق النظام HTS .تشير نتائج التقييم إلى أن المستخدمين حددوا الكلمات والعبارات بمعدل 8.97 من 10 وصنفوا جودة الصوت بـ 7.2 من 10

كلمات مفتاحية : تحويل نص إلى كلام, اللغة العربية, توليف الكلام

Remerciements

Je tiens à remercier Monsieur Taha ZERROUKI pour nous avoir proposé ce thème, nous avoir encadré durant l'élaboration de ce projet et pour ses précieux conseils et recommandations tout au long de ces mois et surtout pour nous avoir initié à un nouveau domaine de la synthèse vocale et du traitement de la langue arabe. Un grand merci à vous monsieur. Je n'oublierais jamais vos bons conseils et vos encouragements continus.

J'adresse toute ma reconnaissance et mes remerciements les plus profonds à Madame L. OUAHRANI pour avoir dirigé et supervisé ce mémoire.

Je remercie tous les membres de mon club "IT Community", ma deuxième famille, pour toutes ces années. Je remercie également tous mes enseignants durant mes cinq ans à l'Université de Blida 1.

J'exprime aussi mes remerciements à toutes ces personnes qui travaillent sur le domaine de l'open-source pour la langue arabe ainsi que toute personne ayant contribué et aidé, de près ou de loin, à la réalisation de ce travail.

Mes remerciements finaux et non les moindres vont à toute ma famille qui m'a toujours soutenu, mes frères A. HEMINA, M. HEMINA et S. HEMINA et plus particulièrement mes parents A. HEMINA et D. HEMINNA dont l'affection, l'amour, le soutien et l'encouragement constants m'ont été d'un grand réconfort et ont contribué à l'aboutissement de ce travail.

HEMINA Karim

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je voudrais tout d'abord adresser toute ma reconnaissance à madame L. OUAHRANI et monsieur T. ZERROUKI, pour leur patience, leur disponibilité et surtout leurs conseils avisés, qui ont contribué à alimenter ma réflexion. Merci pour votre aide.

Je dédie ce travail à mes parents et mes sœurs qui m'ont apporté leur soutien avec tout ce qu'ils possèdent, je ne vous remercierai jamais assez.

Un grand merci à mon père O. HEMINNA pour ses conseils concernant mon style d'écriture, il a grandement facilité mon travail.

Enfin, je tiens à remercier tout mes amis et toute les personnes de mon entourage, pour leur soutien moral tout au long de mon travail.

HEMINNA Oussama

Table des matières

Introduction Générale **2**

I Etat de l'art 4

1 Synthèse de la parole à partir du texte (TTS) **5**

1.1 Introduction 5
1.2 Principe de la synthèse vocale 6
1.2.1 Traitement linguistique (front-end) 6
1.2.1.1 L'analyse de texte 7
1.2.1.2 L'analyse phonétique 8
1.2.1.3 L'analyse prosodique 9
1.2.2 La synthétisation vocale (Back-end) 9
1.3 Les techniques de la synthèse vocale 9
1.3.1 La synthèse articulatoire 10
1.3.2 La synthèse par règles (synthèse du formant) 11
1.3.3 La synthèse par concaténation 12
1.3.4 La synthèse statistique paramétrique 13
1.3.4.1 Synthèse basée sur le Modèle de Markov Caché (HMM) 14
1.3.4.2 Synthèse basée sur les réseaux de neurones profonds (DNN) 15
1.4 Étude comparative des techniques de la synthèse vocale 17
1.5 Les applications des systèmes de synthèse vocale 18
1.5.1 Interface vocale (IHM) 19
1.5.2 Accessibilité 19
1.5.3 Applications éducatives 19
1.6 Conclusion 19

2 La phonologie de la langue arabe **21**

2.1 Introduction 21

2.2 La langue arabe 22
2.3 L'écriture arabe 22
2.3.1 Les consonnes "الحروف الساكنة" 24
2.3.2 Les voyelles "حروف العِلَّة" 24
2.3.3 Le signe de diacritique "Sukun السكون" 25
2.3.4 Les diacritiques doubles "التَّنوين" 25
2.3.5 La gémination "الشَّدة" 26
2.3.6 La ponctuation 26
2.4 La phonétique arabe "الصوتيات العربية" 27
2.4.1 Les caractéristiques phonétiques des consonnes 27
2.4.2 Les caractéristiques phonétiques des voyelles 30
2.5 Les règles de prononciation 31
2.6 Le problème des diacritiques 33
2.7 Conclusion 33

3 Les systèmes de synthèse vocale Open Source 35
3.1 Introduction 35
3.2 Les systèmes de synthèse open-source 36
3.2.1 ESpeak 36
3.2.1.1 Les caractéristiques de eSpeak 36
3.2.1.2 Architecture et fonctionnement pour la langue Arabe 36
3.2.2 Festival TTS 38
3.2.2.1 Utilisation 38
3.2.2.2 Créer une nouvelle voix 38
3.2.2.3 La langue arabe pour Festival TTS 39
3.2.3 Mary TTS 39
3.2.3.1 Architecture 39
3.2.3.2 La langue arabe pour Mary TTS 41
3.2.4 Tacotron 41
3.2.4.1 Séquence à séquence (seq2seq) 41
3.2.4.2 Architecture 42
3.2.4.3 La langue arabe pour Tacotron 43
3.3 Comparaison 44
3.4 Conclusion 44

II Conception et Implémentation 46

4 Conception 47

4.1 Introduction 47
4.2 Choix du système 47
4.3 Architecture générale du système 50
4.4 La préparation d'un corpus de la parole naturelle 51
4.5 L'analyse de texte 51
4.5.1 Définition de la liste des phonèmes 53
4.5.2 Tokenisation et normalisation du texte 53
4.5.3 Lettres-au-son 53
4.5.4 Le lexique 54
4.5.5 L'annotation 55
4.6 La production de la voix par HTS 55
4.6.1 La phase d'entraînement 56
4.6.2 La phase de synthétisation 59
4.7 L'intégration de la voix dans Festival 60
4.8 Conclusion 60

5 Implémentation 61
5.1 Introduction 61
5.2 Les outils utilisés 62
5.3 Les démarches de travail 65
5.3.1 Traitement linguistique 67
5.3.1.1 Installation 67
5.3.1.2 Préparation 68
5.3.1.3 Configuration 70
5.3.1.4 Exécution 78
5.3.2 La production de la voix par HTS 78
5.3.2.1 Installation 79
5.3.2.2 Configuration 80
5.3.2.3 L'extraction des caractéristiques acoustiques . . 82
5.3.2.4 L'étiquetage 82
5.3.2.5 Nœuds de l'arbre de décision 85
5.3.2.6 L'entraînement 86
5.3.3 L'intégration de la voix créée au système de synthèse vocale Festival 89
5.3.4 L'amélioration de la qualité vocale générée 90
5.4 Applications 90
5.5 Conclusion 91

6 Test et évaluation **92**
6.1 Introduction 92
6.2 Méthodologie 92
6.3 Résultats 95
6.4 Discussion 100
6.5 Conclusion 100

Conclusion Générale **102**

Bibliographie **104**

Table des figures

1.1 Procédure de la synthèse vocale.[2] 6
1.2 Les tâches du front-end.[3] 7
1.3 Les techniques de la synthèse vocale. 10
1.4 Machine de Von Kempelen. 10
1.5 Structure de base du synthétiseur de formants en cascade.[2] . . . 11
1.6 Structure de base du synthétiseur de formants parallèle.[2] 12
1.7 Schéma fonctionnel d'un synthétiseur de concaténation.[3] 13
1.8 Modèle de gauche à droite à trois états.[8] 15
1.9 Un système de synthèse vocale basé sur un DNN.[11] 16

2.1 Signe de diacritique "السكون" 25
2.2 Gémination 26
2.3 Caractéristiques phonétiques des voyelles arabes 31

3.1 Étapes de la synthétisation de eSpeak [6] 37
3.2 Organigramme de eSpeak arabe [6] 37
3.3 Architecture de Mary TTS[27] 40
3.4 Un réseau seq2seq simple effectuant la traduction automatique de l'anglais vers l'allemand[30] 42
3.5 Architecture du modèle.[29] 42
3.6 Module CBHG.[29] 43

4.1 Choix de la technique de synthèse vocale 49
4.2 Architecture générale du système. 50
4.3 L'analyse de texte 52
4.4 Les tâches de l'analyse de texte. 53
4.5 Un aperçu d'un système de synthèse vocale basé sur HMM [35] . 55
4.6 La phase d'entraînement. 56
4.7 Arbres de décision pour le clustering de contexte[3] 58
4.8 La phase de synthétisation 59

5.1 Démarches de la réalisation. 66

5.2 L'analyse de texte. 67
5.3 Préparation du corpus pour l'analyse de texte. 69
5.4 Configuration des fichiers Scheme. 70
5.5 Production de la voix par HTS. 78
5.6 Configuration de HTS. 81
5.7 Extraction des caractéristiques acoustiques. 82
5.8 Étiquetage. 83
5.9 Création des questions. 86
5.10 La formation vocale. 87

6.1 Pourcentage de participants parlant l'arabe. 95
6.2 Acuité visuelle des participants. 96
6.3 Représentation graphique des moyennes du test d'identification. . 97
6.4 La moyenne générale du test d'identification. 97
6.5 Représentation graphique des moyennes du test de qualité. 98
6.6 La moyenne générale du test de qualité. 99
6.7 Nombre de tests avec et sans commentaire. 99
6.8 Nombre d'e-mails reçus. 100

Liste des tableaux

1.1 Résumé des avantages et inconvénients des différentes techniques de la synthèse vocale 18

2.1 Lettres arabes[7] 23
2.2 Autres lettres arabes particulières[7] 24
2.3 Voyelles courtes 25
2.4 Voyelles longues 25
2.5 Diacritiques doubles 26
2.6 Caractéristiques phonétiques des consonnes arabes 30
2.7 Différentes prononciations d'un mot selon sa vocalisation. 33

3.1 Les caractéristiques des systèmes de synthèse vocale 44

4.1 Exemple de l'entrée et la sortie des lettre-au-son 54
4.2 Exemples des mots irrégulier 54

5.1 Outils utilisés. 62

6.1 Moyennes du test d'identification 96
6.2 Moyennes du test de qualité 98

Abréviations

TTS (Text-To-Speech)	Synthèse de la parole à partir du texte.
HMM (Hidden Markov Model)	Modèle de Markov caché.
DNN (Deep Neural Networks)	Réseaux de neurones profonds.

Introduction Générale

Fournir aux machines la capacité de parler comme des humains, c'est le défi de la synthèse vocale.

La parole est le moyen de communication le plus naturel chez l'homme, qui l'a intégré dans ses interactions avec la machine. Cela a été rendu possible grâce aux effort consentis en technologie de la parole.

La technologie de la parole est l'un des principaux domaines du langage naturel, elle est composée de techniques pour la synthèse vocale, la reconnaissance vocale et les systèmes de dialogue. La synthèse vocale donne aux ordinateurs la capacité de communiquer avec les utilisateurs par la voix en convertissant un texte écrit en parole.

La synthèse vocale n'est pas une nouvelle technologie, les premières tentatives de construction d'un synthétiseur vocal mécanique ont déjà commencé en 1700. Au cours de la dernière décennie, la synthèse vocale est devenue si naturelle qu'un auditeur croirait écouter une voix humaine. Il a fallu beaucoup de temps et de travaux pour obtenir ce résultat. Au début l'idée était d'utiliser des modèles physiques du tractus vocal en utilisant des synthétiseurs de formants. De nombreuses années de recherche ont permit de perfectionner l'encapsulation des propriétés acoustiques du tractus vocal. Puis, avec l'aide d'une technologie informatique plus puissante, il est devenu viable d'utiliser directement des extraits de discours enregistrés et de les coller ensemble pour créer de nouvelles phrases sous la forme de synthétiseurs de concaténation. Maintenant, avec les progrès de la technologie, nous pouvons utiliser des méthodes probabilistes et paramétriques pour générer une voix synthétisée de très haute qualité avec des technologies telles que le modèle de Markov caché ou les réseaux de neurones profonds.

La langue arabe a accusé un grand retard dans les domaines de la technologie et de l'informatique. Il faut donc travailler à son développement, afin que le monde arabe puisse l'utiliser. L'objectif de ce projet est justement de développer une

voix synthétique open source de haute qualité pour la langue arabe. Les systèmes commerciaux sont trop chers et non accessibles, d'où la nécessité de l'alternative open-source. Cette voix peut être utilisée dans de nombreuses applications telles que des applications éducatifs ou à des fins académiques. La voix synthétisée doit être compréhensible par les auditeurs humains et proche de la voix naturelle.

Ce mémoire est organisé en deux parties comprenant chacune trois chapitres. La première partie est consacrée à l'état de l'art où nous donnons un aperçu complet du domaine étudié et nous enquêtons sur les projets existants. Nous avons commencé par un chapitre sur la synthèse vocale, ce qu'elle signifie, quelles sont les différentes technologies et méthodes utilisées et quelles sont les applications qui impliquent la synthèse vocale et s'en servent. Dans le deuxième chapitre, nous aborderons la phonologie arabe, son écriture ainsi que ses règles et propriétés phonologiques afin d'extraire les informations nécessaires dont nous avons besoin pour implémenter notre système. Nous terminons la première partie par un chapitre sur les systèmes de synthèse open-source existants, où nous étudions les caractéristiques de chacun d'entre eux et les travaux préexistants sur l'arabe pour chaque système pour aboutir à une comparaison entre ces différents systèmes.

La deuxième partie décrit le travail que nous avons effectué pour créer une voix arabe. Le premier chapitre porte sur l'étude conceptuelle de notre système, où nous expliquons notre choix des technologies utilisées tout en présentant l'architecture générale du système et les différentes tâches que nous devons mettre en œuvre. Le deuxième chapitre montre les différentes étapes que nous avons traversées pour la production d'une nouvelle voix, et comment elle peut être utilisée. Le dernier chapitre de cette partie et de ce mémoire est consacré à l'évaluation des résultats obtenus par un test de la voix créée. Différents utilisateurs ont participé à ce test en remplissant le formulaire d'évaluation.

Première partie

Etat de l'art

Chapitre 1

Synthèse de la parole à partir du texte (TTS)

1.1 Introduction

La parole est le principal moyen de communication entre les personnes. La synthèse de la parole à partir du texte (synthèse vocale) désigne l'ensemble des traitements permettant à une machine de transformer un énoncé de l'écrit à l'oral. Elle a pour but de générer une voix synthétique intelligible et naturelle qui lit n'importe quel texte et imite au mieux la voix humaine.

Ils existent plusieurs techniques de synthèse vocale. Nous en parlerons de chacune par la suite. En ce qui concerne notre intérêt à synthétiser automatiquement de nouvelles phrases, ils existent des différences fondamentales entre le système abordé ici et les autres machines parlantes comme le lecteur de cassettes par exemple.

La synthèse vocale fait appel à de multiples connaissances : informatique (architecture logicielle, temps réel...), linguistique (analyse lexical, morphologique, syntaxique ...), traitement du signal, etc. Ces technologies sont très importantes pour répondre aux besoins humains concernant la communication, l'apprentissage et le contrôle des dispositifs électroniques.

Dans ce chapitre, nous présenterons le principe de la synthèse vocale, où nous allons expliquer les composants de tels systèmes, puis les différentes technologies utilisées dans ce domaine. Nous terminerons en faisant référence à des applications de la parole synthétique.

1.2 Principe de la synthèse vocale

La procédure de la synthèse vocale comprend deux principaux modules[1].

- Un module de traitement linguistique appelé front-end, capable de produire une transcription phonétique du texte à lire.
- Un module de synthétisation appelé back-end, qui transforme les unités phonétiques générées par le précédent module en parole , en utilisant des techniques de synthèse vocale.

Ces deux modules sont indépendants, ce qui donne la possibilité de choisir parmi différents front-end et back-end, à condition qu'ils soient compatibles entre eux.

Une version simplifiée de cette procédure est illustrée à la figure 1.1.

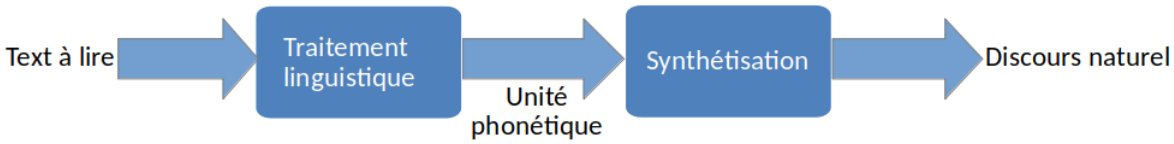


FIGURE 1.1 : Procédure de la synthèse vocale.[2]

L'entré comprend des données textuelles à synthétiser, provenant par exemple d'un article ou d'un livre électronique, d'un contenu de site web ou d'un e-mail. Ensuite, ces données passent par plusieurs étapes de traitement linguistique pour générer leurs transcriptions phonétiques et leurs informations prosodiques, afin de les transformer en parole par un module de synthétisation. La qualité du résultat de la synthèse vocale dépend de la technique utilisée.

1.2.1 Traitement linguistique (front-end)

Le traitement linguistique comprend trois tâches principales (figure 1.2), prenant le texte comme entrée et créant une série de phonèmes comme sortie.

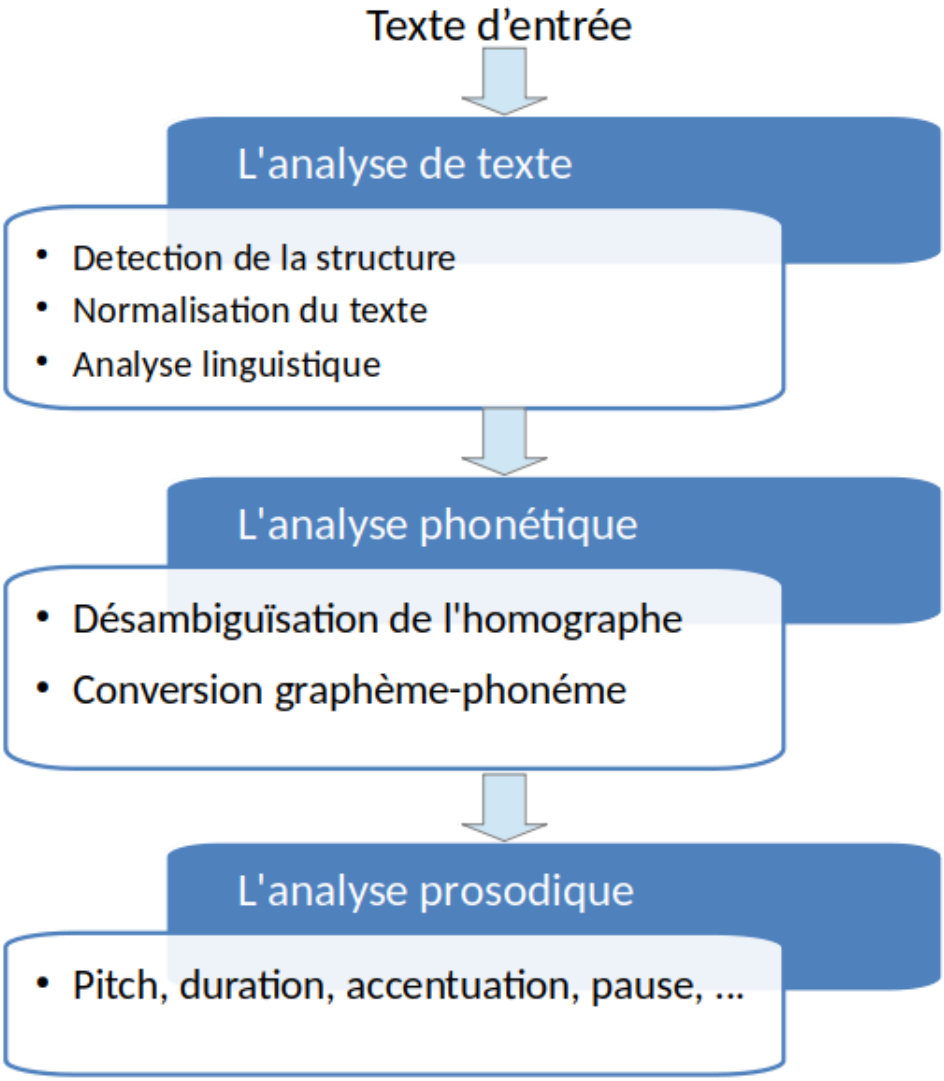


FIGURE 1.2 : Les tâches du front-end.[3]

1.2.1.1 L'analyse de texte

L'analyse de texte est un processus dépendant de la langue. Selon la description de Bensesty et al [3], cette phase comprend la détection de la structure du document, la normalisation du texte, et l'analyse linguistique.

a. La détection de la structure du document.

La détection de la structure du document comprend l'interprétation des signes de ponctuation, mais elle ne se limite pas à ce niveau, elle peut inclure aussi le filtrage des en-têtes de courrier électronique, et elle peut même prendre en compte le formatage des paragraphes.[3]

b. La normalisation du texte

De nombreux éléments du texte apparaissent de telle manière que leurs prononciation n'a pas de relation évidente avec leurs apparence, comme les abréviations,

les acronymes et les chiffres. La normalisation du texte gère ces éléments pour les convertir en texte indiquant comment ils doivent être prononcés [4]. Exemple «Mr.» pourrait être rendu comme monsieur, «Dr» comme docteur, «10» comme dix, ...

c. L'analyse linguistique

Elle comprend deux phases [1] :

- Une analyse morphologique qui décompose toutes les parties possibles (racine, préfixe et suffixe) pour proposer la prononciation correcte des mots en fonction de leur orthographe.
- Une analyse syntaxique qui organise le texte et détecte le type de chaque mot et phrase. Cela facilite l'accentuation et gère les ambiguïtés en déterminant le type de chaque phrase (déclarative, question ...).

1.2.1.2 L'analyse phonétique

Elle prend les résultats de l'analyse précédente comme données, afin de formuler la transcription phonétiques du texte à synthétisé.

L'analyse phonétique comprend la désambiguïsation de l'homographe afin de déterminer la prononciation selon la nature des mots (par exemple spécifier quand le mot "content" dans la phrase suivante : "je suis content qu'ils nous content cette histoire" est utilisé comme adjectif et quand il est utilisé comme verbe), et la conversion graphème-phonème pour transformer le texte en phonème.[3]

La conversion graphème-phonème comprend deux approches : la première est basée sur un dictionnaire et la deuxième sur des règles lettre-au-son.

- Le dictionnaire consiste à stocker un maximum de mots de la langue avec leurs prononciation (transcription phonétique). Cette méthode est rapide et précise, mais l'inconvénient est la possibilité d'échec, si un mot n'est pas disponible. Afin de surmonter ce problème, une liste de règles lettre-au-son peut être utilisée.
- Les règles lettre-au-son sont utilisées pour couvrir toutes les possibilités de la composition des lettres dans le texte selon les règles d'orthographe de la langue. Lors de l'utilisation de ces règles, le dictionnaire sera donc limité à un ensemble de mot dont la prononciation diffère de leur écriture. Par conséquent, cette approche peut compléter le manque de mots par des règles de conversion prédéfinies, afin d'obtenir la transcription phonétique correcte des mots. Cela nécessite une connaissance approfondie de la langue.

Les dictionnaires de prononciation et les règles lettre-au-son peuvent être adaptés en fonction du dialecte linguistique spécifique du locuteur, car la prononciation de certains phonèmes est différente d'un dialecte à l'autre.

1.2.1.3 L'analyse prosodique

Trouver les caractéristiques qui rendent le discours naturel, est probablement le processus le plus difficile pour la synthèse vocale.

L'analyse prosodique est la dernière tâche du traitement du langage naturel, elle détermine la progression de l'intonation, du débit de parole et du volume sonore à travers un énoncé, qui sont finalement représentés au niveau du phonème en tant que fréquence fondamentale, durée et amplitude[3].

L'intonation signifie comment le motif de hauteur ou la fréquence fondamentale change pendant la parole[2].

Générer une parole avec une bonne prosodie aide les auditeurs à mettre en évidence les parties importantes du discours, et à déterminer également le sens de certaines phrases qui peuvent être comprises de différentes manières selon la façon dont elles sont prononcées.

1.2.2 La synthétisation vocale (Back-end)

Toutes les informations extraites du front-end sont transmises au back-end pour la synthèse.

La phase de synthétisation consiste à générer la voix à partir de la transcription phonétique et ses informations prosodiques produites dans le traitement linguistique.

Ils existent quatre catégories différentes de génération de forme d'onde ou ce que l'on appelle les techniques de synthèse : synthèse articulatoire, synthèse par règles, synthèse par concaténation et synthèse statistique paramétrique. Ces techniques sont discutées en détail dans la section suivante.

1.3 Les techniques de la synthèse vocale

Cette section présente les différentes techniques utilisées dans la synthèse vocale (figure 1.3). La différence entre ces techniques réside dans la clarté et la natu-

ralité de la voix générée, ainsi que dans la puissance de calcul et les ressources nécessaires pour chacune.

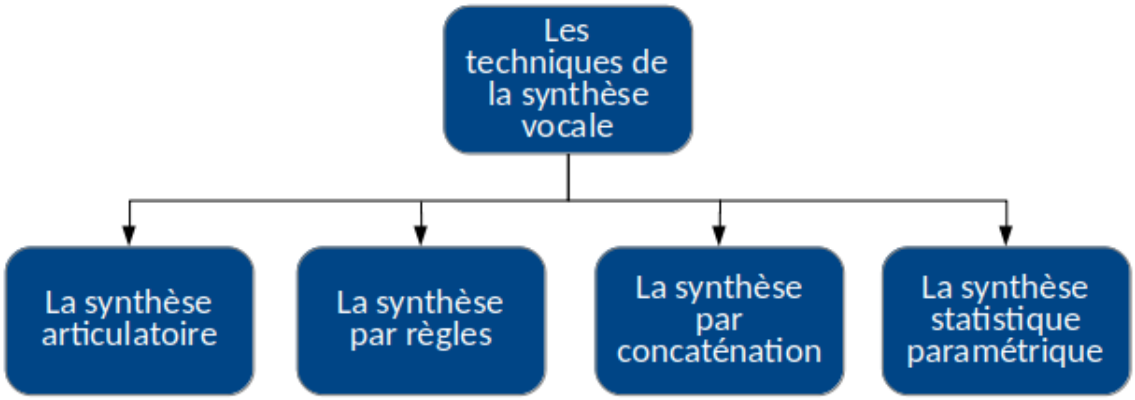


FIGURE 1.3 : Les techniques de la synthèse vocale.

1.3.1 La synthèse articulatoire

Essayer de simuler la production de la parole humaine est la technique la plus évidente pour synthétiser la parole, c'est la synthèse articulatoire. Cette approche est la plus ancienne en considérant la célèbre machine parlante de Von Kempelen (figure 1.4) comme un synthétiseur articulatoire. [5]

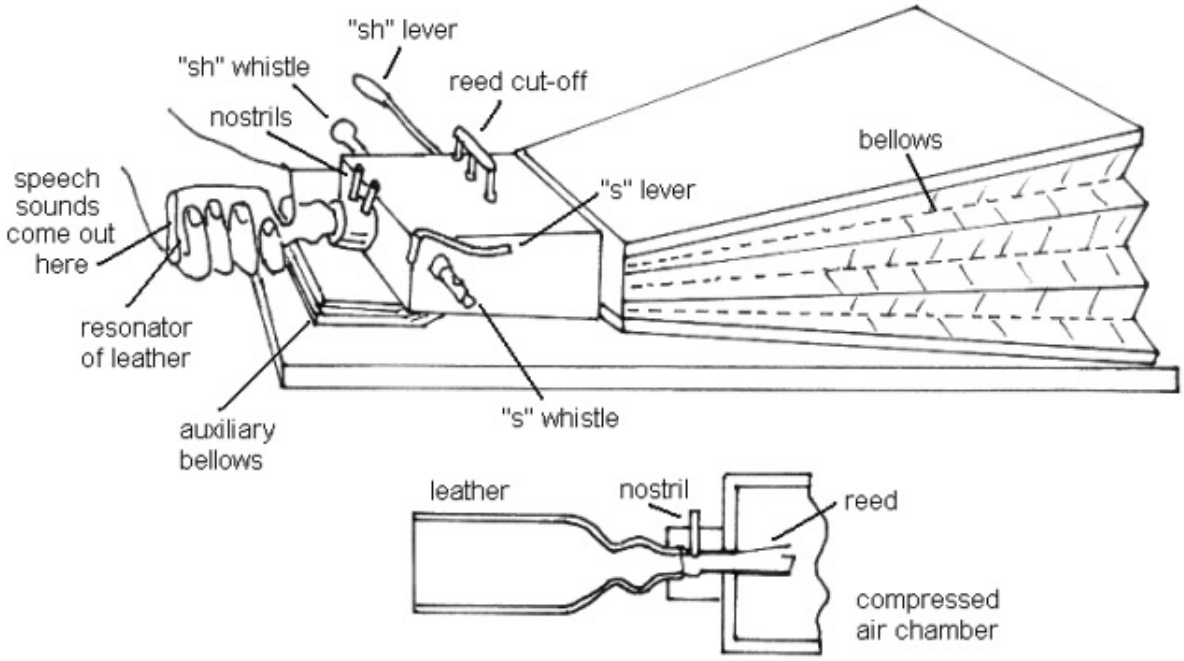


FIGURE 1.4 : Machine de Von Kempelen.

La synthèse articulatoire utilise des modèles mécaniques et acoustiques de production de la parole pour la synthétisation[3]. Elle tente de modéliser le mieux

possible les organes vocaux humains, de sorte qu'elle soit la méthode la plus satisfaisante pour produire une parole synthétique de haute qualité.

Les données du modèle articulatoire dérivent de l'analyse par rayons X de la parole naturelle. Cependant, ces données ne sont généralement que sur 2D, alors que le conduit vocal réel est naturellement à trois dimensions. En raison de ce manque de données sur les mouvements des articulateurs pendant la parole, la synthèse articulatoire est très difficile à optimiser. Par conséquent, cette approche a reçu moins d'attention et n'a pas encore atteint le même niveau de succès que d'autres méthodes de synthèse.[2]

1.3.2 La synthèse par règles (synthèse du formant)

C'était la première technique à être développée et la plus largement utilisée jusqu'au début des années 80. La synthèse du formant produit de la parole en générant des signaux basés sur des règles qui imitent le plus précisément possible la structure des formants et d'autres propriétés spectrales de la parole naturelle. Cependant, les techniques basées sur des règles produisent une voix artificielle robotique.[6]

Les paramètres d'entrée sont la fréquence fondamentale (F0), le quotient ouvert d'excitation (QO), le degré de sonorisation en excitation (VO), la fréquence et amplitudes de formants (F1 … F3 et A1 … A3), la fréquence d'un résonateur basse fréquence supplémentaire (FN) et l'intensité de la région à faible et haute fréquence (ALF, AHF).[2]

Ils existent deux structures de base, cascade (figure 1.5) qui est utilisée pour synthétiser les consonnes nasales, fricatives et occlusives, et parallèle (figure 1.6) qualifié comme une bonne structure pour synthétiser les consonnes voisées non-nasales. Mais pour de meilleures performances, et sachant que certains phonèmes ne peuvent pas être synthétisés par les deux structures, elles sont donc combinées afin de construire des structures plus efficaces.

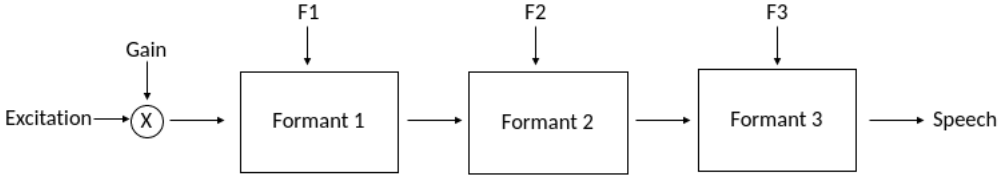


FIGURE 1.5 : Structure de base du synthétiseur de formants en cascade.[2]

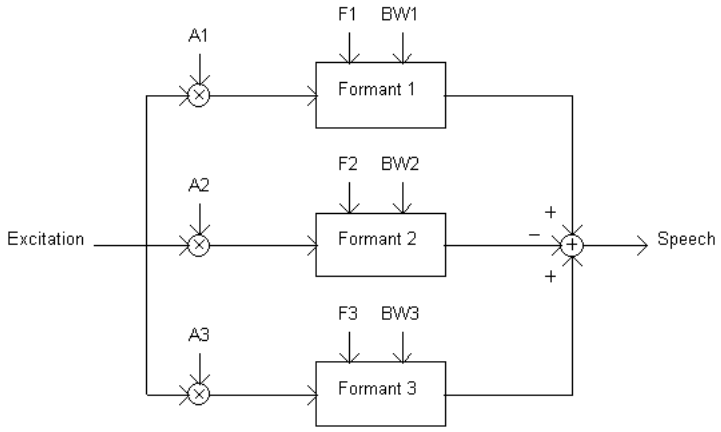


FIGURE 1.6 : Structure de base du synthétiseur de formants parallèle.[2]

La synthèse du formants est recommandée pour certaines applications telles que dans les appareils portables (comme les dictionnaires parlants), ou même dans les téléphones portables, car ses exigences de calcul sont modestes, par exemple les besoins en mémoire peuvent être aussi faibles que 1 Mo[3].

La synthèse par règle produit une voix hautement compréhensible (une vois intelligible) et fournit un nombre illimité de sons, ce qui la rend une approche puissante et la plus utilisée lorsque l'objectif est d'obtenir une parole intelligible même si elle manque de naturalité. Contrairement aux autres méthodes, l'inconvénient de la synthèse par règle est qu'elle produit une voix robotique.

1.3.3 La synthèse par concaténation

Lier des entités sonores naturelles préenregistrées semble être le moyen le plus simple pour produire une voix synthétique, compréhensible et naturelle.

La synthèse par concaténation utilise des unités de différentes longueurs préenregistrées à partir de la parole humaine et stockées dans une base de données. Les unités sont choisies en fonction de l'identité du phonème et d'autres critères tels que la prosodie, la position dans la phrase, la position dans le mot, etc. Ensuite, après avoir effectué des modifications acoustiques sur les segments individuels, elles sont concaténées pour produire l'énoncé souhaité[7].

La figure 1.7 montre le schéma fonctionnel d'un synthétiseur de concaténation.

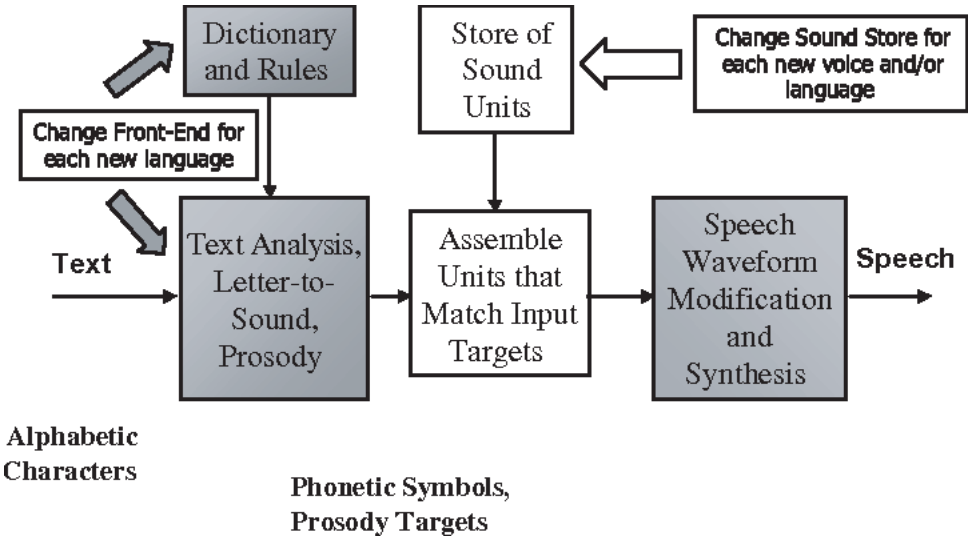


FIGURE 1.7 : Schéma fonctionnel d'un synthétiseur de concaténation.[3]

L'un des aspects les plus importants de la synthèse par concaténation est de choisir la bonne longueur de l'unité. Les unités longues sont plus naturelles avec moins de points de concaténation, mais elles consomment beaucoup de mémoire et le degré de flexibilité se dégrade avec des unités plus longues. Les unités courtes sont économiques en termes de mémoire, mais les procédures de collecte et d'étiquetage des échantillons sont complexes, et ils demandent beaucoup de temps. Généralement, les unités utilisées dans les systèmes actuels sont des mots, des syllabes, des demi-syllabes, des triphones, des diphones, et des phonèmes.[2]

La base de données utilisée par un synthétiseur de concaténation est créée en passant par trois étapes principales [2] :

1. L'enregistrement de la parole naturelle pour que toutes les unités seront utilisées dans le maximum de contextes possibles.
2. Les unités enregistrées doivent être étiquetées.
3. Choisir les unités les plus appropriées.

Les synthétiseurs par concaténation génèrent une voix intelligible, et donnent la possibilité de conserver la voix originale. Mais ils sont limités à une seule voix (un seul locuteur), et exigent une capacité de mémoire supérieure (une grande base de données).

1.3.4 La synthèse statistique paramétrique

Avec le développement de l'intelligence artificielle, de nouvelles méthodes ont été utilisées pour la synthétisation vocale, ce sont des méthodes statistiques pa-

ramétriques.

La synthèse statistique paramétrique comprend les étapes suivantes [8] :

- L'extraction des paramètres de la parole (spectraux et d'excitation par exemple) à partir d'une base de données de la parole naturelle (corpus de parole).
- La modélisation des paramètres ainsi extraits en utilisant un ensemble de modèles génératifs comme le modèle de Markov caché.
- Les paramètres du modèle sont estimés en utilisant le critère du maximum de vraisemblance.
- La forme d'onde est reconstruite à partir des représentations paramétriques pour la synthétisation.

Les techniques les plus populaires de l'approche statistique paramétrique sont les modèles de Markov cachés et les réseaux de neurones profonds, qui peuvent générer une voix compréhensible proche de celle de l'humain.

1.3.4.1 Synthèse basée sur le Modèle de Markov Caché (HMM)

Le modèle de Markov caché est une méthode statistique puissante pour caractériser les échantillons de données observées d'une série chronologique discrète d'états reliés par des transitions. Ce modèle introduit un processus non déterministe qui génère des données d'observation dans n'importe quel état. Il est devenu l'une des méthodes statistiques les plus puissantes pour modéliser les signaux vocaux.[9]

Les HMM ont été utilisés avec succès dans plusieurs domaines comme la reconnaissance automatique de la parole, la synthèse vocale et la traduction automatique.

La figure 1.8 montre un modèle de Markov chaché à 3 états (3-state left-to-right model).

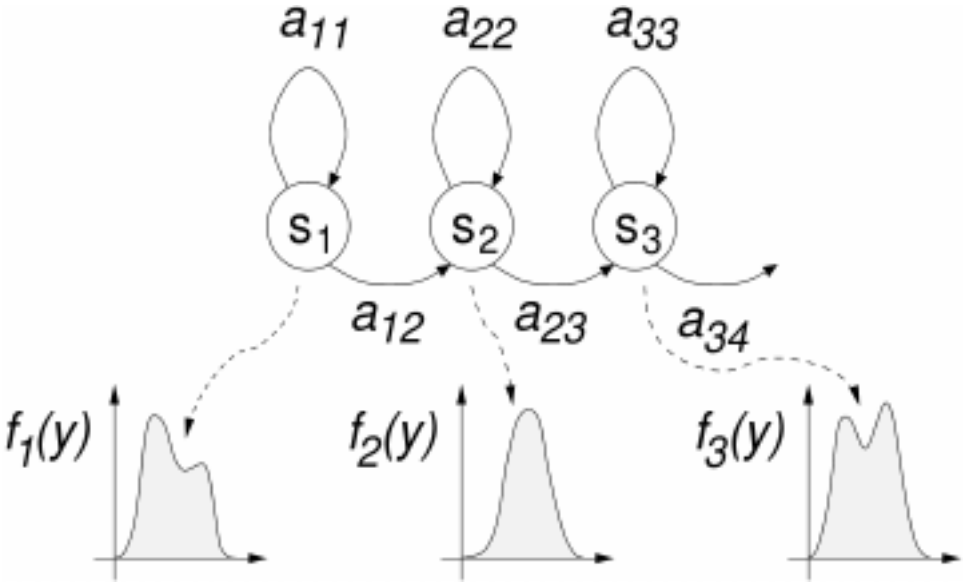


FIGURE 1.8 : Modèle de gauche à droite à trois états.[8]

Le processus de synthétisation nécessite la conversion du texte en une séquence de phonèmes et de fonctionnalités représentant le contexte (les phonèmes adjacents, syllabe, prosodie …). Sur la base de ces derniers, une séquence des HMM dépendants du contexte est choisie, afin de générer les paramètres vocaux correspondants. La parole est ensuite synthétisée à partir de cet ensemble de paramètres à l'aide d'un vocodeur. [7].

Cette approche présente divers avantages comme la flexibilité de modifier ses caractéristiques vocales, la douceur et l'intelligibilité. Sa principale limitation est qu'elle génère une voix avec un léger bourdonnement.

1.3.4.2 Synthèse basée sur les réseaux de neurones profonds (DNN)

Une solution alternative basée sur les réseaux de neurones profonds a été trouvée, afin de surmonter les limites de la synthèse basée sur les modèle de Markov cachés.

Cette technique est utilisée comme modèle de régression pour mapper des caractéristiques linguistiques d'entrée vers des caractéristiques acoustiques de sortie. En raison de cette transformation, ce type de synthèse est très coûteux en terme de calcul.[10]

La figure 1.9 montre un système de synthèse vocale basé sur les réseaux de neurones.

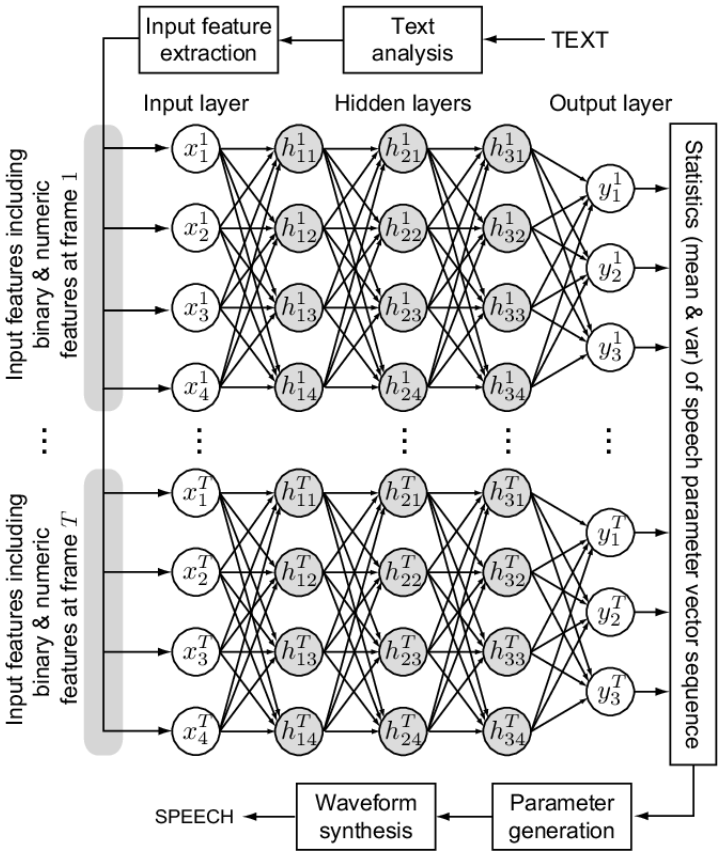


FIGURE 1.9 : Un système de synthèse vocale basé sur un DNN.[11]

Tout d'abord, l'énoncé à synthétiser passe par un analyseur de texte pour être converti en une séquence de caractéristiques d'entrée (séquence de phonème et de fonctionnalités). Les fonctionnalités comprennent :

- Des réponses binaires aux questions sur les contextes linguistiques (par exemple, la question suivante : le phonème actuel est-il 'b' ?).
- Des valeurs numériques telles que le nombre de mots dans la phrase, la position du phonème actuel et la durée du phonème actuel.

Par la suite, les caractéristiques d'entrée sont mappées aux caractéristiques de sortie par un réseau de neurones formé utilisant la propagation directe. Les caractéristiques de sortie comprennent les paramètres spectraux et d'excitation et leurs dérivées temporelles (caractéristiques dynamiques). Finalement, la parole est synthétisée à partir des caractéristiques générées.[11]

1.4 Étude comparative des techniques de la synthèse vocale

Comme il est mentionné ci-dessus, nous pouvons utiliser 4 techniques de synthèse :

1. La synthèse articulatoire : Elle utilise des modèles mécaniques et acoustiques pour modéliser le mieux possible les organes vocaux humains.

2. La synthèse par règles : Elle est basée sur un ensemble de règles utilisées pour déterminer les paramètres nécessaires, afin de synthétiser la parole désirée.

3. La synthèse par concaténation : Elle utilise des unités préenregistrées, de différentes longueurs à partir de la parole humaine, qui seront concaténées pour produire l'énoncé souhaité.

4. La synthèse statistique paramétrique : Les techniques les plus populaires de cette approche sont :

    - La technique basée sur le modèle de Markov caché (HMM). Elle utilise une base de données des HMM formés pour choisir une séquence dépendant du contexte, afin de générer des paramètres vocaux correspondants pour la synthétisation.
    - La technique basée sur les réseaux de neurones (DNN). Les caractéristiques d'entrée sont mappées par un réseau de neurones, afin de synthétiser la parole à partir des caractéristiques dynamiques générées en sortie.

La tableau 1.1 résume les avantages et les inconvénients de chaque technique :

TABLE 1.1 : Résumé des avantages et inconvénients des différentes techniques de la synthèse vocale

Les méthodes	Avantages	Inconvénients
Articulatoire	• Parole de haute qualité.	• Manque de données sur les mouvements des articulateurs humains.
Par règles	• Parole de haute qualité en terme d'intelligibilité. • Ne nécessite aucune base de données. • Nombre illimité de sons.	• Une voix robotique.
Par concaténation	• Parole de haute qualité en terme d'intelligibilité • Voix naturelle grâce à la possibilité de conserver la voix original	• Nécessite une grande base de données. • Limités à une seule voix.
Modèle de Markov Caché (HMM)	• Ne nécessite pas une grande base de données. • Voix compréhensible et naturelle.	• Audio légèrement bourdonné
Les réseaux de neurones (DNN)	• Modéliser les dépendances de contexte complexes. • Voix intelligible et proche de celle de l'humain.	• Coût de calcul élevé

1.5 Les applications des systèmes de synthèse vocale

Le domaine d'application de la parole synthétique se développe rapidement, et la qualité des systèmes TTS s'améliore régulièrement. Les systèmes de synthèse vocale deviennent également plus abordables pour les utilisateurs individuels.

Aujourd'hui, nous avons des systèmes de synthèse vocale qui produisent des voix intelligibles et naturelles. Ces synthétiseurs de haute qualité ont de nom-

breuses applications, comme les exemples suivants :

1.5.1 Interface vocale (IHM)

la synthèse vocale peut être utilisée pour diverses interactions homme-machine. Par exemple, dans les systèmes d'avertissement et d'alarme, elle est utilisée pour donner des informations plus précises sur la situation. Le synthétiseur vocal peut également être utilisé pour recevoir certaines notifications de bureau à partir d'un ordinateur, telles que l'activité de l'imprimante ou un courrier électronique reçu.

1.5.2 Accessibilité

Les applications les plus importantes et les plus utiles en synthèse de la parole sont celles qui aident les aveugles à lire et à communiquer. Avant la synthèse vocale, des livres audio spécifiques étaient utilisés, où leur contenu est lu sur une cassette audio. De toute évidence, faire de telles copies orales de livres prend du temps et coûte très cher.

Les personnes aveugles ou malvoyantes peuvent utiliser les systèmes de synthèse vocale pour un accès très facile aux informations écrites grâces à la lecture des écrans informatiques.

La parole synthétique offre aussi aux sourds et handicapées de la voix l'opportunité de communiquer avec des personnes qui ne comprennent pas la langue des signes. Elle peut offrir aux handicapés le moyen de s'intégrer dans le domaine technologique et de parler à travers la parole artificielle.

1.5.3 Applications éducatives

La parole synthétisée peut également être utilisée dans de nombreuses situations éducatives. Un ordinateur avec synthétiseur vocal peut être programmé pour des tâches spéciales, telles que l'orthographe et la prononciation pour différentes langues. Il peut également être utilisé avec des applications éducatives interactives.

1.6 Conclusion

Ce chapitre commence par présenter les éléments de système de synthèse vocale, qui sont principalement composés d'un module de traitement linguistique et d'un module de synthétisation.

Par la suite, les quatre principales techniques de synthèse vocale (articulatoire, par règles, par concaténation et statistique paramétrique) sont discutées. La méthode articulatoire fait référence aux techniques computationnelles basées sur des modèles de l'appareil vocale humain, et des processus d'articulation qui s'y produisent. La méthode par règles peut produire une parole assez intelligible, mais avec une voix robotique. Nous pouvons utiliser la méthode par concaténation pour résoudre le problème de naturalité et obtenir une voix proche de celle de l'humain. Mais comme il est difficile d'avoir une base de données qui couvre tous les contextes possibles de chaque phonème, une alternative est d'utiliser des techniques statistiques paramétriques où nous essayons d'apprendre des données au lieu de les concaténer.

Pour terminer ce chapitre, nous avons mentionné les diverses utilisations des systèmes de la synthèse vocale. Dans le chapitre suivant, nous entamerons l'étude de la langue arabe et de sa phonologie.

Chapitre 2

La phonologie de la langue arabe

2.1 Introduction

L'arabe est l'une des langues les plus utilisées dans le monde, mais elle est encore très tardive dans les domaines de la technologie et de l'informatique. C'est pourquoi il est nécessaire pour nos travaux de recherche d'aller dans des technologies nouvelles.

Dans le chapitre ci-dessus, nous avons introduit le principe des systèmes de synthèse vocale, et expliqué les différentes techniques utilisées dans ce domaine. La parole synthétique a un large éventail d'applications dans la vie quotidienne, qui touchent différents groupes de personnes dans de nombreux domaines.

Afin d'adopter cette technique en arabe, une étude attentive de la phonologie doit être effectuée, dans le but de créer une voix intelligible et naturelle qui lit n'importe quel texte et imite au mieux la voix humaine.

L'étude phonologique de l'arabe permet de bien comprendre cette langue, et analyse la manière dont les unités sonores appelées phonèmes se combinent pour fournir la parole.

Notre travail dans le domaine de la synthèse vocale est un exemple de plusieurs recherches scientifiques visant à supporter l'arabe dans divers domaines technologiques.

Dans ce chapitre, nous allons d'abord donner un bref aperçu sur l'arabe et son écriture. Ensuite, nous présenterons la phonétique arabe et la technique pour convertir un texte en phonèmes. Nous terminerons par mentionner le problèmes des diacritiques dans la synthétisation d'un texte arabe.

2.2 La langue arabe

L'arabe est une langue sémitique et l'une des langues les plus anciennes du monde. Puisqu'il s'agit de la langue d'enseignement religieux de l'Islam, plus de locuteurs ont au moins une connaissance de base de cette langue.

L'arabe a toujours été un sujet de recherche relativement limité, elle présente de nombreuses différences avec les autres langues, telles que des formes différentes pour la même lettre, une morphologie riche et productive, des propriétés phonétiques uniques et des signes diacritiques.[12]

2.3 L'écriture arabe

L'arabe s'écrit de droite à gauche. L'alphabet se compose de 29 lettres (tableau 2.1), dont 26 représentent des consonnes. Les 3 lettres restantes peuvent être des consonnes ou des voyelles longues " المد ". Les lettres peuvent apparaître sous différentes formes : au début, au milieu, à la fin d'un mot, ou d'une manière isolée. Le concept de lettres majuscules et minuscules n'existe pas.[13]

Le tableau 2.1 montre toutes les lettres de l'alphabet arabe et leurs correspondances avec les phonèmes en utilisant l'alphabet IPA (International Phonetic Alphabet).

Le tableau 2.2 montre des lettres arabes particulières et leurs correspondances avec les phonèmes en utilisant l'alphabet IPA.

L'alphabet phonétique international (IPA) est un alphabet utilisé pour la transcription phonétique des sons du langage parlé. Il est prévu pour couvrir l'ensemble des langues du monde. Il a été publié pour la première fois en 1888.[14]

Les signes diacritiques sont des petits symboles qui apparaissent avec les lettres. Ces signes ont un rôle dans la prononciation qu'il faut connaître.

TABLE 2.1 : Lettres arabes[7]

Lettre	Nom		PhonèmeIPA
ء	Hamza	همزة	/ ʔ/
ا	Alif	ألف	/ a : /
ب	Ba	باء	/ b /
ت	Ta	تاء	/ t /
ث	Tha	ثاء	/ ɵ/
ج	Jim	جيم	/ dʒ/
ح	Ha	حاء	/ ħ /
خ	Kha	خاء	/ x /
د	Dal	دال	/ d /
ذ	Dhal	ذال	/ ð /
ر	Ra	راء	/ r /
ز	Zay	زاي	/ z /
س	Sin	سين	/ s /
ش	Shin	شين	/ ɺ/
ص	Sad	صاد	/ $s^ʕ$ /
ض	Dad	ضاد	/ $d^ʕ$ /
ط	Ta	طاء	/ $t^ʕ$ /
ظ	Za	ظاد	/ $ð^ʕ$ /
ع	Ayn	عين	/ ʕ/
غ	Rhayn	غين	/ ɣ/
ف	Fa	فاء	/ f /
ق	Qaf	قاف	/ q /
ك	Kaf	كاف	/ k /
ل	Lem	لام	/ l /
م	Mim	ميم	/ m /
ن	Nun	نون	/ n /
ه	Ya	ياء	/ h /
و	Waw	واو	/ w / ou / u : /
ي	Ya	ياء	/ j / ou / i : /

TABLE 2.2 : Autres lettres arabes particulières[7]

Lettre	Nom		PhonèmeIPA
ة	Ta marbuta	تاء مربوطة	/ t / ou Ø
ى	Alif maqsura	ألف مقصورة	/ a : /
آ	Alif maddah	ألف ممدودة	/ a : /
أ	hamza	همزة فوق الألف	/ ʔ/
إ	hamza	همزة تحت الألف	/ ʔ/
ؤ	hamza	همزة فوق الواو	/ ʔ/
ئ	hamza	همزة فوق الياء	/ ʔ/

2.3.1 Les consonnes "الحروف الساكنة"

Dans la langue arabe, chaque consonne a un seul son qui la représente. En terme d'articulation, une consonne est le son produit lorsque le flux d'air est obstrué dans le tractus (canal) vocal.[15]

En regroupant les lettres qui ont le même son dans une seule consonne, l'alphabet arabe compte 28 consonnes. D'après les deux tableaux ci-dessus, et afin d'obtenir le nombre correcte de consonnes, nous devons analyser différents cas possibles :

- Les lettres "ئ ؤ إ أ ء" ont le même son /ʔ/ qui signifie la consonne "hamza".
- Si la lettre "ة" est suivie d'une voyelle elle se prononce /t/ comme la consonne "ت" , sinon elle est muette Ø.
- La lettre alif "ا " a un cas particulier où elle peut se prononcer /ʔ/. Exemple le mot : اِذْهَبْ

2.3.2 Les voyelles "حروف العِلَّة"

Une voyelle est le son produit avec un flux d'air relativement libre dans le tractus vocal. Les phonéticiens arabes ont reconnu six voyelles parmi lesquelles trois sont longues et les trois autres sont courtes.[16]

- Les voyelles courtes : Elles sont considérées comme une lettre de son, qui peuvent se prononcer seules (tableau 2.3).
- Les voyelles longues : C'est l'allongement des voyelles courtes à cause du «Mad», nous pouvons distinguer 3 formes de Mad (tableau 2.4).

TABLE 2.3 : Voyelles courtes

Signe	Nom		transcription	Exemple	
◌َ	الفتحة	Fatha	/ a /	بَ	/ ba /
◌ُ	الضمة	Damma	/ u /	بُ	/ bu /
◌ِ	الكسرة	Kasra	/ i /	بِ	/ bi /

TABLE 2.4 : Voyelles longues

Nom		transcription	Exemple	
مَدْ الفتحة	Mad Al Fatha	/ a : /	بَا	/ ba : /
الضمة	Damma	/ u : /	بُو	/ bu : /
الكسرة	Kasra	/ i : /	بِي	/ bi : /

Dans la langue arabe, les voyelles courtes sont des signes diacritiques liés aux lettres afin de se préserver toute ambiguïté d'interprétation.
Les voyelles courtes ne peuvent pas apparaître en début de mot, elles peuvent apparaître entre deux consonnes ou en position finale du mot.

2.3.3 Le signe de diacritique "Sukun السكون"

C'est un petit rond apposé sur une consonne lorsqu'elle n'est liée à aucune voyelle "حرف ساكن". Exemple la lettre (ص) dans le mot (خَصْمٌ).

Ce signe de diacritiques est éliminé du système vocalique car il n'a pas de son.

FIGURE 2.1 : Signe de diacritique "السكون"

2.3.4 Les diacritiques doubles "التَّنوين"

Le signe de diacritique double est ajouté à la fin des mots indéterminés. Il est en relation d'exclusion avec l'article de détermination "ال " placé en début de mot. Ils existent trois symboles de Tanwin. Dans l'écriture, ils sont le doublement de signes qui présentent les voyelles courtes, mais en réalité ils n'ajoutent que le son de la lettre "ن / n /" après.

TABLE 2.5 : Diacritiques doubles

Signe	Nom		transcription	Exemple	
◌ً	تنوين الفتحة	Tanwin Al Fatha	/a n/	لاً	/lan/
◌ٌ	تنوين الضمة	Tanwin Al Damma	/u n/	لٌ	/lun/
◌ٍ	تنوين الكسرة	Tanwin Al Kasra	/i n/	لٍ	/lin/

2.3.5 La gémination "الشَّدة"

Dans la langue arabe, si deux lettres identiques se suivent directement dans le même mot, le premier n'est liée à aucune voyelle (ساكن) et le deuxième est liée à une voyelle (متحرك), alors ils sont écrit dans une seule lettre avec un signe spéciale qui s'appelle (الشَّدة).[17] Exemple : رَكَّبَ se prononce /rakkaba/.

FIGURE 2.2 : Gémination

2.3.6 La ponctuation

La ponctuation facilite la prononciation et la compréhension des textes, en gardant le sens des phrases pour une lecture intelligible. Elle organise l'écriture et sépare les mots, les sous-phrases, les phrases et les paragraphes dans le texte. Elle est définie par des signes bien précis qui ne sont pas des lettres prononçables. Chacun de ces signes a un rôle important dans le texte.

Les signes de la langue arabe ont la même signification que les signes latins. Dans certains cas leurs formes changent pour s'adapter à l'écriture de droite à gauche.
Exemple :

- Le point «.» :Indique la fin des paragraphes et des phrases complets.
- La virgule « ، » : Sépare les phrases successives qui ont un sens lié.
- Le point-virgule « ؛ » : Sépare les phrases longues, et il sépare l'idée de son explication.
- Le point d'interrogation « ؟ » : Il est placé à la fin des phrases interrogatives.

- Le point d'exclamation « ! » : Il est placé à la fin des phrases exclamatives.
- Les deux points « : » : Ils sont mis pour citer des exemples ou avant de citer la parole de quelqu'un.

2.4 La phonétique arabe "الصوتيات العربية"

La phonétique est l'étude des sons de la parole dans le langage humain, et comment ils sont articulés par le tractus vocal humain.

La phonétique arabe se concentre sur les phonèmes spécifique à cette langue, en les divisant en deux groupes de base : les consonnes et les voyelles. Cette langue est marquée par un système vocalique limité et un système consonantique riche.[18]

Un phonème est la plus petite unité sonore qui correspond à un élément de la parole humaine pouvant indiquer des différences de sens entre des mots ou des phrases.[12]

L'arabe a 34 phonèmes composés de :

- Trois voyelles courtes.
- Trois voyelles longues.
- 28 consonnes.

Les consonnes et les voyelles peuvent être décrites en termes d'un certain nombre de caractéristiques phonétiques individuelles.

2.4.1 Les caractéristiques phonétiques des consonnes

Les sons des consonnes sont décrits selon trois propriétés phonétiques principales (tableau 2.6) : [15][12]

Lieu d'articulation : C'est un paramètre principal dans la description des consonnes. Il se réfère au point exact du tractus vocal, où se trouve un obstacle au passage de l'air.[19]

- Les consonnes glottales : Elles sont produites au niveau de la glotte. Ils existent deux consonnes glottales :

  ه / h / : هدهد

ء / ʔ/ : إمام

- Les consonnes bilabiales : Elles sont produites avec les deux lèvres. Ils existent trois bilabiaux :

  م /m/ : مكتبة

  و /w/ : وسيط

  ب /b/ : بقرة

- Les consonnes alvéolaires : Elles sont produites lorsque la partie antérieure de la langue est en contact avec la crête alvéolaire. Ils existent dix consonnes alvéolaires :

  ض / $d^ʕ$ / : ضابط

  س / s / : سائق

  ص / $s^ʕ$ / : صندوق

  ت / t / : تفاحة

  ط / $t^ʕ$ / : طماطم

  د / d / : دلو

  ن / n / : نوم

  ر / r / : رسام

  ز /z/ : زميل

  ل / l / : لافتة

- Les consonnes alvéolo-palatales : Elles sont produites lorsque la partie antérieure de la langue touche la crête alvéolaire puis le palais dur. Ils existent deux consonnes alvéolo-palatales :

  ج / dʒ/ : جمل

  ش / ʃ / :شجرة

- Les consonnes palatales : Elles sont produites lorsque la partie antérieure de la langue s'élève vers le palais. Il existe une seule consonne palatale :

  ي / j / : ياسمين

- Les consonnes labio-dentales : Elles sont produites avec les dents supérieures et la lèvre inférieure. Il existe une seule consonne de ce type :

  ف / f / : فهد.

- Les consonnes dentales : Elles sont produites avec la pointe de la langue entre les dents supérieures et inférieures. Ils existent trois consonnes de ce types :

  ث / ɵ/ : ثلاثة

  ذ / ð / : ذبابة

  ظ / ðˤ / : ظبي

- Les consonnes vélaires : Elles sont produites au niveau de l'arrière du palais. Ils existent trois consonnes vélaires :

  غ / ɣ/ : غراب

  ك / k / : كهف

  خ / x / : خصم

- Les consonnes uvulaires : Elles sont produites en soulevant l'arrière de la langue vers la luette. Il existe une seule consonne uvulaires :

  ق / q / : قريب

- Les consonnes pharyngales : Elles sont produites au niveau du pharynx. Ils existent deux pharyngales :

  ح / ħ / : حلم

  ع / ʕ/ : عاصمة

Mode d'articulation : Il se réfère aux mouvements des organes vocaux pour articuler un son, et traite la façon dont le flux d'air se déplace dans la bouche.[19]

- Les consonnes occlusives : Elles sont produites par une obstruction complète du flux d'air dans la bouche.

- Les consonnes fricatives : Elles sont produites par une obstruction partielle du flux d'air, où le passage dans la bouche par lequel l'air s'échappe est très étroit, provoquant des frottements.

- Les consonnes affriquées : Elles sont produites par une fermeture où le flux d'air est bloqué, suivie par une étape fricative où l'air retenu est relâché pour passer.

- Les consonnes nasales : Elles sont produites en abaissant le voile du palais.

- Les consonnes roulées : Elles sont produites par des vibrations entre la partie avant de la langue et la crête alvéolaire.

- Les consonnes liquides : Lors de la production de ces sons, il y a une certaine obstruction du flux d'air dans la bouche, mais pas assez pour provoquer une constriction ou une friction réelle.
- Les consonnes glides (semi-voyelles ou semi-consonne) : Elles sont produites avec peu ou pas d'obstruction de l'air dans la bouche. Lorsqu'elles se produisent dans un mot, elles doivent toujours être suivies ou précédées d'une voyelle, et dans leur articulation la langue se déplace rapidement de manière glissante vers ou loin d'une voyelle voisine.

Voisement : Ils existent deux types de consonnes, voisées et non voisées (sourdes). C'est l'une des principales caractéristiques qui différencient les consonnes, elle dépend du fonctionnement des cordes vocales, avec ou sans vibration.[19]

TABLE 2.6 : Caractéristiques phonétiques des consonnes arabes

		Bilabial	Alvéolaire	Alvéolo-Palatale	Palatale	Labio-Dentale	Dentale	Vélaire	Uvulaire	Pharyngale	Glottale
Occlusive	voisée	ب /b/	ض /dˤ/ د /d/								
	sourde		ط /tˤ/ ت /t/					ك /k/	ق /q/		ء /ʔ/
Fricative	voisée		ز /z/				ظ /ðˤ/ ذ /ð/	غ /ɣ/		ع /ʕ/	
	sourde		ص /sˤ/ س /s/	ش /ʃ/		ف /f/	ث /θ/	خ /x/		ح /ħ/	ه /h/
Affriquée	voisée			ج /dʒ/							
Nasale	voisé	م /m/	ن /n/								
Roulée	voisée		ر /r/								
Semi- voyelle	voisée	و /w/			ي /j/						
Liquide	voisée		ل /l/								

2.4.2 Les caractéristiques phonétiques des voyelles

Pour distinguer les différentes voyelles, nous nous appuyons sur trois propriétés phonétiques (Figure 2.3) :[16]

Hauteur de la langue : Trois degrés de hauteur de la langue sont reconnus :

- Voyelle haute : Position au plus près du palais.
- Voyelle basse : Position aussi éloignée que possible du palais.

- Voyelle moyenne : Position médiane entre une voyelle basse et une voyelle haute.

Quelle partie de la langue est impliquée : Concernant la position de la langue dans la production des voyelles arabes, trois places sont reconnues :

- Voyelle antérieure (avant) : Position le plus en avant possible.
- Voyelle postérieure (arrière) : Position le plus en arrière possible.
- Voyelle centrale : Position médiane entre une voyelle basse et une voyelle haute.

Arrondissement des lèvres : Sur la base de la position des lèvres, les voyelles arabes sont classées en voyelles non arrondies et arrondies.

- Voyelle arrondi : Se prononce avec les lèvres formant une ouverture circulaire.
- Voyelle non arrondi : Se prononce avec les lèvres relâchées.

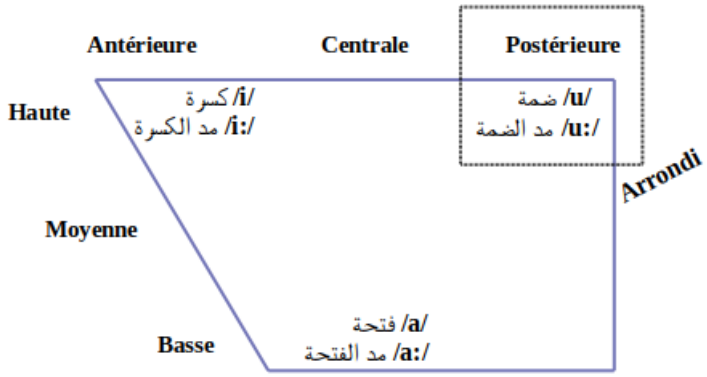


FIGURE 2.3 : Caractéristiques phonétiques des voyelles arabes

2.5 Les règles de prononciation

Dans ce projet, notre travail consiste toujours à créer une voix arabe synthétique. La vocalisation (التشكيل) seule ne permet pas de prédire la prononciation réelle. Nous devons donc décrire un ensemble de règles basées sur la phonologie arabe qui étendront un mot vocalisé à un ensemble de prononciations possibles.

Il faut noter que même les locuteurs formés à l'arabe, peuvent ne pas suivre la prononciation correcte selon la syntaxe et la phonologie arabes. Nous essayons donc d'intégrer ces variantes en utilisant des règles de prononciation.

Les règles suivantes sont appliquées sur chaque mot vocalisé :

- "ألف ممدودة " Alif-mamdouda : [آ]→ /ʔa :/
  Ex : (الآن)

- "التنوين " Tanwin : [AF] → /a n/, [F]→ /a n/, [K]→ /i n/, [N]→ /u n/
  Tel que A est la lettre (ا) et F,K,N sont les diacritiques doubles de fatha, kasra et dama respectivement.

- "الهمزة " Hamza : [أ , إ , ؤ , ئ , ء] → /ʔ/

- "التاء المربوطة" Ta-Marbota : [ة]+voyelle→ /t/, [ة] #→ ∅
  Tel que le symbole # signifie la fin du mot.

- "ألف مقصورة" Alif-Maqsura : ى → /a :/
  Ex : (على)

- "الشَّدة "Shadda : [s+~]→/ss/
  Tel que (~) est le signe de shadda, et (ss) est la transcription qui signifie un doublement de la lettre (س). Cette règle est appliquée sur toutes les lettres.
  Ex : (السَّيف)

- "المَّد" Voyelles langue : [aA]→ /a :/, [uw]→ /u :/, [iy]→ /i :/
  Tel que A, w, y signifie les lettres (ا , و , ي) respectivement.

- "واو الجماعة" Waw Al-jamaa : [uwoA]→ /u :/
  Tel que (o) est le signe qui signifie le diacritique sukun (السكون).
  Ex : (كتبُوا)

- "ال الشمسية" Al-chamsiya : [Al$]→ /ʔa ɬ/
  Tel que $ signifie la lettre (ش). Cette règle est appliquée sur toutes les lettres Al-chamsiya.
  Ex : (الشَّمس)

- "ال القمرية" Al-qamariya : [Alom]→ /ʔa m/
  Tel que m signifie la lettre (م). Cette règle est appliquée sur toutes les lettres Al-qamariya.
  Ex : (الموز)

- "التفخيم" Tafkhim : [a,u,i] ص ض ط ظ ق → /Â, Û, Î/
  Une voyelle (a,u,i) suivie d'une lettre dite "مفخمة" telle que (ص ض ط ظ ق) se prononce différemment d'une voyelle suivie d'une lettre normale.

- "الحروف الأجنبية" Lettres étrangères : ق → /q/, ڤ → /f/, پ → /b/
  Tel que p, v, g sont l'équivalent des lettres ق ڤ پ en arabe.

- "همزة الوصل" Hamzat Al-wasl : [ٱ] → /ʔ/, [ٱ] →Ø
  La lettre ٱ se prononce comme hamza si elle se trouve au début d'un mot, sinon elle est muette.

Les locuteurs arabes ignorent quelques règles de prononciation dans le discours quotidien comme l'arrêt d'une phrase sur le diacritique sukun "السكون",où la dernière lettre du dernier mot dans la phrase doit se prononcer sans la suivre d'une voyelle.

Ils existent d'autres règles de prononciation qui consistent à définir la transcription phonétique d'un mot, selon sa position dans la phrase, et selon les lettres d'un autre mot situé avant ou après.

2.6 Le problème des diacritiques

Lire un texte arabe non vocalisé (sans signes diacritiques) peut mener à changer sa vrai signification, surtout pour un locuteur qui n'a pas une connaissance suffisante de la langue. Voici un exemple de mot "كتب" :

TABLE 2.7 : Différentes prononciations d'un mot selon sa vocalisation.

Vocalisation	Pronciations	signification
كَتَبَ	/kataba/	a écrit
كُتِبَ	/kutiba/	a été écrit
كُتُبٌ	/kutubun/	livres

Il est difficile pour la machine de trouver la prononciation correcte des mots ayants la même écriture, c'est l'un des défis auxquels l'arabe est confrontée dans le domaine de la synthèse vocale.

Pour résoudre ce problème, les systèmes de synthèse vocale pour l'arabe utilisent une application tierce telle que «Mishkal[1]», pour vocaliser le texte avant de le prononcer.

2.7 Conclusion

Dans ce chapitre, nous avons présenté la phonologie de la langue arabe, en définissant sont écriture et ses différents composants et propriétés linguistiques.

[1]http ://tahadz.com/mishkal

Ensuite, nous avons introduit la phonétique arabe et les caractéristiques phonétiques des consonnes et des voyelles, puis discuté sur le problème des diacritiques, en terminant par définir les règles de prononciation.

Le chapitre suivant aborde différents systèmes de synthèse vocale open-source, en examinant leurs architectures et caractéristiques, afin de choisir un système approprié pour la réalisation de ce projet.

Chapitre 3

Les systèmes de synthèse vocale Open Source

3.1 Introduction

Nous avons présenté dans les deux premiers chapitres le concept de la synthèse vocale et la phonologie arabe. Pour créer une voix arabe synthétique, nous avons besoin d'un système de synthèse vocale open source.

La majorité des systèmes de synthèse vocale existants utilisent la synthèse par règles ou la synthèse de concaténation. Comme nous l'avons vu dans le chapitre 1, la synthèse par règles produit une voix robotique, tandis que la méthode de concaténation est largement utilisée à cause de la voix de haute qualité qu'elle produit en utilisant une grande base de données de sons. Les méthodes paramétriques ne sont toujours pas assez utilisées même si elles produisent une voix intelligible et naturelle car c'est une nouvelle technologie.

Même si la synthèse vocale pour la langue arabe est encore à ses débuts, comparativement à d'autres langues comme l'anglais, de nombreux développeurs ont travaillé sur l'adaptation de certains systèmes de synthèse vocale open-source pour la langue arabe

Dans ce chapitre, nous discuterons des principaux systèmes TTS open source existants, et de l'ajout de l'arabe à ces systèmes, et nous examinerons leurs architectures et caractéristiques, en terminant avec une comparaison entre ses systèmes.

3.2 Les systèmes de synthèse open-source

Ils existent de nombreux systèmes open-source pour la synthèse vocale. Dans cette section, nous passons en revue quatre systèmes connus, nous définissons chacun d'eux et nous voyons comment ils fonctionnent et quel travail a déjà été fait avec chaque système pour l'adaptation de la langue arabe.

3.2.1 ESpeak

ESpeak est un système de synthèse vocal open source compact pour Linux et Windows, adapté à l'anglais et à d'autres langues. Il utilise la méthode de synthèse par règles (synthèse du formant) qui permet à de nombreuses langues d'être intégrées, et dont les fichiers sont de petites tailles. La voix générée est claire et peut être utilisée à des vitesses élevées, mais il n'est pas naturel.[20]

3.2.1.1 Les caractéristiques de eSpeak

Les principales caractéristiques de eSpeak sont les suivantes[20][21] :

- Il comprend différentes voix, dont les caractéristiques peuvent être modifiées.
- Il peut produire une sortie vocale ou un fichier wav.
- Petite taille. Le programme et ses données (y compris plusieurs langues) totalisent environ 1,4 Mo.
- La possibilité d'ajouter d'autres langues. Plusieurs sont incluses à différents stades d'avancement.
- Il est écrit en C, donc c'est rapide..

3.2.1.2 Architecture et fonctionnement pour la langue Arabe

Selon T. Zerrouki et al [6], le processus commence avec une entrée (texte arabe) non vocalisée. Après sa vocalisation, le texte sera transféré dans le processus linguistique afin de générer sa représentation phonétique qui sera utilisée pour générer la parole comme le montrent les figures 3.1 et 3.2

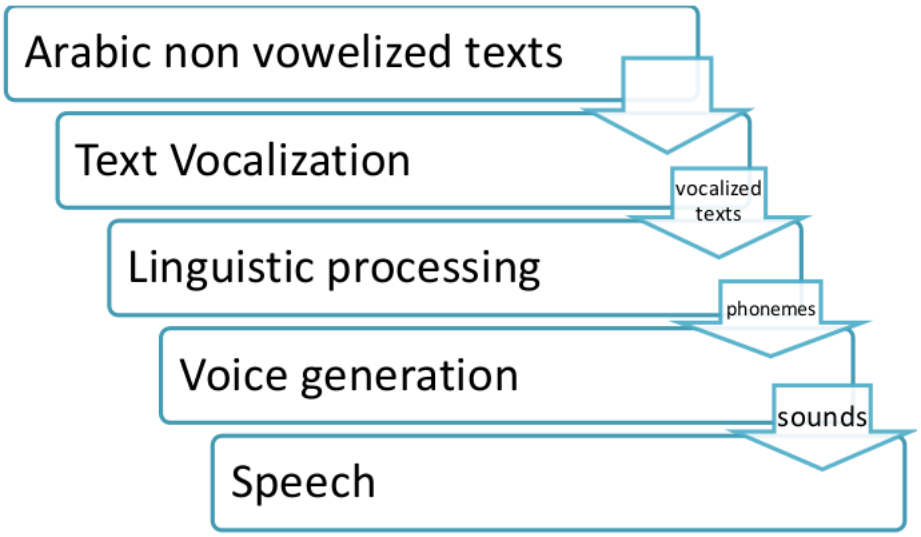


FIGURE 3.1 : Étapes de la synthétisation de eSpeak [6]

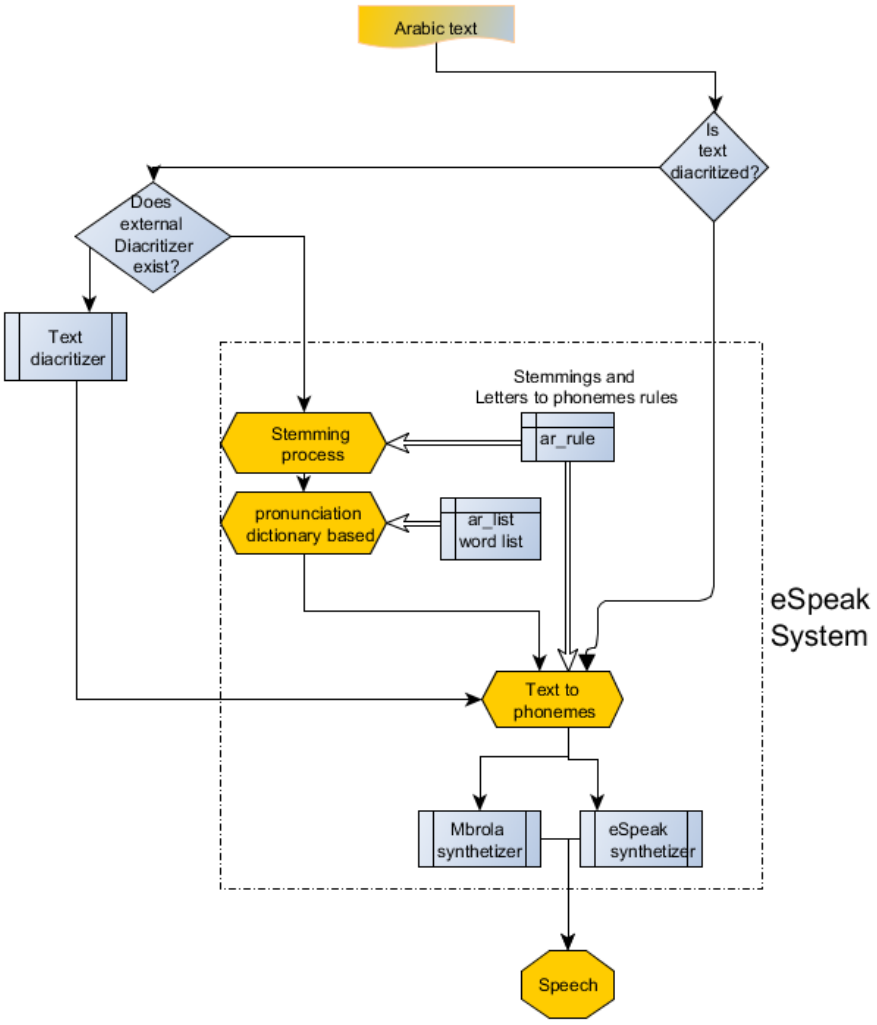


FIGURE 3.2 : Organigramme de eSpeak arabe [6]

3.2.2 Festival TTS

Le système de synthèse vocale du Festival a été développé à la CSTR (The Center for Speech Technology Research) de l'Université d'Edinburgh à la fin des années 90 par Alan Black, Paul Taylor et Richard Caley. Ce système TTS adopte la technique de synthèse vocale par concaténation, qui prend des diphones comme unité vocale de base[4][22].

Festival TTS utilise plusieurs méthodes pour arriver à synthétiser la voix comme la méthode de 'clustergen' ou 'unit selection', et il peut même utiliser des modèles entraînés par le système HTS qui est un outil de génération des modèles de Markov cachés pour la synthèse vocale.

3.2.2.1 Utilisation

Festival peut être utilisé à plusieurs niveaux[22] :

1. Les utilisateurs simples : Pour traduire le texte en parole, l'utilisateur entre le texte à traiter pour obtenir la parole en sortie.

2. Les développeurs des applications qui nécessitent la synthèse vocale : Les développeurs peuvent intégrer des composants de Festival dans leurs projets.

3. Les chercheurs dans le domaine de la synthèse vocale : La réalisation d'un système de synthèse vocale tout entier est une tâche difficile. Tous les modules de Festival sont open source et les chercheurs peuvent les utiliser pour tester de nouveaux algorithmes.

Le système Festival est implémenté en C++, mais afin de fournir des paramètres et de spécifier le flux de contrôle, Festival propose également un langage de script basé sur le langage de programmation Scheme. Scheme a une syntaxe très simple, mais il est à la fois puissant pour spécifier des paramètres et des fonctions simples qui n'augmentent pas la taille du système.[23]

3.2.2.2 Créer une nouvelle voix

Black et Lenzo[24] proposent une liste de processus de base permettant de créer une nouvelle voix dans Festival. Les processus les plus importants sont :

- Déterminer un ensemble de phonèmes avec leurs caractéristiques linguistiques (points d'articulation, type de voyelles, ...)

- Créer un lexique et des règles de lettre-au-son. Pour cela, Black et Lenzo [25] ont mis au point des techniques pour faciliter la construction de nouveaux lexiques. Pour les langues où la relation entre l'orthographe et la phonétique est très proche, comme la langue arabe et espagnole, il est possible d'utiliser un ensemble de règles lettre-au-son écrites à la main. Pour les autres langues, où la prononciation diffère de l'orthographe, ils existent des techniques d'apprentissage automatiques dans lesquelles les règles sont construites à partir des mots existants avec des prononciations existantes.

- Construire des modèles prosodiques afin de rendre le discours naturel et de bien comprendre le sens.

3.2.2.3 La langue arabe pour Festival TTS

Il y a eu de nombreux travaux académiques sur l'ajout de la langue arabe au Festival TTS, comme ceux de M. Hamad, M. Hussain [26] et M. Assaf [4]. La seule implémentation réelle et disponible est celle de Abdullah Alrajeh[1], qui a réalisé une voix de bonne qualité et disponible sur "github" en open source. Malheureusement aucun article ou explication de comment le système a été implémenté n'est disponible.

3.2.3 Mary TTS

Mary (Modular Architecture for Research on speech sYnthesis) TTS est une plateforme de synthèse vocale multilingue open-source écrite en Java. Elle a été initialement développée comme un projet commun entre le laboratoire de technologie linguistique de DFKI "Centre allemand de recherche pour l'intelligence artificielle", et l'institut de Phonétique de l'université de Sarre. Cette plateforme utilise la synthèse par concaténation avec les diphones comme unité vocale de base. Depuis la version 5.2, Mary TTS supporte l'allemand, l'anglais britannique et américain, le français, l'italien, le suédois, le russe et le turc. Cette plateforme est livrée avec des boîtes à outils, afin de faciliter l'ajout de nouvelles langues.[27]

3.2.3.1 Architecture

La Figure 3.3 illustre l'architecture de traitement adoptée par le système Mary TTS pour la langue allemande. Pour les autres langues, l'architecture de traitement est similaire.[27]

[1]https ://github.com/asrajeh/arabic-tts

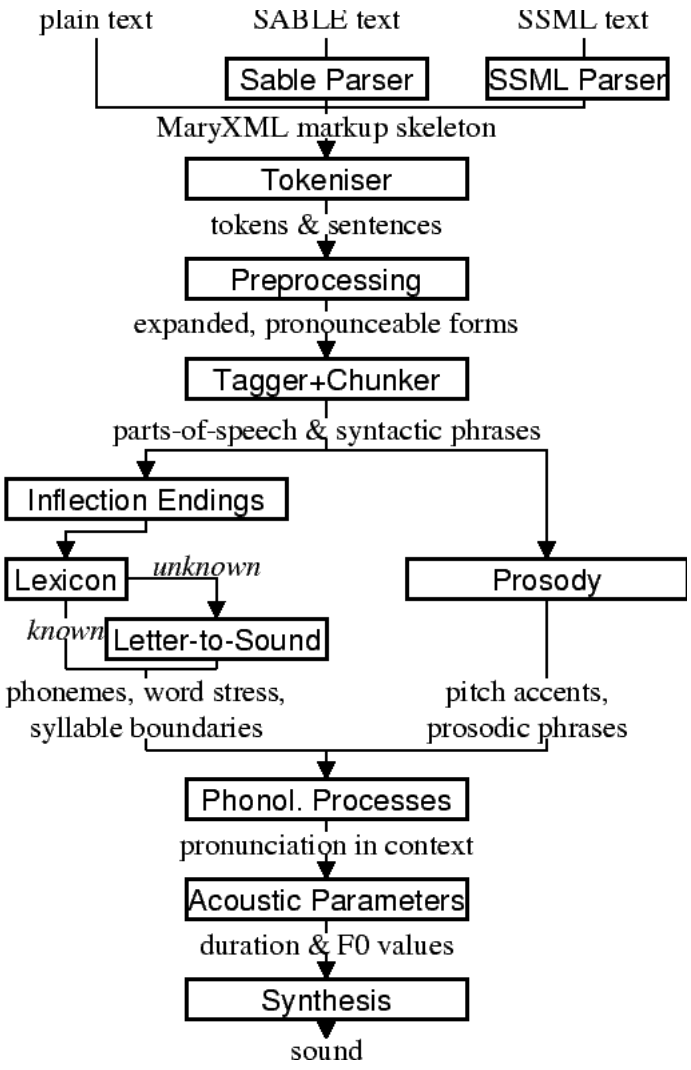


FIGURE 3.3 : Architecture de Mary TTS[27]

Le texte en entré passe par un module de tokenisation, afin de le décomposer en jetons. Ces derniers passent par un module de prétraitement, où les jetons dont la forme parlé ne correspond pas à la forme écrite seront remplacé par leurs formes prononçable. Après le prétraitement, Marry TTS associe à chaque segment ses informations grammaticales correspondantes (verbe, noms, genre ...). Tout les jetons passent par un module de prosodie afin d'extraire leurs informations prosodiques, et passent en parallèle par un lexique, qui est un dictionnaire de transformation du graphème vers phonèmes. Les mots qui ne se trouvent pas dans le dictionnaire passent par un module de lettre-au-son qui utilise un algorithme de conversion, afin de les convertir en phonèmes. La transcription phonétique résultante passe par un module de post-lexiques qui associe chaque phonème avec ses informations prosodiques et contextuelles correspondantes, afin qu'elles soient transformées en parole par le module de synthétisation. Plusieurs formats audio peuvent être générés (wav, mp3 ...).

3.2.3.2 La langue arabe pour Mary TTS

Rashad et al[28] ont travaillé sur un projet pour ajouter la langue arabe à Mary TTS en 2010. Ils ont réalisé deux tâches pour ajouter le support de l'arabe à Mary TTS [28] :

- La première consiste à construire un ensemble minimal de composants de traitement du langage naturel (nouveau modèle) pour la langue arabe. Dans cette tâche, une sorte de script est appliqué sur un corps volumineux de texte arabe, puis un lexique de prononciation doit être construit.
- La deuxième tâche est la création d'une voix arabe en utilisant l'outil d'enregistrement vocal 'redstart' pour que ces enregistrements soient concaténés.

Ce travail était académique mais le projet n'est pas supporter par Mary TTS et nous n'avons pas réussi à trouver le projet pour faire des testes.

3.2.4 Tacotron

En avril 2017, Google a publié un article[29], Tacotron : "Towards End-to-End Speech Synthesis". Dans cet article, un modèle neuronal de synthèse vocale est présenté, il apprend à synthétiser la parole directement à partir des enregistrements audios associés à leurs transcriptions textuelles. Cependant, Google n'a pas publié le code source de ce système ni les données d'entraînement utilisées. Keith Ito[2] en a fait une implémentation open-source.

Tacotron est une architecture de réseau neuronal pour la synthèse vocale. Le modèle est basé sur l'architecture de réseaux de neurones séquence à séquence (seq2seq). Il est composé d'un encodeur et d'un décodeur. L'encodeur convertit la phrase d'entrée en une représentation d'entités cachées que le décodeur transforme en spectrogramme.[30]

3.2.4.1 Séquence à séquence (seq2seq)

Seq2seq est une architecture de réseaux de neurones artificiels (ANN) qui intègre des réseaux de neurones récurrents. Elle a d'abord été inventée pour faire face à la tâche de la traduction automatique.[30]

[2]https ://keithito.com/

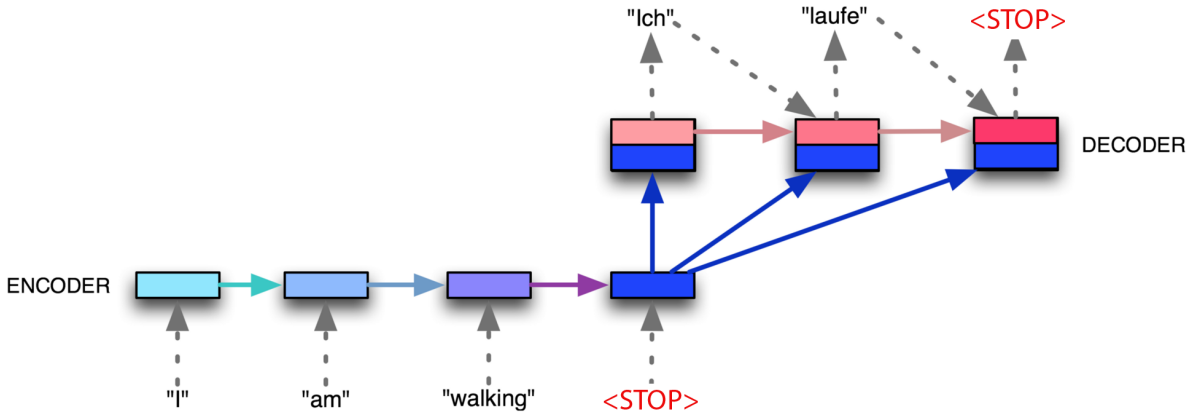


FIGURE 3.4 : Un réseau seq2seq simple effectuant la traduction automatique de l'anglais vers l'allemand[30]

3.2.4.2 Architecture

Tacotron est basé sur le modèle seq2seq. La figure 3.5 illustre le modèle, qui comprend un encodeur, un décodeur et un réseau de post-traitement. À un niveau élevé, ce modèle prend des caractères en entrée et produit des trames de spectrogrammes, qui sont ensuite converties en formes d'onde.[29]

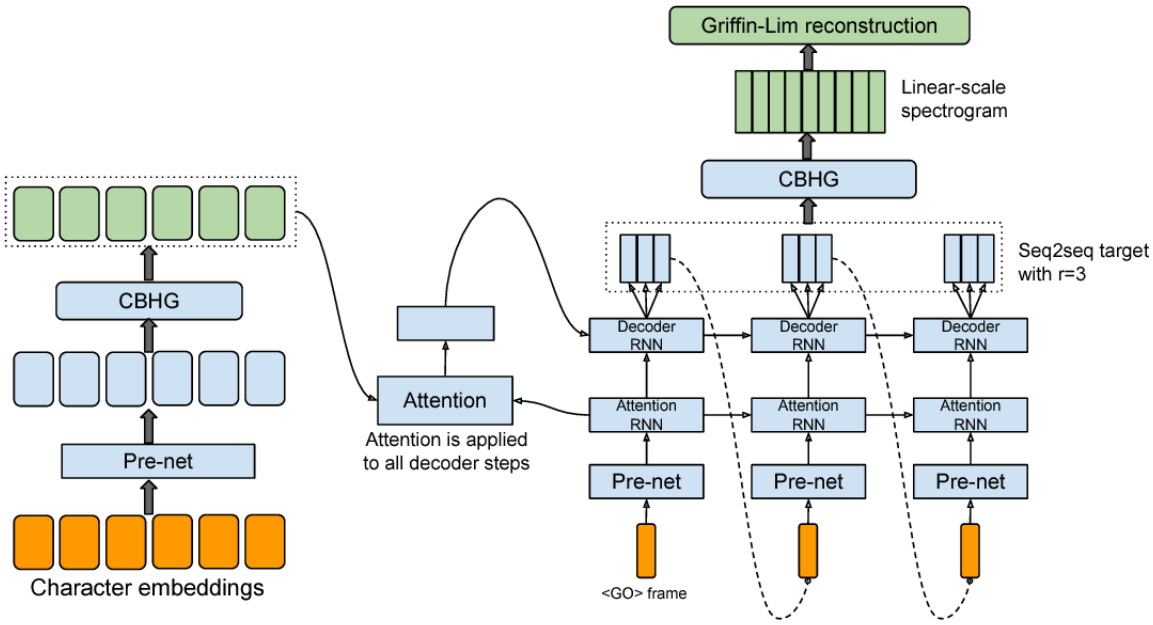


FIGURE 3.5 : Architecture du modèle.[29]

Module CBHG

CBHG (figure 3.6) est un type de réseaux de neurones artificiels (ANN) introduit dans l'article de Google sur Tacotron.[29]

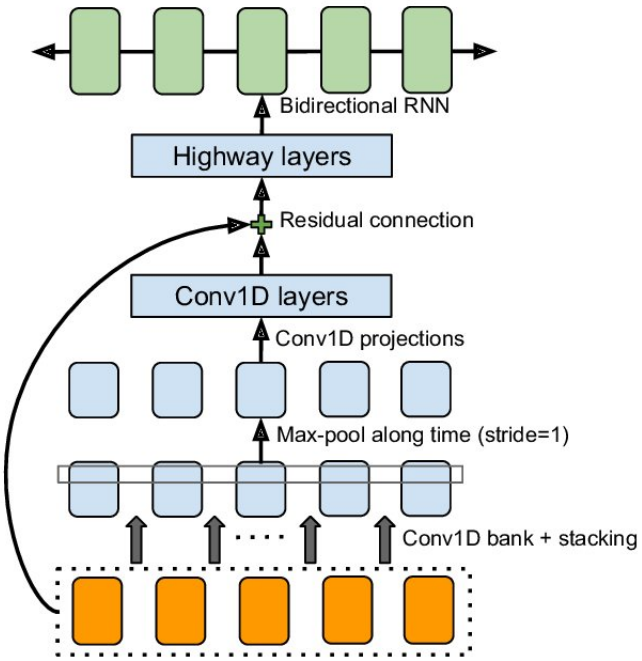


FIGURE 3.6 : Module CBHG.[29]

Il se compose d'un réseau convolutif, d'un réseau routier "highway layers" et d'un réseau de neurones récurrent (RNN) bidirectionnel. Le module CBHG est à la fois utilisé comme encodeur et comme réseau de post-traitement. Le réseau de post-traitement est chargé de convertir les spectrogrammes en échelle linéaire.[30]

Fonctionnement L'encodeur reçoit en entrée une séquence de caractères présentée comme un vecteur. Un ensemble de transformations non linéaires, appelées «pré-net», est appliqué pour transformer cette séquence en une représentation d'entités cachées qui passe par la suite par le décodeur. Le décodeur produit à partir de la représentation résultante de l'encodeur des trames de spectrogrammes qui seront converties par le réseau de post-traitement en échelle linéaire, et ensuite en parole.[29]

3.2.4.3 La langue arabe pour Tacotron

Youssef Sharief[3] a réalisé une version arabe open-source de Tacotron en se basant sur l'implémentation open-source faite par Keith Ito. Son modèle a été entraîné sur la base du corpus de Nawar Halabi[4].

[3]https ://youssefsharief.github.io/Arabic-Tacotron-Text-To-Speech/
[4]http ://ar.arabicspeechcorpus.com/

3.3 Comparaison

Nous avons étudié 4 systèmes open-source connus dans le monde de la synthèse vocale. Le tableau 3.1 résume les caractéristique de ces systèmes :

TABLE 3.1 : Les caractéristiques des systèmes de synthèse vocale

Système	Qualité de la voix générée	Technique de synthèse	Support de la langue arabe
ESpeak	Une voix robotique	Par règle	Officiellement à partir de la version 2.5.2
Festival TTS	Une voix de haute qualité	Synthèse par concaténation	Des initiatives open-source non supportées par le système officiel
Mary TTS	Une voix de haute qualité	Synthèse par concaténation	Des travaux académiques mais aucune implémentation n'existe réellement.
Tacotron	Une voix de haute qualité	Synthèse basée sur les réseaux de neurones	Une implémentation a été faite

Chacun de ces systèmes utilise une méthode ou une technique de synthèse vocale. Il y a ceux qui utilisent la méthode par règle (eSpeak), et certain la méthode par concaténation (Mary TTS, Festival TTS), et d'autres qui utilisent les réseaux de neurones (Tacotron). Festival TTS peut aussi être utilisé avec d'autres outils qui implémentent la technique de HMM.

La qualité de la voix généré diffère d'un système à l'autre selon la technique utilisée. ESpeak génère une voix robotiques contrairement aux autres systèmes qui génèrent des voix de haute qualité.

Des projets pour la langue arabe ont été faits sur les quatre systèmes mais seul eSpeak supporte la langue arabe d'une maniéré officiel à partir de sa version 2.5.2.

3.4 Conclusion

Dans ce chapitre, nous avons discuté de quatre principaux systèmes de synthèse vocale open-source et les travaux de la langue arabe faites sur ces systèmes. ESpeak a été présenté en premier, il utilise la technique de synthèse par règles pour générer des mots compréhensibles mais avec une voix robotique. Ensuite,

nous avons abordé les deux systèmes qui utilisent la technique de synthèse par concaténation, Mary TTS et Festival TTS. Nous avons terminé par le système Tacotron qui utilise une architecture basée sur les réseaux de neurones.

Dans la partie suivante, nous ferons notre étude conceptuelle et le choix de système, puis traiterons les différents processus de l'implémentation du projet, en terminant par tester la voix créée pour l'évaluer afin d'améliorer la qualité.

Deuxième partie

Conception et Implémentation

Chapitre 4

Conception

4.1 Introduction

Dans la partie ci-dessus, nous avons étudié et testé plusieurs systèmes et techniques de synthèse vocale, dans le but de choisir l'approche la plus convenable pour créer une voix arabe synthétique open-source proche de la voix humaine.

Cette partie décrit l'étude conceptuelle de notre projet. Nous allons commencer par justifier notre choix du système, puis montrer l'architecture générale et les différentes étapes abordées pour réaliser ce projet.

Deux chemins sont suivis pour le choix du système. Le premier est lié aux techniques utilisées, le deuxième est lié aux systèmes de synthèse vocal open-source que nous pouvons adapter pour la langue arabe.

L'architecture que nous allons montrer est le résultat de plusieurs études sur des systèmes similaires, et d'une étude plus large sur la création d'une voix anglaise pour bien comprendre le concept. Une fois l'architecture du système choisie, nous présenterons en détail les points abordés, afin de définir les étapes nécessaires pour l'implémentation du projet.

4.2 Choix du système

La figure 4.1 montre les étapes que nous avons suivies, afin de choisir une technique de synthèse vocale.

D'après l'étude comparative faite dans le chapitre 1, nous pouvons constater que la technique articulatoire n'est pas réalisable pour le moment à cause du

manque de données sur les mouvements des articulateurs humains. C'est pourquoi, nous l'avons retirée de nos choix, avec la technique par règle qui génère une voix robotique contrairement aux autres méthodes.

La méthode de concaténation génère une parole intelligible et naturelle, mais cela nécessite une quantité considérable de données phonétiques enregistrées pour les concaténer dans la synthétisation. Par contre, les méthodes basées sur DNN et HMM nécessitent un corpus de paroles naturelles pour l'entraînement mais ensuite ne nécessitent aucune base de données pour la synthétisation et donnent aussi des résultats intelligibles et naturelles.

Sachant que les Modèles de Markov Cachés ont un coût de calcul moins élevé que les DNN, et leurs résultats sont d'une qualité assez proche, nous avons décidé d'adopter une architecture basée sur les HMM.

Concernant les systèmes de synthèse vocale, ils existent 4 open-source qui sont déjà détaillés dans le chapitre 3, et nous pouvons les adapter pour la langue arabe.

1. Espeak : Est un synthétiseur vocal open source qui utilise la technique de synthèse par règles. Il permet à de nombreuses langues d'être intégrées, et dont les fichiers sont de petites tailles. Le discours est clair et peut être utilisé à des vitesses élevées, mais il n'est pas aussi naturel que les autres synthétiseurs basés sur des enregistrements de discours humains.

2. Festival : Est un système de synthèse par concaténation, il n'a pas besoin de ressources puissantes pour fonctionner. Il est largement utilisé dans divers appareils tels les ordinateurs, les lecteurs d'écran et même les smartphones. Festival est utilisé comme une plateforme de développement et de test, et il peut être intégré à d'autres projets qui nécessitent une sortie vocale. De plus, il est conçu pour permettre l'ajout de nouveaux modules d'une manière simple et efficace, tels que ceux de Markov cachés entraînés par le système HTS.

3. Marry TTS : Est un système de synthèse vocale open-source écrit en Java, livré avec des boîtes à outils, afin de faciliter l'ajout de nouvelles langues. Il utilise la méthode de synthèse par concaténation avec les diphones comme unité de base.

4. Tacotron : Est une architecture de réseau neuronal pour la synthèse vocale. Il est composé d'un encodeur, d'un décodeur et d'un réseau de post-traitement, afin de convertir la phrase d'entrée en une représentation d'entités cachées, puis en spectrogramme et finalement en parole.

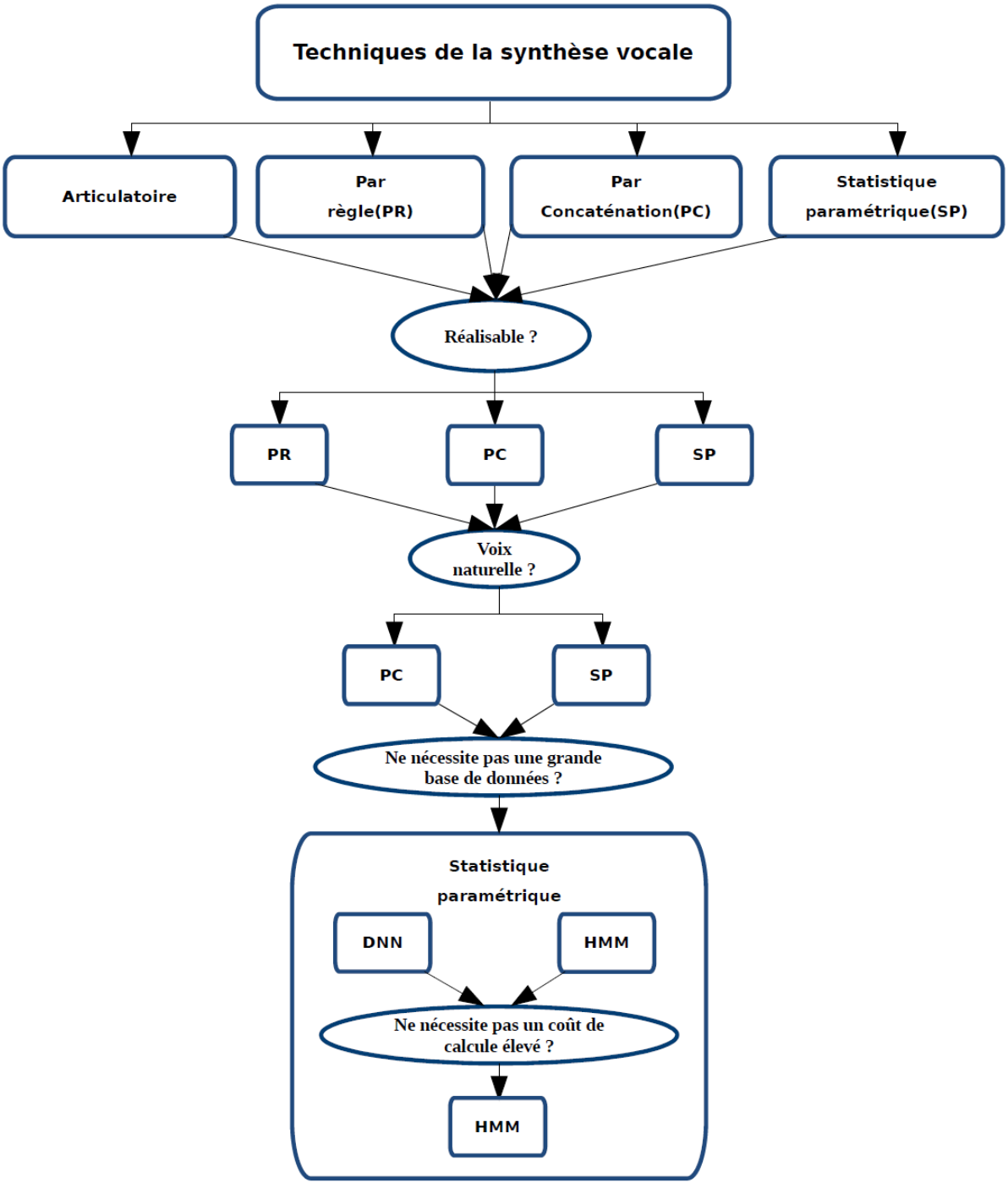


FIGURE 4.1 : Choix de la technique de synthèse vocale

Sachant que Festival est utilisé dans divers périphériques, qu'il ne nécessite pas de ressources puissantes et qu'on peut utiliser avec des voix basées sur HMM. Pour cela, Nous allons construire notre voix à l'aide du modèle de Markov caché implémenté par le système HTS, avec Festival comme module de traitement linguistique et de synthétisation.

HTS a été développé par le groupe de travail HTS et d'autres. La partie d'entraînement de HTS a été implémentée en tant que version modifiée de HTK (une Boîte à outils pour la construction et la manipulation des modèles de Markov cachés.) et publiée sous forme de code de patch pour ce dernier[31].

4.3 Architecture générale du système

Nous proposons le schéma suivant (figure 4.2) pour montrer l'architecture générale de notre système :

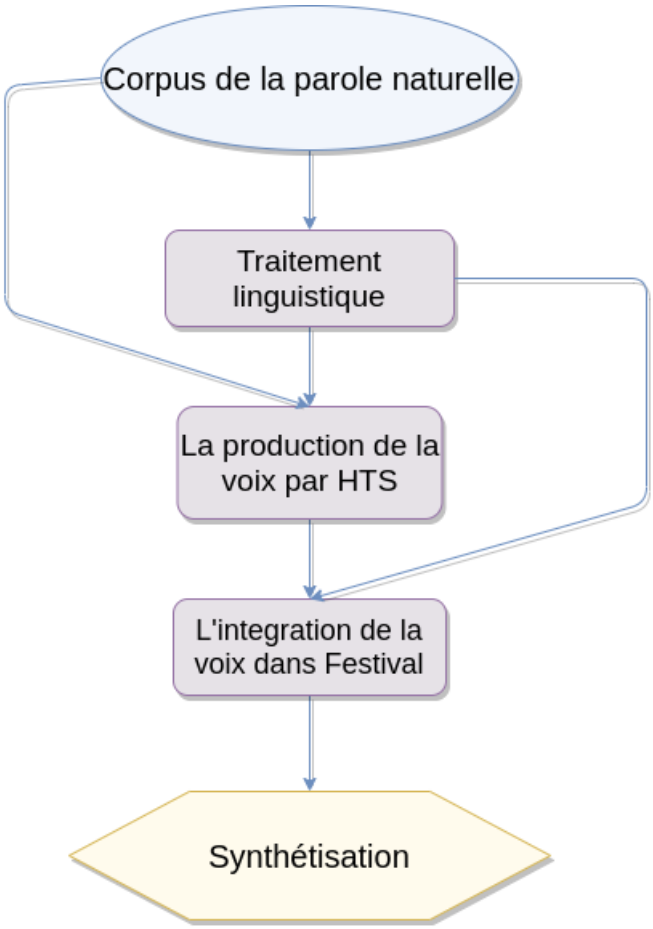


FIGURE 4.2 : Architecture générale du système.

Dans le but de créer une voix arabe synthétisée, nous avons besoin d'un corpus de la parole naturelle, qui va passer par l'analyse de texte puis par la production

de la voix en utilisant HMM afin de l'intégrer dans Festival. Pour cela nous allons aborder les points suivants :

- La préparation d'un corpus de la parole naturelle, sur lequel notre système va s'entraîner.
- Le traitement linguistique, afin de produire les caractéristiques linguistiques du corpus utilisé.
- Ajouter le corpus et les caractéristiques extraites au système HTS pour faire l'entraînement et la production de la voix.
- Le dernier point consiste à intégrer cette voix au système Festival, afin qu'il sera utilisé pour la synthétisation de la parole à partir d'un texte arabe vocalisé.

4.4 La préparation d'un corpus de la parole naturelle

Un corpus de parole est une base de données de fichiers audios de parole et de transcriptions de texte correspondantes.

Dans ce projet, nous aurons besoin d'un corpus convenable en termes de taille et de contenu, la construction d'un tel corpus prend du temps et nécessite des outils avancés. Les phonèmes doivent êtres prononcées d'une manière correcte et le texte doit couvrir le maximum des combinaison phonétiques possibles. Selon N. Halabi[7], pour obtenir une bonne voix synthétique nous avons besoin de 2 à 16 heures d'enregistrement.

Pour la langue arabe, il existe un corpus open source contenant 1813 fichiers audios (.wav), qui correspondent à 3.7 heures de parole avec leurs transcriptions textuelles vocalisées. Ce corpus a été créé dans le cadre du projet de doctorat de Nawar Halabi à l'université de Southampton[7]. Nous utilisons ce corpus pour extraire les caractéristiques linguistiques et acoustiques, afin que nous puissions former les HMM.

4.5 L'analyse de texte

Le but de l'analyse de texte est d'avoir des caractéristiques linguistiques représentées par des fichiers Utterance (.utt). Ces fichiers contiennent la transcription

phonétique correspondante de chaque audio, ainsi que le temps de début et de fin de chaque phonème.

Dans cette phase, nous allons exploiter le système Festvox qui est une suite d'outils créés par Alan W. Black et Kevin Lenzo pour construire des voix synthétiques pour Festival. Ce projet vise à rendre la construction d'une nouvelles voix synthétisée plus systémique et mieux documentée, afin d'automatiser la création des voix par des scripts configurables.[32]

Figure 4.3 montre les différentes opérations qui seront appliquées sur le corpus. Afin qu'elles soient annotées avec les fichiers audios du corpus, les transcriptions textuels seront tokenisées et normalisées pour être converties en phonèmes en utilisant des règles de transformation et un lexique. Nous détaillerons toutes ces opérations par la suite.

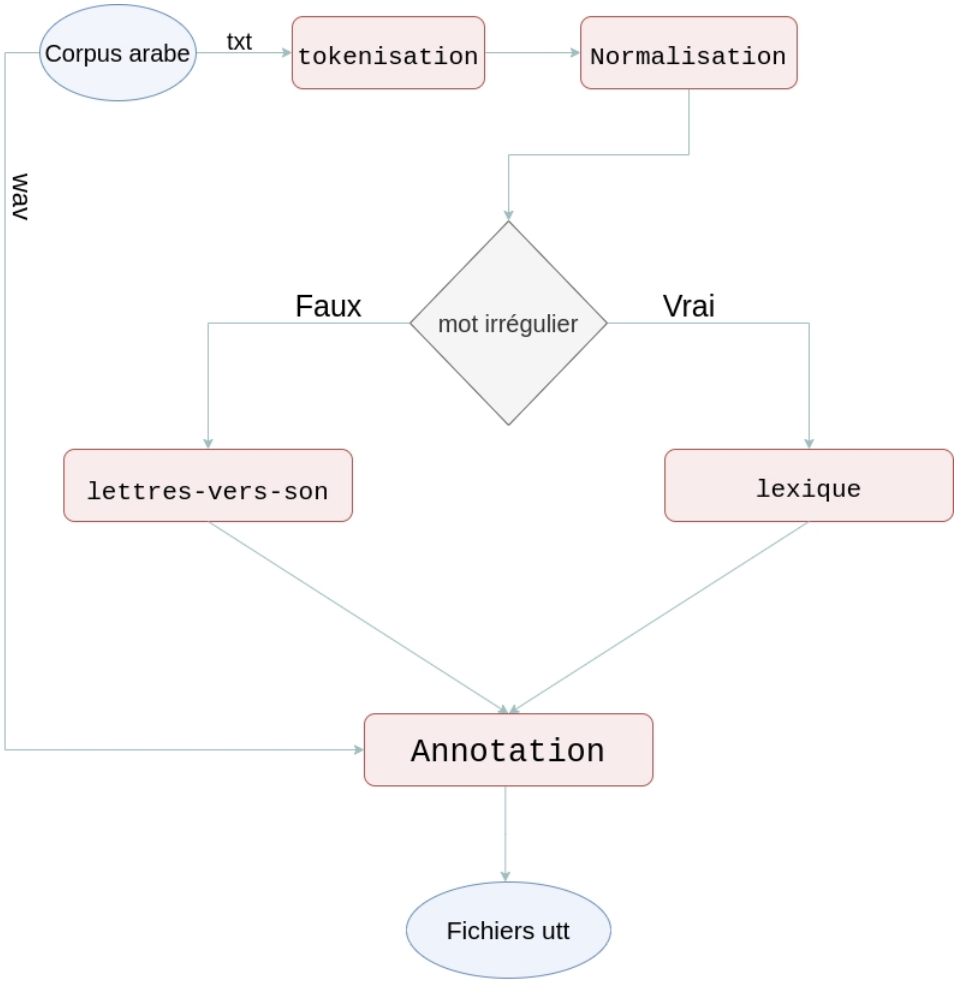


FIGURE 4.3 : L'analyse de texte

Festvox propose différents modules pour le traitement automatique du langage naturel, qui apparaissent sous forme de fichiers écrits en langage de programma-

tion Scheme dans le but d'être personnalisés pour une nouvelle langue.

Afin d'assurer l'analyse de texte, nous allons réaliser les tâches suivantes(figure 4.4) en suivant la documentation décrite dans le manuel du Festival[22].

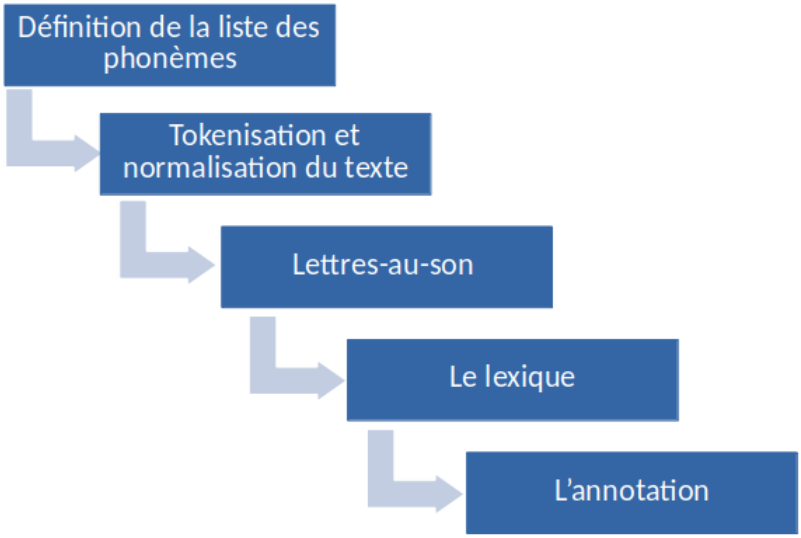


FIGURE 4.4 : Les tâches de l'analyse de texte.

4.5.1 Définition de la liste des phonèmes

La liste de tous les phonèmes de la langue arabe avec leurs propriétés phonétiques est déjà mentionnée dans le chapitre 2. Un ensemble de phonèmes est un ensemble de symboles qui peuvent être définis par des caractéristiques, telles que le genre des lettres, les point d'articulation des consonnes, les type de voyelles, etc.[22]

4.5.2 Tokenisation et normalisation du texte

Les règles de tokenisation incluent le découpage du texte en tokens en utilisant les espaces et les signes de ponctuation.

Après la tokenisation, la normalisation du texte est effectuée dans le but de transformer les chiffres, les abréviations et les symboles en texte. Par exemple " لَقَدْ دَفَعَ 1520 د.ج مِنْ أَجْلِ شِرَائِهَا " devient " لَقَدْ دَفَعَ أَلْف وَ خَمْسمِئَةٍ وعِشْرُونَ دِينَار جَزَائِرِي مِن أَجْلِ شِرَائِهَا".

4.5.3 Lettres-au-son

Les règles de conversion lettres-au-son représentent un dictionnaire qui transforme les lettres en phonèmes.

Une étape très importante dans la synthèse vocale, consiste à générer la représentation de prononciation correcte en fonction du texte saisi (texte aux phonèmes). Cela nécessite un mécanisme pour convertir les mots épelés dans la phrase d'entrée en une représentation phonétique.

Une approche de la conversion des lettres en phonèmes comporte l'utilisation d'un ensemble de règles linguistiques bien définies qui, dans la plupart des cas, sont complétées par un dictionnaire de mots spéciaux.[33]

Vue que l'écriture des mots arabes est très proche à leurs prononciation, les lettres peuvent être facilement transcrites en son grâce à un ensemble de règles lettre-au-son.

Ces règles sont exprimées de la forme suivante :[33]

$$A \rightarrow B/X\_Y$$

Où A et B peuvent être un seul caractère orthographique, une chaînes de caractères ou nul. La règle ci-dessus signifie que A devient B si A est entre le contexte gauche X et le contexte droit Y.

Il est important de noter que l'entrée et la sortie des règles lettre-au-son sont différentes, l'entrée contient des lettres et la sortie contient des phonèmes.

Le tableau 4.1 représente un exemple d'une entrée et la sortie correspondante.

TABLE 4.1 : Exemple de l'entrée et la sortie des lettre-au-son

يَتَمَثَّلُ الْإِبْدَاعُ الْفَنِّيُّ وَالْحَضَارِيُّ الَّذِي يَكْشِفُ عَنْهُ الْمَعْرِضُ فِي تُحَفٍ كَثِيرَةٍ	yatama^^alu ahalahibdaaEu ahalfanniyyu walHADaariyyu ahallathii yakchifu Eanhu ahalmaErIDu fii tuHafian ka^iiratian

4.5.4 Le lexique

Est un dictionnaire de mots spéciaux, indiquant un ensemble de correspondances de l'orthographe à la phonétique, qui est utilisé pour les mots irréguliers par apport à la prononciation, par exemple «هذا » se prononce «هاذا ».

Le tableau 4.2 contient des exemples de ces mots irréguliers.

TABLE 4.2 : Exemples des mots irrégulier

هذا	هذان	هؤلاء	الرحمن	ذلك
هذه	ذلكم	أولئك	طه	لكن

4.5.5 L'annotation

Elle prend en charge la mise en correspondance de chaque phonème avec son temps de début et de fin en utilisant un logiciel dédié au traitement de la parole.

4.6 La production de la voix par HTS

Après avoir obtenu la transcription phonétique correspondante de chaque audio, ainsi que le temps de début et de fin de chaque phonème, nous pouvons passer à la production de notre voix synthétique.

Cette partie présente un système de synthèse vocale basé sur le modèle de Markov caché. Dans la synthèse vocale basée sur HMM, les paramètres vocaux d'une unité vocale, tels que le spectre, la fréquence fondamentale (F0) et la durée du phonème sont modélisés statistiquement, et générés à l'aide de modèle de Markov caché basé sur le critère de vraisemblance maximale (maximum likelihood criterion).[34]

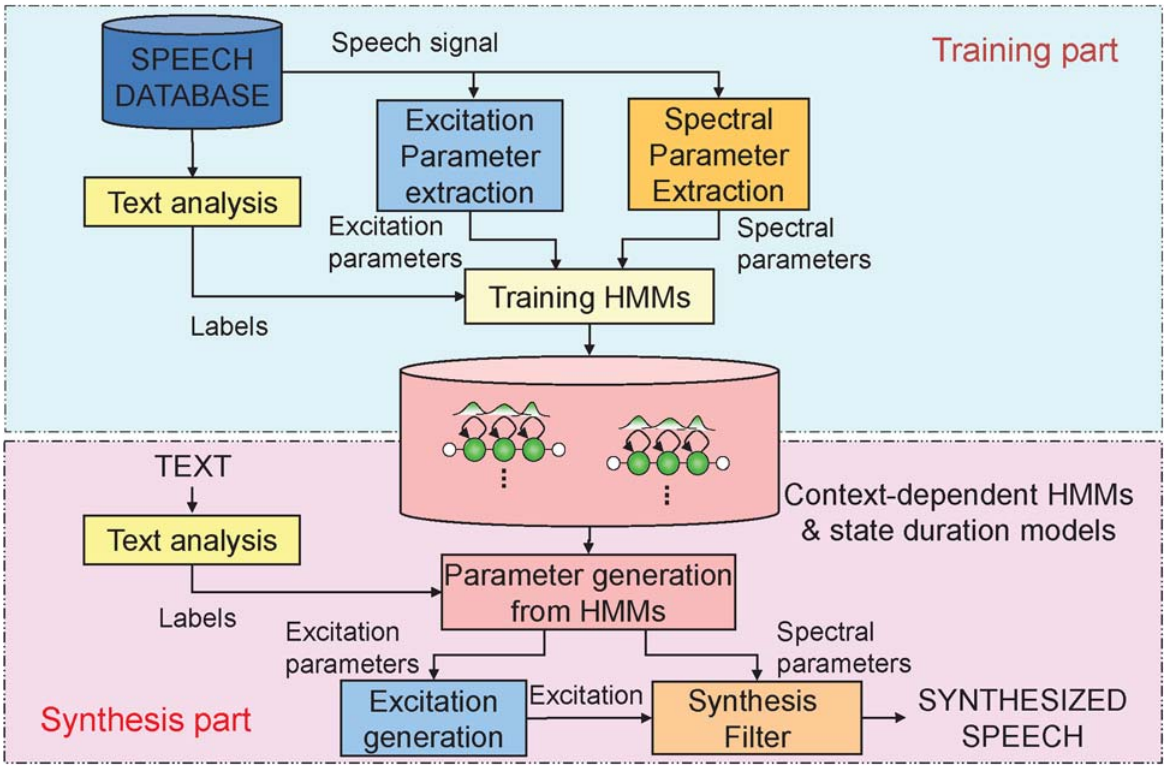


FIGURE 4.5 : Un aperçu d'un système de synthèse vocale basé sur HMM [35]

Un aperçu d'un système de synthèse vocale basé sur HMM proposé par Tokuda et al [35] est illustré à la figure 4.5. Le système est divisé en deux phases :

- La phase d'entraînement où des séquences de vecteurs de caractéristiques sont modélisées par des HMM dépendant du contexte. La procédure d'entraînement des HMM dépendants du contexte est presque la même que celle des systèmes de reconnaissance vocale[36].
- La phase de la synthétisation où un texte donné à synthétiser est converti en une séquence d'étiquette contextuelle. Une séquence HMM est construite en concaténant les HMM dépendants du contexte en fonction de la séquence d'étiquette.

4.6.1 La phase d'entraînement

La phase d'entraînement (figure 4.6) nécessite un corpus de la parole naturelle enregistrée, qui doit être correctement transcrite dans des fichiers Utterance précédemment obtenus.

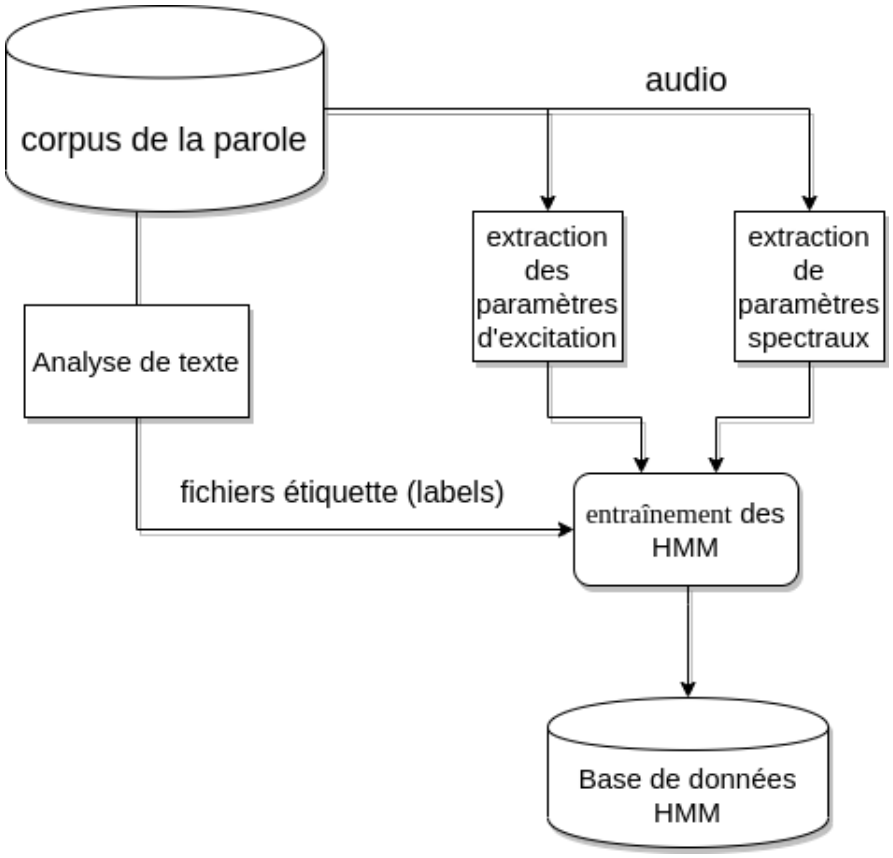


FIGURE 4.6 : La phase d'entraînement.

Nous devons utiliser un Vocodeur ("un outil de traitement de signal"), afin d'extraire les paramètres spectraux et d'excitation des fichiers audio [37]

Dans le but d'avoir la forme finale des caractéristiques linguistiques du corpus, nous devons créer les fichiers étiquettes (.lab). Ces fichiers contiennent des informations sur le contexte phonétique (durée, phonème actuel, phonèmes précédent / suivant), ainsi que les informations prosodiques (nombre de mots dans la phrase, nombre de syllabes dans le mot, position de la syllabe vis-à-vis du mot ...)[37].

Les étiquettes incluent les dépendances de contextes suivantes[38] :

- Phonème :
  - Le phonème actuel.
  - Les précédant et succédant à deux phonèmes.
  - La position du phonème courant dans la syllabe courante.
- Syllabe :
  - Le nombre de phonèmes dans les syllabes précédentes, actuelles et suivantes.
  - Le stress et l'accentuation des syllabes précédentes, actuelles et suivantes.
  - La position de la syllabe courante dans le mot et la phrase courants.
  - Le nombre de syllabes stressées avants et après, dans la phrase courante.
  - Le nombre de syllabes accentuées précédentes et suivantes dans la phrase courante.
  - Le nombre de syllabes à partir de la syllabe stressée précédente.
  - Le nombre de syllabes jusqu'à la syllabe stressée suivante.
  - Le nombre de syllabes à partir de la syllabe accentuée précédente.
  - Le nombre de syllabes jusqu'à la syllabe accentuée suivante.
  - L'identité de voyelle dans la syllabe courante.
- Mot :
  - Une estimation de la partie du discours des mots précédents, actuels et suivants.
  - Le nombre de syllabes dans les mots précédents, actuels et suivants.
  - La position du mot courant dans la phrase courante
  - Le nombre de mots lexicaux précédents et suivants, dans la phrase actuelle

  - Le nombre de mots à partir de mot lexical précédent.
  - Le nombre de mots jusqu'au mot lexical suivant.

- Phrase :
  - Le nombre de syllabes dans les phrases précédentes, actuelles et suivantes.
  - La position de la phrase courante dans l'énoncé.
  - ToBI endtone de la phrase actuelle.

- Utterance :
  - le nombre de syllabes, de mots et de phrases dans l'énoncé.

Les étiquettes phonétiques et les caractéristiques acoustique sont utilisées pour modéliser les HMM dépendants du contexte

Une base de données vocale limitée ne peut pas couvrir le grand nombre de phonèmes contextuels d'une langue. L'un des avantages apportés par la technique de synthèse basée sur HMM consiste à la nécessité d'une petite base de données, contenant environ 500 à 1 000 phrases pour effectuer la formation de modèles de phonèmes dépendant du contexte. Une telle limitation de corpus n'est possible que dans la mesure où HTS utilise la technique de l'arbre de décision. Cette technique permet d'estimer les paramètres HMM pour les phonèmes dans des contextes qui n'apparaissent pas dans la base de données vocales utilisée pour l'entraînement des modèles. Étant donné que les facteurs contextuels ont des influences différentes sur le spectre, la fréquence de la hauteur et la durée, trois arbres de décision (Figure 4.7) différents sont nécessaires pour traiter les différents contextes.[39]

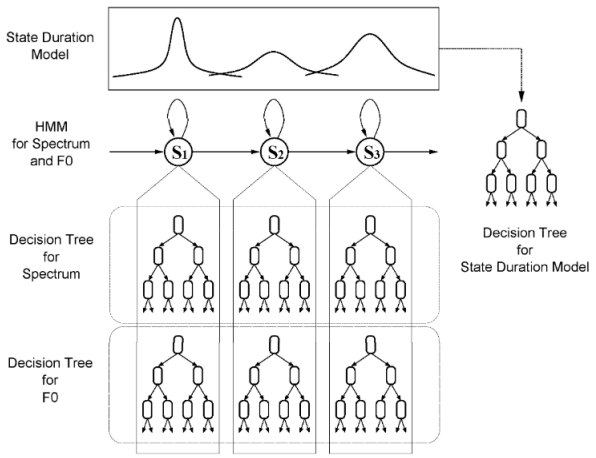


FIGURE 4.7 : Arbres de décision pour le clustering de contexte[3]

L'entraînement des HMM effectue l'estimation du maximum de vraisemblance des paramètres HMM en utilisant l'algorithme Baum–Welch[38].

Dans ce processus, les paramètres statistiques des HMM sont calculés. Ensuite, des arbres de décision qui décrivent tous les facteurs contextuels sont utilisés pour regrouper les HMM entraînés.[40]

4.6.2 La phase de synthétisation

La phase de synthétisation montrée dans la partie inférieure de la figure 4.5, est illustré dans la figure 4.8

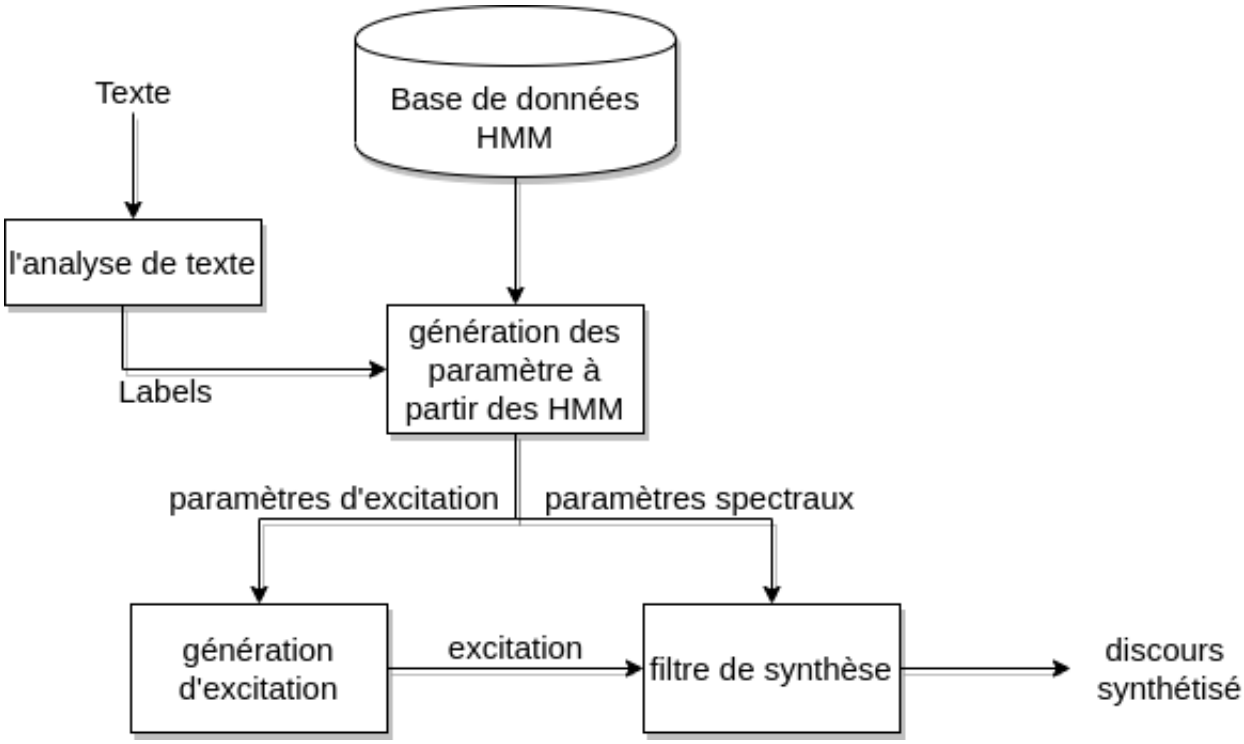


FIGURE 4.8 : La phase de synthétisation

Pour la synthétisation, des HMM dépendants du contexte sont concaténés afin de synthétiser le texte en entrée. Tout d'abord ce texte est transcrit en un fichier étiquette (lab), et pour chaque phonème, le HMM du même contexte est sélectionné pour construire toute la phrase. Ensuite, une séquence de paramètres vocaux comprenant des paramètres spectraux et d'excitation est déterminée de manière à maximiser sa probabilité de sortie[35].Finalement, une forme d'onde de la parole est créée à l'aide du module de génération d'excitation et un filtre de synthèse défini par les paramètres générés par HMM.

4.7 L'intégration de la voix dans Festival

Après la production de la voix arabe par le systèmes HTS, seuls les paramètres HMM et l'arborescence de contexte seront stockés. Par la suite, nous pouvons les intégrer dans le système de synthèse vocal Festival avec les scripts de traitement linguistique (normalisation, règles de phonétisation, ...), afin que les utilisateurs puissent utiliser notre voix.

4.8 Conclusion

Dans ce chapitre, nous avons présenté l'étude conceptuelle du projet, en commençant par le choix du système, qui consiste à utiliser le système Festival comme un module de traitement linguistique et de synthétisation, et le système HTS pour la production des HMM. Par la suite, nous avons expliqué notre architecture globale, y compris la préparation du corpus, le traitement linguistique, la production de la voix avec HTS et enfin l'intégration de la voix dans Festival.

Le chapitre suivant traite les différents processus de l'implémentation du projet pour construire une voix arabe proche de la voix humaine.

Chapitre 5

Implémentation

5.1 Introduction

Après la présentation de notre solution d'un point de vue conceptuel, nous passons au point de vue technique qui détaillera les démarches de travail et déterminera les étapes de la création d'une voix arabe à l'aide du modèle de Markov caché implémenté par le système HTS, avec le système Festival comme module de traitement linguistique et de synthétisation.

Les caractéristiques les plus demandées pour un système de synthèse vocale sont :

- La naturalité : indique la similitude de la parole synthétisée avec la voix humaine.
- L'intelligibilité : se réfère à la facilité de compréhension de la voix synthétisée par les auditeurs.

Il existe deux principaux types de données pour la création d'une nouvelle voix : acoustiques et linguistiques, qui commencent généralement par les enregistrements audio et leurs transcriptions textuelles respectives.

Le développement d'une voix nécessite deux phases : le traitement linguistique et l'entraînement. Le système HTS supporte l'entraînement et fournit les outils nécessaires pour travailler avec les modèles statistiques et former des HMM contextuels. Mais ce logiciel ne comprend pas d'analyseur de texte. Par conséquent, ce chapitre se concentre d'abord sur l'utilisation du système Festival pour effectuer le traitement linguistique de toutes les transcriptions textuelles présentes dans le corpus de Nawar Halabi[1]. Ensuite, nous allons utilisé HTS pour

[1]http ://en.arabicspeechcorpus.com/

faire l'étiquetage, extraire les caractéristiques acoustiques et définir des questions qui permettent la construction de l'arbre de décision. Une fois que ces fichiers sont connus, nous pouvons commencer l'entraînement.

Dans ce chapitre, nous allons d'abord présenter les outils nécessaires, puis détailler les différentes étapes de la production d'une nouvelle voix. Nous terminerons en faisant référence à l'utilisation de notre système.

5.2 Les outils utilisés

Afin de réaliser ce projet, un ensemble d'outils doit être utilisé pour effectuer correctement toutes les taches nécessaires liées les unes aux autres. (Tableau 5.1)

TABLE 5.1 : Outils utilisés.

Festival	La production des caractéristiques linguistiques et la synthétisation.	
Festvox	Produire une représentation linguistique du texte en utilisant des scripts à base de Festival.	
Speech Tools	Manipuler les types d'objets utilisés dans le traitement de la parole.	
HTS-demo	Système de synthèse vocale basé sur les modèles de Markov cachés.	
HTK	Boîte à outils pour la construction et la manipulation de modèles de Markov cachés.	
SPTK	Suite d'outils de traitement du signal vocal pour les environnements UNIX.	
HTS_Engine	Logiciel pour synthétiser la parole à partir de HMM's formés par le système HTS.	

Le système de synthèse vocale Festival

Le système Festival définit dans le chapitre 3 est utilisé en deux phases : la production des caractéristiques linguistiques et la synthétisation.

Dans la première phase, Festival va être configuré est utilisé avec Festvox et Speech Tools afin de créer des fichiers Utterance dépendant des caractéristiques linguistiques du texte saisi.

La deuxième phase consiste à utiliser Festival pour synthétiser le texte écrit en arabe en intégrant la voix créée dans ce système.

Festvox

La création de la parole synthétique couvre toute une gamme de processus, afin que chacun puisse construire de nouvelle voix.

Le projet Festvox a été créé pour produire une représentation linguistique du texte en utilisant des scripts à base de Festival écrits en Scheme. Cette représentation est utilisée par HTS comme un élément essentielle dans la préparation des données pour la construction d'une nouvelle voix synthétique.[2]

Scheme est un langage de programmation fonctionnel et embarqué. Il permet aux développeur de configurer et de programmer des modules sans toucher à la structure du système[3].

La bibliothèque Speech Tools d'Edinburgh

La bibliothèque Speech Tools d'Edinburgh est un ensemble de classes C++, de fonctions et de programmes associés permettant de manipuler les types d'objets utilisés dans le traitement de la parole. Elle a été développée par Alan Black, Paul Taylor et d'autres, au centre de technologie de la parole à l'Université d'Edinburgh. Elle prend en charge la lecture et l'écriture des formes d'onde, des fichiers de paramètres vocaux dans différents formats, des objets de type linguistique et de la conversion entre eux.

Cette bibliothèque comprend de nombreux programmes intégrés. Une bibliothèque d'intonation, y compris un système de suivi de la hauteur, un système de lissage et d'étiquetage et un programme pour la construction d'arbre de classification. Speech Tools est conçue pour faciliter la construction d'autres systèmes vocaux comme Festival. Elle est actuellement distribuée gratuitement pour une utilisation complète et sans restriction.[4]

[2]http ://www.festvox.org/
[3]https ://www.scheme.com
[4]http ://www.cstr.ed.ac.uk/projects/speech$_t$*ools*/

HTS-demo

HTS est un outil capable de travailler avec les Modèles de Markov Cachés. Il permet à l'utilisateur d'effectuer des formulations probabilistes, de construire des modèles acoustiques, d'effectuer une analyse spectrale et de créer des filtres de synthèse. Il peut donc effectuer toutes les procédures nécessaires à la synthèse de la parole.[39]

Le système HTS a été construit sur la base de la boîte à outils du modèle de Markov caché (HTK). La version HTS-2.3 utilisée a été mise à disposition en décembre 2015. Ce système fonctionne en adaptant certaines routines et fonctions HTK pour permettre la synthèse vocale.[5]

HTS a une structure fonctionnelle basé sur HMM qui peut être divisée en deux étapes de formation et de synthèse. En outre, il est nécessaire d'installer d'autres programmes pour l'exécuter. Le premier programme à installer est HTK, qui est adapté pour fonctionner avec la synthèse vocale par application de patch. Le deuxième programme est HDecode, qui avec HTK, forment la plate-forme de base de HTS.

Afin d'effectuer certaines fonctions telles que l'extraction des paramètres acoustiques du corpus, il peut être utile d'utiliser SPTK. De plus, le moteur HTS_Engine fournit des routines orientées pour les développeurs, ce qui réduit la charge de calcul dans la synthèse vocale.[39]

HTS utilise plusieurs langages tels que Perl, AWK et python. Perl et AWK sont utilisés pour le traitement des séquences et des chaînes de caractères, ainsi que l'écriture des programmes courts et ponctuels afin de travailler sur des corpus pour l'extraction des colonnes de plusieurs ensembles de données[6].[41]

La boîte à outils du modèle de Markov caché (HTK)

La boîte à outils HTK est une boîte pour la construction et la manipulation de Modèles de Markov Cachés. HTK est principalement utilisée pour la recherche sur la reconnaissance vocale, bien qu'elle ait été utilisée pour de nombreuses autres applications, notamment la recherche sur la synthèse vocale et la reconnaissance de caractères dans des centaines de sites dans le monde.

[5]http ://hts.sp.nitech.ac.jp/
[6]https ://perldoc.perl.org/

HTK a été initialement développée au (Machine Intelligence Laboratory) du CUED (Cambridge University Engineering Department), où elle a été utilisée pour construire les grands systèmes de la reconnaissance vocale.[7]

La boîte à outils de traitement du signal vocal (SPTK)

La boîte à outils SPTK est une suite d'outils de traitement du signal vocal pour les environnements UNIX, elle a été développée par le groupe de travail SPTK[8] et quelques étudiants diplômés de l'institut de technologie de Nagoya. SPTK est open source pour une utilisation complète et gratuite. Les codes sources originaux ont été écrits par de nombreuses personnes qui ont participé aux activités du groupe de recherche.[9]

HTS_Engine API

HTS_Engine est un logiciel pour synthétiser la parole à partir de HMM formés par le système HTS. Il a été développé par le groupe de travail HTS et l'Institut de Technologie de Nagoya.[10]

5.3 Les démarches de travail

Nous pouvons résumer les démarches de la réalisation montrées dans la figure 5.1 comme suit :

Le traitement linguistique : C'est la production de caractéristiques linguistiques dans des fichiers Utterance à partir des transcriptions de texte arabe vocalisé et des fichiers audio correspondants. Cela nécessite généralement des ressources supplémentaires telles que des dictionnaires, des règles de prononciation et la description des phonèmes pour la langue arabe.

La production de la voix par HTS : Elles commence par l'extraction des caractéristiques acoustiques du signal audio brut telles que la représentation de la hauteur, les propriétés spectrales de l'audio, et la composition entre ces deux caractéristiques. Ensuite, un étiquetage des fichiers Utterance est demandé, puis une définition de fichiers de questions pour construire l'arbre de décision. Après

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

[8]Le groupe de travail SPTK est un groupe volontaire pour développer les outils de traitement du signal vocal. http ://sp-tk.sourceforge.net/readme.php

[9]http ://sp-tk.sourceforge.net/

[10]http ://hts-engine.sourceforge.net/

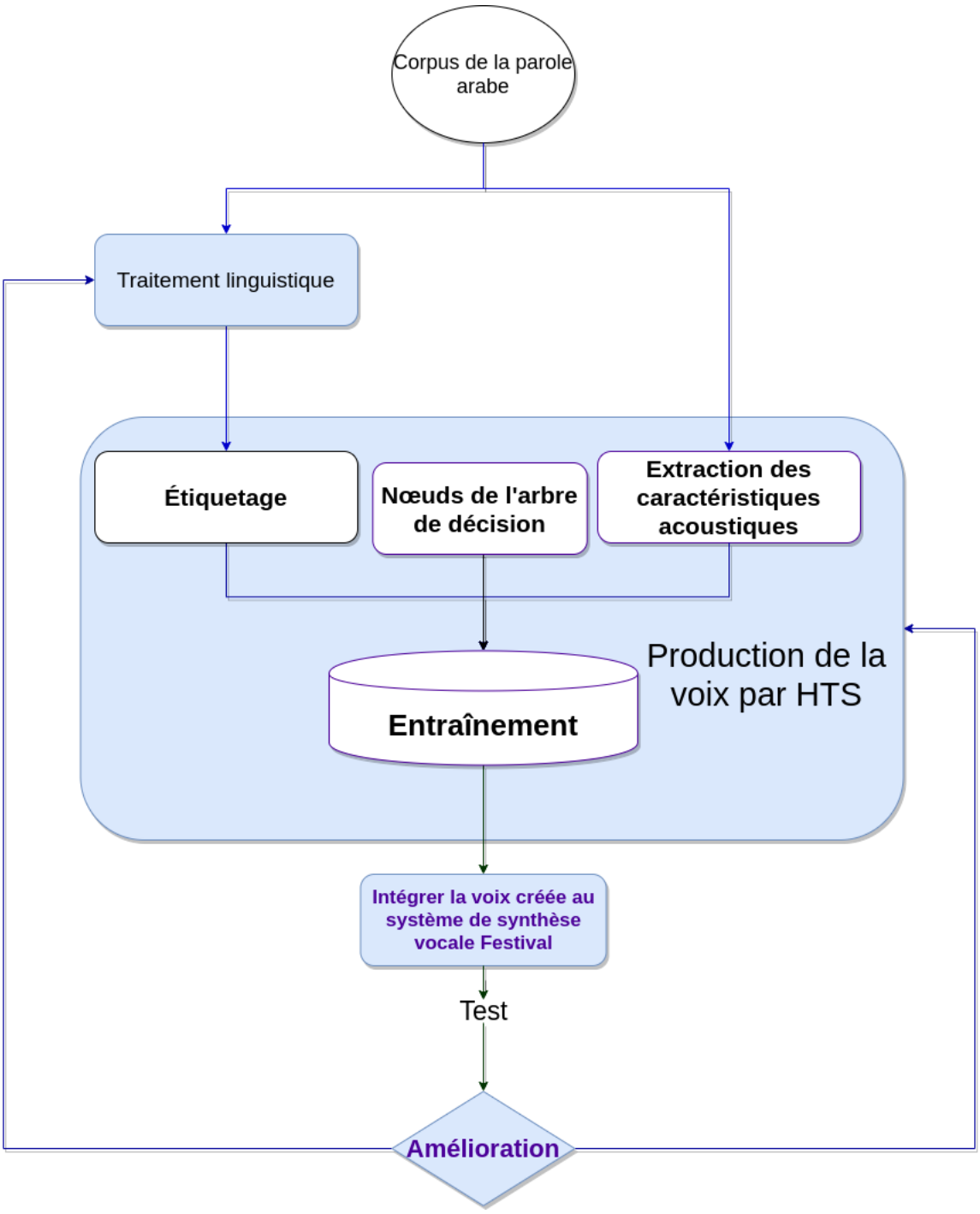


FIGURE 5.1 : Démarches de la réalisation.

avoir préparé toutes les données et extrait les fichiers nécessaires, nous les utiliserons pour l'entraînement des modèles HMM.

L'intégration de la voix créée au système de synthèse vocale Festival : Après avoir terminé les étapes d'entraînement, des fichiers décrivant toutes les caractéristiques vocales seront créés en sortie pour la synthétisation. Par la suite, nous pouvons les ajouter au Festival, avec les fichiers qui font le traitement linguistique, afin de représenter la langue arabe.

L'amélioration de la qualité vocale générée : Cette phase est très importante pour avoir une meilleure qualité de la voix arabe synthétisée. Après chaque entraînement, il faut améliorer l'extraction des caractéristiques acoustiques et linguistiques, en comparant les résultats obtenus et les résultats voulus.

5.3.1 Traitement linguistique

Le but est d'obtenir à partir d'une analyse linguistique du texte arabe saisi, sa transcription phonétique dépendant des caractéristiques linguistiques qui vont être représentées par des fichiers Utterance de format (.utt) (figure 5.2).

Toutes les particularités de la langue doivent être traitées afin d'éviter les erreurs de prononciation artificielle de la parole.

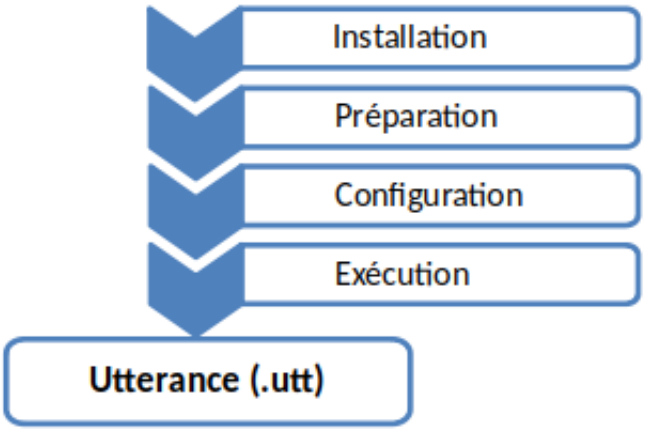


FIGURE 5.2 : L'analyse de texte.

5.3.1.1 Installation

Nous devons d'abord préparer l'environnement de travail en spécifiant le chemin de Festival, Festvox et Speech Tools, et en appliquant quelques commandes pour avoir au final un répertoire qui contient l'architecture complète du système

nécessaire. Ces programmes sont open-source et peuvent être téléchargés à partir des adresses en ligne suivantes [39] :

- Festival 2.4 et Speech Tools 2.4 : http ://festvox.org/packed/festival/
- Festvox 2.7 : http ://festvox.org/festvox-2.7/

Une fois tous les fichiers créés, nous devons ouvrir le terminal et taper les commandes suivantes [42] :

⇒ tar xvf festival-2.4-release.tar.gz
cd festival/
./configure
make

⇒ tar xvf speech_tools-2.4-release.tar.gz
cd speech_tools/
./configure
make

⇒ tar xvf festvox-2.7.0-release.tar.gz
cd festvox/
./configure
make

Maintenant que tous les programmes nécessaires ont été installés, l'étape suivante consiste à indiquer leurs chemins et à créer un environnement de travail [43] :

⇒ export PATH=/home/HTS/festival/bin :$PATH
export PATH=/home/HTS/speech_tools/bin :$PATH
export FESTVOXDIR=/home/HTS/festvox
export ESTDIR=/home/HTS/speech_tools

⇒ $FESTVOXDIR/src/clustergen/setup_cg ara norm ziad

Telle que :
ara_norm est le nom de la base de Nawar Halabi.
ziad est le nom de locuteur.

5.3.1.2 Préparation

Une fois tous les outils sont installés, plusieurs fichiers seront crées par Festvox pour couvrir le processus de création de fichiers Utterance via des scripts Festival, Festvox et Speech Tools qui se trouvent dans le dossier /bin/.

Les fichiers que nous devons configurer et adapter avec la langue arabe se présentent dans le répertoire /festvox/.

Avant de commencer la configuration, nous devons préparer les transcriptions textuelles et les fichiers audios présentés dans le corpus, afin qu'ils conviennent avec la forme exigée dans le traitement linguistique. (figure 5.3)

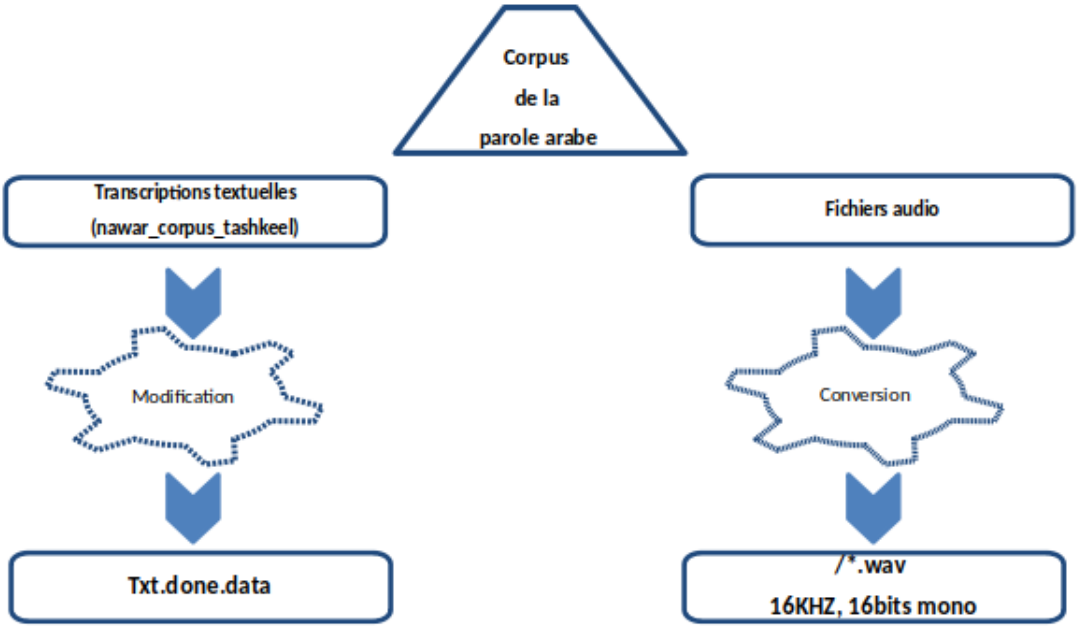


FIGURE 5.3 : Préparation du corpus pour l'analyse de texte.

a- La création de fichier 'txt.done.data'

'Txt.done.data' a été créé à partir des transcriptions textuelles écrites en arabe présentées dans le fichier (nawar_corpus_tashkeel), qui se trouvent dans le projet de Motaz Saad[11]. Afin de rendre le texte cohérent avec la forme requis par Festival, nous devons apporter certaines modifications comme suit :

(Le nom de fichier audio "La transcription textuelle correspondante")

Ce fichier doit être placé dans : /Etc/ .[43]

```
      ...
( ara_norm_ziad_0010 "وَكَامِرَاتِ التَّصْوِيرِ الْمُدْمَجَةِ ذَاتِ الدِّقَّةِ الْعَالِيَةِ")
( ara_norm_ziad_0011 "وَالْمُعَالِجَاتِ الثُّنَائِيَّةِ وَالرُّبَاعِيَّةِ النَّوَى" )
( ara_norm_ziad_0012 "وَحَالَةِ الْإِسْتِقْطَابِ السِّيَاسِيِّ الْحَادِّ" )
( ara_norm_ziad_0013 "لِيَتَمَكَّنَ الْقَارِئُ مِنْ مُتَابَعَةِ تَطَوُّرِ الْقِصَّةِ الْخَبَرِيَّةِ بِشَكْلٍ سَلِسٍ وَفَعَّالٍ" )
( ara_norm_ziad_0014 "وَمِنْهَا أَيْضاً - أَلدَّعْمُ الْمُتَقَدِّمُ لِلْأَرْشَفَةِ" )
      ...
```

[11]https ://github.com/motazsaad/ara-pronunciation-tool

b- Ajouter les fichiers audio (.wav)

Avant d'ajouter les fichiers .wav, il faut les convertir d'abord au format 16 kHz, 16 bits mono, RIFF.
Ces fichiers doivent être placés dans : /wav/ .[43]

5.3.1.3 Configuration

Cette phase est le corps principal de traitement linguistique, où nous allons définir l'architecture complète telle que les règles et les fichiers nécessaires pour convertir le texte en phonèmes (figure 5.4). Certaines règles, ainsi que le choix des phonèmes et le codage des lettres sont basés sur le travail de Nawar Halabi[12].

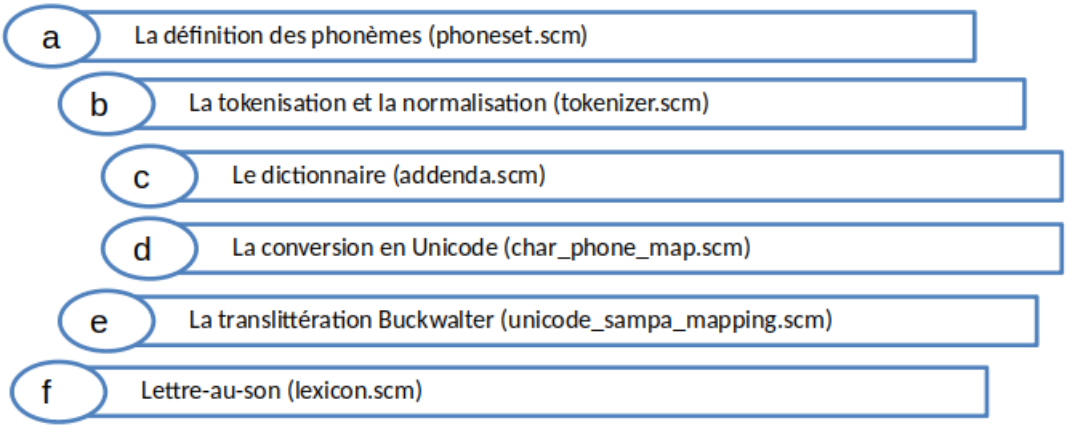


FIGURE 5.4 : Configuration des fichiers Scheme.

Pour la configuration, nous devons déterminer les fichiers suivants (écrits en Scheme) qui se trouvent dans : /festvox/ .[44]

a- La définition des phonèmes (phoneset.scm)

La détermination de la liste des phonèmes nécessite une reconnaissance globale sur les caractéristiques et les points d'articulation des consonnes, et le type des voyelles (chapitre 2). Toutes les opérations ultérieures, dépendent de cette étape pour fonctionner.

Voici la forme de fichier phoneset.scm :

```
(defPhoneSet
    NAME
    FEATUREDEFS
```

[12]https ://github.com/nawarhalabi/Arabic-Phonetiser

```
PHONEDEFS )
```

I 'NAME' est n'importe quel symbole unique utilisé pour nommer la fonction. Par exemple :

```
(defPhoneSet
    ara_norm
```

II 'FEATUREDEFS' est une liste prédéfinie, qui inclut les propriétés phonétiques et ses valeurs possibles comme suit :

```
(;; vowel or consonant
  (vc + -)
  ;; vowel length: short long dipthong schwa
  (vlng s l d a 0)
  ;; vowel height: high mid low
  (vheight 1 2 3 0)
  ;; vowel frontness: front mid back
  (vfront 1 2 3 0)
  ;; lip rounding
  (vrnd + - 0)
  ;; consonant type: stop fricative affricative nasal liquid approximant
  (ctype s f a n l r 0)
  ;; place of articulation: labial alveolar palatal labio-dental dental
      ↪ velar glottal
  (cplace l a p b d v g 0)
  ;; consonant voicing
  (cvox + - 0)
  )
```

III 'PHONEDEFS' est une liste de définition des phonèmes, chacune se compose de nom de phonème et des valeurs des fonctions précédentes, en suivant l'ordre dans lesquels ont été définies ci-dessus.

Les consonnes : Nous avons défini 56 phonèmes pour les consonnes qui sont représentées par les 28 lettres de la langue arabe et leurs phonèmes doublés (lettre+shada) correspondants. Voici un exemple :

```
      ...
(H - 0 0 0 0 f v -) ;;ح
(HH - 0 0 0 0 f v -) ;;ح + شدة

(x - 0 0 0 0 f v -) ;;خ
(xx - 0 0 0 0 f v -) ;;خ + شدة
```

```
(d - 0 0 0 0 s a +) ;;د
(dd - 0 0 0 0 s a +) ;;د + شدة
       ...
```

Les voyelles : voici toute la liste de voyelles utilisées.

```
(a + s 3 2 - 0 0 0) ;;فتحة
(aa + l 3 2 - 0 0 0) ;;فتحة ممدودة

(A + s 3 2 - 0 0 0) ;;فتحة + (ص,ض,ط,ظ,ق)
(AA + l 3 2 - 0 0 0) ;;فتحة ممدودة + (ص,ض,ط,ظ,ق)

(i + s 1 1 - 0 0 0) ;;كسرة
(ia + s 1 1 - 0 0 0) ;;كسرة تقارب الفتحة
(ii + l 1 1 - 0 0 0) ;;كسرة ممدودة

(I + s 1 1 - 0 0 0) ;;كسرة + (ص,ض,ط,ظ,ق)
(II + l 1 1 - 0 0 0) ;;كسرة ممدودة + (ص,ض,ط,ظ,ق)

(u + s 1 3 + 0 0 0) ;;ضمة
(uu + l 1 3 + 0 0 0) ;;ضمة ممدودة

(U + s 1 3 + 0 0 0) ;;ضمة + (ص,ض,ط,ظ,ق)
(UU + l 1 3 + 0 0 0) ;;ضمة ممدودة + (ص,ض,ط,ظ,ق)

(ua + s 1 3 + 0 0 0) ;;ضمة تقارب الفتحة
(uua + l 1 3 + 0 0 0) ;;(الفيديو)
```

Une voyelle suivit d'une lettre emphatique (ص, ض, ط, ظ, ق) sera identifiée différemment d'une voyelle suivit d'une lettre normale en raison de la différence de prononciation.

Il faut noter que la liste de phonèmes doit également inclure une définition pour tous les phonèmes silencieux, exemple :

```
(sil - 0 0 0 0 0 0 -)
(pau - 0 0 0 0 0 0 -)
```

b- La tokenisation et la normalisation (tokenizer.scm)

Les règles de tokenisation, qui sont les même pour la langue arabe, sont déjà définies comme une fonction standard pour l'anglais. De plus, il faut ajouter les signes de ponctuation arabe (virgule, point-virgule؛ point d'interrogation ؟).

La normalisation nécessite que les nombres, abréviations, symboles et autres caractères non alphabétiques soient regroupés pour les convertir en texte indiquant comment ils doivent être prononcés.

Par exemple, le nombre 112 est composé de trois chiffres. Il faut convertir d'abord le nombre 1xx en (مَائَةٌ وَ), ensuite, nous passons à la fonction inférieure afin de convertir le nombre 12 en (إِثنَا عَشَرَ) comme suit :

```
(define (ara_norm_nawar::token_to_words token name)
       ...
       (let ((l (length name)))
       ...
              ((equal? l 1);; single digit
         (cond
          ((string-equal (car digits) "0") (list "صِفْرٌ"))
          ((string-equal (car digits) "1") (list "وَاحِدٌ"))
          ((string-equal (car digits) "2") (list "اِثْنَانِ"))
                      ...

        (equal? l 2);; less than 100

          ((string-equal (car digits) "1");; 1x
           (cond
            ((string-equal (car (cdr digits)) "0") (list "عَشْرَةٌ"))
            ((string-equal (car (cdr digits)) "1") (list "اِحْدَى عَشَرَ"))
            ((string-equal (car (cdr digits)) "2") (list "اِثْنَا عَشَرَ"))
                ...
         ((equal? l 3);; in the hundreds
              (cond
               ((string-equal (car digits) "1");; 1xx
                       (if (just_zeros (cdr digits)) (list "مَائَةٌ/")
                       (cons "مَائَةٌ وَ" (arabic_number_from_digits (cdr digits)))))
                ...
     ...
)
```

c- Le dictionnaire (addenda.scm)

Une fois que la normalisation du texte est effectuée, les mots sont transcrits phonétiquement. La langue arabe contient des mots dont la prononciation diffère de leur écriture comme le mot هَذَا qui se prononce هَاذَا. Par conséquent, avant de poursuivre les règles de conversion des lettres en sons, ces mots doivent d'abord passer par un dictionnaire contenant la fonction 'ADDENDA', qui consiste en des mots irréguliers ajoutés à la main.

```
. . .
;;بذلك
(lex.add.entry
'("بذلك" nil (((b i) 0) ((th aa) 0) ((l i) 0) ((k a) 0))))
(lex.add.entry
'("بِذَلِكَ" nil (((b i) 0) ((th aa) 0) ((l i) 0) ((k a) 0))))

;;كذلك
(lex.add.entry
'("كذلك" nil (((k a) 0) ((th aa) 0) ((l i) 0) ((k a) 0))))
(lex.add.entry
'("كَذَلِكَ" nil (((k a) 0) ((th aa) 0) ((l i) 0) ((k a) 0))))

;;ذلكم
(lex.add.entry
'("ذلكم" nil (((th aa) 0) ((l i) 0) ((k u m) 0))))
(lex.add.entry
'("ذَلِكُمْ" nil (((th aa) 0) ((l i) 0) ((k u m) 0))))

;;أولئك
(lex.add.entry
'("أولئك" nil (((ah u) 0) ((l aa) 0) ((ah i) 0) ((k a) 0))))
(lex.add.entry
'("أُولَئِكَ" nil (((ah u) 0) ((l aa) 0) ((ah i) 0) ((k a) 0))))
. . .
```

d- La conversion en Unicode (char_phone_map.scm)

Les lettres arabes et les diacritiques (فتحة, ضمة,كسرة, سكون, التنوين, الشدة) ne sont pas lisibles par le langage Scheme. Par conséquent, avant d'appliquer les règles de conversion des lettres en sons, nous devons convertir le texte en Unicode via une liste de type de cartes <lettre,Unicode>.
Ce mécanisme facilite l'analyse de texte écrit en arabe, et nous permettra d'éviter toutes les erreurs qui peuvent survenir lors de la transcription phonétique. Exemple :

```
...
( "ا" u0627p )
( "و" u0648p )
( "ل" u0644p )
( "ي" u064Ap )
( "ر" u0631p )
( "م" u0645p )
( "ب" u0628p )
```

```
( "ع" u0639p )
( "ة" u0629p )
( "ف" u0641p )
( "ه" u0647p )
( "ق" u0642p )
( "ً" u064Dp )
```

e- La translittération Buckwalter (unicode_sampa_mapping.scm)

Le processus fait appel à ce fichier pour symboliser les lettres converties en Unicode afin de les utiliser dans les règles de la transcription phonétique. Chaque lettre a un symbole spécifique pour la représenter comme suit :

```
...
( u0629p (( p ))) ;;ة
( u064Fp (( u ))) ;;ضمة
( u0637p (( T ))) ;;ط
( u0645p (( m ))) ;;م
( u0631p (( r ))) ;;ر
( u0638p (( Z ))) ;;ظ
( u0646p (( n ))) ;;ن
( u0632p (( z ))) ;;ز
( u0639p (( E ))) ;;ع
...
```

f- Lettre-au-son (lexicon.scm)

La lecture d'un texte arabe vocalisé nous permettra de simplifier les règles de conversion.

Après la tokenisation, la normalisation, le dictionnaire de mots irréguliers et le codage des lettres, le texte passe par les règles de la transcription phonétique. La définition de ce fichier nécessite un ensemble précis de règles, qui couvre toutes les possibilités de la composition des lettres dans le texte selon les règles d'orthographe de l'arabe.

Ce fichier contient la fonction 'LTS.RULESET' qui comprend un ensemble de règles pour la conversion des lettres au phonèmes.

Les consonnes :

```
(lts.ruleset
 ...
       ;;;;Consonant
         ( [ b ] = b );;ب
         ( [ t ] = t );;ت
         ( [ ^ ] = ^ );;ث
       ...
         ( [ $ ] = ch );;ش
         ( [ S ] = S );;ص
         ( [ D ] = D );;ض
       ...
         ( [ h ] = h );;ه
         ( [ w ] = w );;و
         ( [ y ] = y );;ي
       ...
         ( [ "\'" ] = ah );;ء
         ( [ } ] = ah );;ئ

         ( [ p ] # = );;ة FIN
         ( [ p ] = t ) ;;ة
 ...
)
```

Les voyelles :

```
(lts.ruleset
 ...
         ;;;;vowel+(ق,ظ,ط,ض,ص)
         ( [ a ] S = A )
         ( [ a ] D = A )
         ( [ a ] T = A )
         ( [ a ] Z = A )
         ( [ a ] q = A )
         ( [ i ] S = I )
         ( [ i ] D = I )
         ( [ i ] T = I )
         ( [ i ] Z = I )
         ( [ i ] q = I )
         ( [ u ] S = U )
         ( [ u ] D = U )
         ( [ u ] T = U )
         ( [ u ] Z = U )
         ( [ u ] q = U )
         ;;;;vowel
         ( [ a ] = a );;فتحة
         ( [ i ] = i );;كسرة
```

```
        ( [ u ] = u );;ضمة
        ( [ o ] = );;سكون
 ...
)
```

Les règles composées :

```
(lts.ruleset
 ...
       ;;;; ال الشمسية في أول الكلمة
       ...
         ( # [ A l ] d = ah a );;الدلو
       ...
       ;;;; ال الشمسية في وسط الكلمة
       ...
         ( [ a A l ] s = a );;فَالسيف
         ( [ i A l ] n = i );;بِالنار
       ...
       ;;;; ال القمرية في أول الكلمة
       ...
         ( # [ A l o ] x = ah a l );;الخبز
         ( # [ A l o ] E = ah a l );;العمل
       ...
       ;;;; ال القمرية في وسط الكلمة
       ...
         ( [ a A l o ] q = a l );;كَالقمر
         ( [ i A l o ] w = i l );;بِالورد
       ...
       ;;;;الشدة
       ...
         ( [ j ~ ] = jj );;رجَّح
         ( [ y ~ ] = yy );;إيَّاك
       ...
       ;;;;السكون
       ...
         ( [ ^ o ] = ^ );;مثلثْ
         ( [ s o ] = s );;مسلمْ
       ...
       ;;;;التنوين
         ( [ F ] = a n );;فتحتان
         ( [ N ] = ua n );;ضمتان
         ( [ K ] = ia n );;كسرتان
       ...
 ...
)
```

5.3.1.4 Exécution

Après avoir terminé la configuration, toutes les propriétés linguistiques et les règles nécessaires pour la transcription phonétique du texte sont maintenant décrites, afin d'obtenir les fichiers Utterance.

En appliquant les 3 commandes suivantes :
./bin/do_build build_prompts
./bin/do_build label
./bin/do_build build_utts
Les fichiers .utt devraient maintenant se trouver sous le répertoire : festival /utts.[43]

5.3.2 La production de la voix par HTS

Premièrement, il est important de souligner que, pour utiliser le système HTS, il est nécessaire d'avoir un corpus avec des phrases enregistrées au format brut (.raw), et correctement transcrites dans des fichiers Utterance produits dans l'analyse de texte. Plus le corpus est grand, plus les modèles seront précis et la voix synthétique sera générée avec une qualité supérieure.

Le programme va extraire les paramètres de la base de données, afin de construire des HMM. En plus des enregistrements et des fichiers de transcription, des questions permettant l'interprétation des étiquettes et la construction d'arbres de décision sont également requises. (figure 5.5)

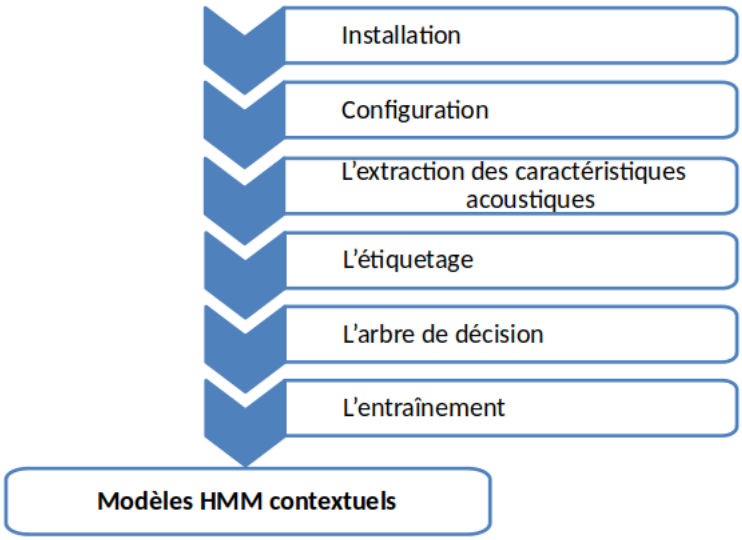


FIGURE 5.5 : Production de la voix par HTS.

Le processus de formation est lent. Sur un Core i5, cadencé à 2,4 GHz, il prend environ huit heures (avec 902 phrases dans la base de données et une configuration des paramètres par défaut).

5.3.2.1 Installation

Une séquence pas à pas des commandes nécessaires pour l'installation complète de HTS sous Linux est présentée. Afin d'utiliser HTS, d'autres programmes doivent être installés. Ils sont open source et peuvent être téléchargés à partir des adresses en ligne suivantes [39] :

- HTK 3.4.1 et HDecode 3.4.1 : http ://htk.eng.cam.ac.uk/
- HTS 2.3 patch for HTK 3.4.1 : http ://hts.sp.nitech.ac.jp/ ?Download
- HTS Engine API 1.1 : http ://hts-engine.sourceforge.net/
- SPTK 3.11 : http ://sp-tk.sourceforge.net
- HTS Demo 2.3 : http ://hts.sp.nitech.ac.jp/ ?Download

Il est conseillé d'utiliser le système d'exploitation Linux pour exécuter HTS. De plus, il est souhaitable de disposer d'au moins 20 Go d'espace libre sur le disque. Une fois les HMM créés, l'étape de synthèse peut être effectuée sur un ordinateur modeste, car cette partie nécessite peu de mémoire et une faible capacité de traitement.[39]

Après le téléchargement des programmes, nous devons exécuter les commandes suivantes [42] :

⇒ tar xvf HTK-3.4.1.tar.gz
tar xvf HDecode-3.4.1.tar.gz
mkdir hts_patch
mv HTS-2.3_for_HTK-3.4.1.tar.bz2 hts_patch/
cd hts_patch/
tar xvf HTS-2.3_for_HTK-3.4.1.tar.bz2
cd ../htk/
patch -p1 -d . < ../hts_patch/HTS-2.3_for_HTK-3.4.1.patch
./configure
make
sudo make install
make hlmtools ; sudo make install-hlmtools

make hdecode ; sudo make install-hdecode

⇒ tar xvf hts_engine_API-1.10.tar.gz
cd hts_engine_API-1.10/
./configure
make
sudo make install

⇒ tar xvf SPTK-3.11.tar.gz
cd SPTK-3.11/
./configure
make
sudo make install

L'étape suivante consiste à exécuter la démo. Mais d'abord, il faut s'assurer que le logiciel "TclTk 8.4", disponible sur http ://www.activestate.com/products/tcl/, est installé. De plus, la présence de programme Festival installé dans l'analyse de texte est requise.

5.3.2.2 Configuration

Maintenant que tous les programmes nécessaires sont installés, l'étape suivante consiste à indiquer leurs chemins en utilisant la configuration décrite dans le fichier INSTALL de la démo HTS.(figure 5.6)

⇒ ./configure \
–with-tcl-search-path=/usr/local/ActiveTcl/bin \
–with-fest-search-path=/home/HTS/festival/examples \
–with-sptk-search-path=/usr/local/SPTK/bin \
–with-hts-search-path=/usr/local/HTS-2.3/bin \
–with-hts-engine-search-path=/usr/local/bin \

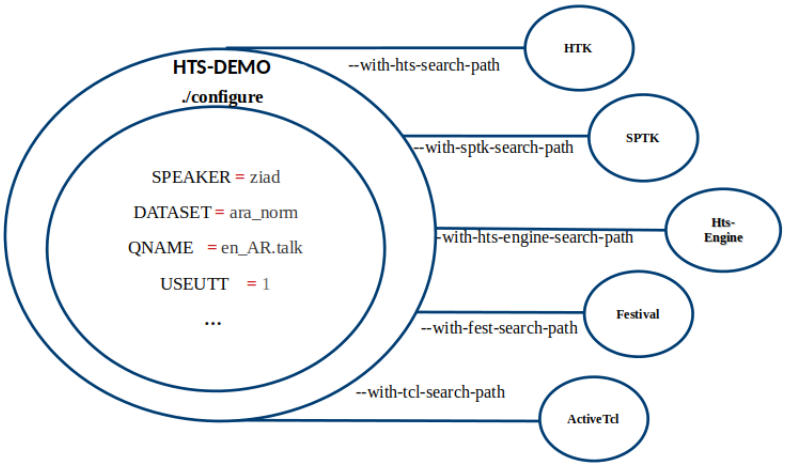


FIGURE 5.6 : Configuration de HTS.

Les variables pouvant être configurées sont mentionnées dans les scripts d'installation.

```
# setting
SPEAKER = ziad
DATASET = ara_norm
QNAME = en_AR.talk
...
# MATLAB and STRAIGHT
USESTRAIGHT = 0
MATLAB = -nodisplay -nosplash -nojvm
STRAIGHT =
# DNN
USEDNN = 0

# Festival commands
USEUTT = 1
...
# speech analysis conditions
SAMPFREQ = 48000 # Sampling frequency (48kHz)
FRAMELEN = 1200 # Frame length in point (1200 = 48000 x 0.025)
FRAMESHIFT = 240 # Frame shift in point (240 = 48000 x 0.005)
WINDOWTYPE = 1 # Window type -> 0: Blackman 1: Hamming 2: Hanning
NORMALIZE = 1 # Normalization -> 0: none 1: by power 2: by magnitude
FFTLEN = 2048 # FFT length in point
FREQWARP = 0.55 # frequency warping facto
GAMMA = 0 # pole/zero weight for mel-generalized cepstral (MGC)
MGCORDER = 34 # order of MGC analysis
...
```

Par exemple, pour modifier l'ordre de filtrage à 50 et le nombre d'états HMM à 7, l'instruction suivante doit être fournie :

⇒ ./configure MGCORDER=50 NSTATE=7

Une fois la configuration terminée et tous les paramètres choisis, nous pouvons lancer l'extraction des données nécessaires pour l'entraînement :

⇒ make

5.3.2.3 L'extraction des caractéristiques acoustiques

Elles sont extraites du signal audio brut (figure 5.7), celles-ci incluent une représentation de la hauteur, les propriétés spectrales de l'audio et la composition entre ces deux caractéristiques [43]. Les fichiers audios doivent être présentés dans le chemin : data/raw/

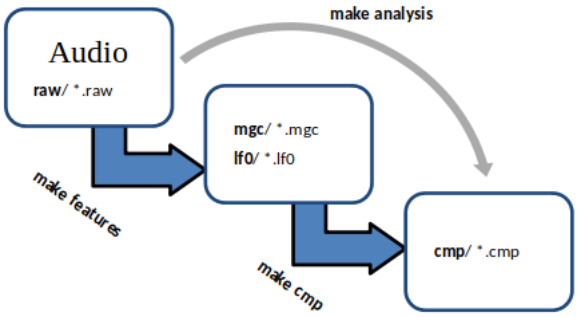


FIGURE 5.7 : Extraction des caractéristiques acoustiques.

Ce processus prend du temps selon le programme utilisé (environ 1 heure si nous utilisons SPTK avec 900 fichiers audios), nous pouvons le lancer par la commande :

⇒ make analysis

Ces paramètres sont regroupés dans des fichiers avec l'extension cmp, ils peuvent être lus en utilisant la commande Hlist de HTK avec la syntaxe suivante [39] :

⇒ /HList -C /data/configs/hlist.conf /data/cmp/nom_de_fichier.cmp

5.3.2.4 L'étiquetage

Dans cette étape, les caractéristiques linguistiques seront représentées par des fichiers de format (.lab) qui contiennent non seulement la transcription phonétique du texte, mais également les informations prosodiques et contextuelles de chaque phonème, syllabe, mot, phrase et énoncé.

Le processus a été automatisé en analysant d'abord les fichiers .utt par le système de synthèse vocale Festival à l'aide de la fonction 'dumpfeats', puis en construisant les fichiers d'étiquettes dépendantes suivant le format d'étiquette de HTS. Les fichiers Utterance doivent être présentés dans le chemin : data/utts/

Cette partie nécessite une petite modification dans le programme Festival, qui consiste à définir le fichier 'radioPhone.scm' qui est l'équivalent de 'phoneset.scm' déjà mentionné dans le traitement linguistique. Par la suite, nous pouvons lancer la production des étiquettes en appliquant la commande :

⇒ make lab

Les résultats vont être présentés dans les deux types de fichiers suivants (figure 5.8) :

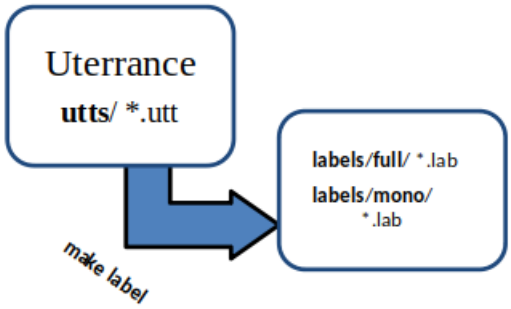


FIGURE 5.8 : Étiquetage.

a- Fullcontexte :

Chaque fichier Utterance est associé à un fichier d'étiquette correspondant qui contient des informations sur le contexte phonétique du signal telles que la durée, le phonème actuel, les phonèmes précédents et suivants, ainsi que les informations prosodiques telles que le nombre de mots dans la phrase, le nombre de syllabes dans le mot, etc.[8]

Les fichiers étiquetés de tous les énoncés de la base de données vocales se trouvent dans le dossier : data/labels/full

```
0 0 xx^xx-pau+w=a@xx_xx/A:xx_xx_xx/B:xx-xx-xx@xx-xx&xx-xx#xx-xx$xx-xx!xx-xx;xx
    ↪ -xx|xx/C:0+0+3/D:xx_xx/E:xx+xx@xx+xx&xx+xx#xx+xx/F:content_2/G:xx_xx/H:
    ↪ xx=xx^xx=xx|xx/I:62=19/J:62+19-1
0 2200000 xx^pau-w+a=f@1_3/A:xx_xx_xx/B:0-0-3@1-2&1-62#0-0$0-0!xx-xx;xx-xx|a/C
    ↪ :0+0+1/D:xx_xx/E:content+2@1+19&0+18#xx+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
```

```
2200000 3300000 pau^w-a+f=ii@2_2/A:xx_xx_xx/B:0-0-3@1-2&1-62#0-0$0-0!xx-xx;xx-
    ↪ xx|a/C:0+0+1/D:xx_xx/E:content+2@1+19&0+18#xx+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
3300000 4400000 w^a-f+ii=ah@3_1/A:xx_xx_xx/B:0-0-3@1-2&1-62#0-0$0-0!xx-xx;xx-
    ↪ xx|a/C:0+0+1/D:xx_xx/E:content+2@1+19&0+18#xx+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
4400000 5500000 a^f-ii+ah=a@1_1/A:0_0_3/B:0-0-1@2-1&2-61#0-0$0-0!xx-xx;xx-xx|
    ↪ ii/C:0+0+1/D:xx_xx/E:content+2@1+19&0+18#xx+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
5500000 6600000 f^ii-ah+a=chh@1_1/A:0_0_1/B:0-0-1@1-4&3-60#0-0$0-0!xx-xx;xx-xx
    ↪ |novowel/C:0+0+2/D:content_2/E:content+4@2+18&1+17#1+1/F:content_4/G:xx
    ↪ _xx/H:62=19^1=1|NONE/I:xx=xx/J:62+19-1
6600000 7700000 ii^ah-a+chh=a@1_2/A:0_0_1/B:0-0-2@2-3&4-59#0-0$0-0!xx-xx;xx-xx
    ↪ |a/C:0+0+3/D:content_2/E:content+4@2+18&1+17#1+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
7700000 8800000 ah^a-chh+a=w@2_1/A:0_0_1/B:0-0-2@2-3&4-59#0-0$0-0!xx-xx;xx-xx|
    ↪ a/C:0+0+3/D:content_2/E:content+4@2+18&1+17#1+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
8800000 9900000 a^chh-a+w=T@1_3/A:0_0_2/B:0-0-3@3-2&5-58#0-0$0-0!xx-xx;xx-xx|a
    ↪ /C:0+0+1/D:content_2/E:content+4@2+18&1+17#1+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
9900000 11000000 chh^a-w+T=i@2_2/A:0_0_2/B:0-0-3@3-2&5-58#0-0$0-0!xx-xx;xx-xx|
    ↪ a/C:0+0+1/D:content_2/E:content+4@2+18&1+17#1+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
11000000 12100000 a^w-T+i=ah@3_1/A:0_0_2/B:0-0-3@3-2&5-58#0-0$0-0!xx-xx;xx-xx|
    ↪ a/C:0+0+1/D:content_2/E:content+4@2+18&1+17#1+1/F:content_4/G:xx_xx/H
    ↪ :62=19^1=1|NONE/I:xx=xx/J:62+19-1
```

b- Monophone :

Chaque fichier Utterance sera combiné avec un fichier d'étiquette de Monophone contenant une liste de phonèmes, ainsi que leur horodatage.[8]

Ces fichiers se trouvent dans : data/labels/mono

```
       0 2200000 w
 2200000 3300000 a
 3300000 4400000 f
 4400000 5500000 ii
 5500000 6600000 ah
 6600000 7700000 a
 7700000 8800000 chh
 8800000 9900000 a
 9900000 11000000 w
11000000 12100000 T
12100000 13200001 i
```

Les deux exemples ci-dessus sont l'étiquetage de la phrase ("وَفِي الشَّوْطِ").

Les fichiers d'étiquettes que nous souhaitons synthétiser doivent être présentés dans le chemin : data/labels/gen

En utilisant les étiquettes et les caractéristiques acoustiques, des listes seront crées pour l'entraînement en appliquant la commande :

⇒ make mlf list scp

Cependant il est impossible d'envisager tous les contextes de chaque phonème car la base de données est limitée. Par conséquent, l'arbre de décision et le clustering sont utilisés pour permettre la synthèse des phonèmes dans des contextes qui n'apparaissent pas dans le corpus de formation.

5.3.2.5 Nœuds de l'arbre de décision

L'arbre de décision est une structure binaire basée sur des étiquettes et des questions relatives aux phonèmes. Le fichier que nous devons adapter aux phonèmes utilisés dans la voix arabe est présent dans le chemin : data/configs/en_AR.talk.conf

Chaque question est définie comme suit :

QS "Description de la question" Ensemble des phonèmes

```
QS "LL-Vowel" {a^,aa^,A^,AA^,i^,ia^,ii^,I^,II^,u^,uu^,U^,UU^,ua^,uua^}
QS "LL-Consonant" {ah^,ahh^,b^,bb^,t^,tt^,^^,^^^,j^,jj^,H^,HH^,x^,xx^,d^,dd^,
    ↪ th^,thh^,r^,rr^,z^,zz^,s^,ss^,ch^,chh^,S^,SS^,D^,DD^,T^,TT^,Z^,ZZ^,E^,
    ↪ EE^,g^,gg^,f^,ff^,q^,qq^,l^,ll^,m^,mm^,n^,nn^,h^,hh^,w^,ww^,y^,yy^,v^}
...
QS "L-Stop" {^b-,^bb-,^D-,^DD-,^d-,^dd-,^T-,^TT-,^t-,^tt-,^k-,^kk-,^q-,^qq-,^
    ↪ ah-,^ahh-}
QS "L-Nasal" {^m-,^mm-,^n-,^nn-}
...
QS "C-Long_Vowel" {-aa+,-AA+,-ii+,-II+,-uu+,-UU+,-uua+}
QS "C-Short_Vowel" {-a+,-A+,-i+,-ia+,-I+,-u+,-U+,-ua+}…

QS "R-No_Continuent" {+b=,+bb=,+D=,+DD=,+d=,+dd=,+T=,+TT=,+t=,+tt=,+k=,+kk=,+q
    ↪ =,+qq=,+ah=,+ahh=,+j=,+jj=}
QS "R-Positive_Strident" {+j=,+jj=,+z=,+zz=,+Z=,+ZZ=,+S=,+SS=,+s=,+ss=,+ch=,+
    ↪ chh=}
...
QS "RR-Unvoiced_Consonant" {=T@,=TT@,=t@,=tt@,=k@,=kk@,=q@,=qq@,=ah@,=ahh@,=S@
    ↪ ,=SS@,=s@,=ss@,=ch@,=chh@,=f@,=ff@,=^@,=^^@,=H@,=HH@,=x@,=xx@,=h@,=hh@}
QS "RR-Front_Consonant" {=b@,=bb@,=m@,=mm@,=w@,=ww@,=v@,=f@,=ff@}
```

Pour l'entraînement, il est nécessaire de spécifier les fichiers de questions qui sont générés à la base du fichier 'en_AR.talk.conf' par la commande (figure 5.9) :

⇒ make question

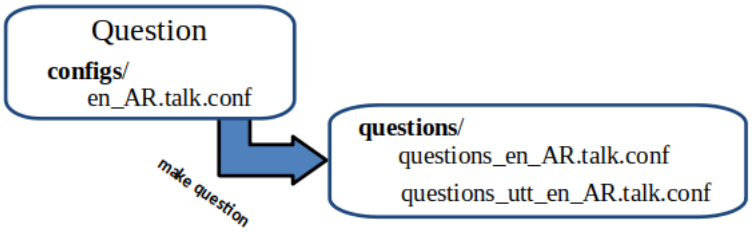


FIGURE 5.9 : Création des questions.

Ces fichiers sont présents dans le chemin data/questions, et doivent contenir autant de questions nécessaires pour couvrir tous les cas possibles selon les étiquettes contextuelles.

La fonction HHEd de HTK permet de générer l'arbre de décision à partir des questions. Pour générer une voix de qualité moyenne, certaines questions sont nécessaires afin d'évaluer le stress, la position relative et le nombre d'occurrences du phonèmes donné.[39]

Les trois arbres générés par la formation des modèles se trouvent dans les chemins suivants :

"Duration tree" : /trees/ver1/dur

"Pitch and spectrum trees" : /trees/ver1/cmp

5.3.2.6 L'entraînement

Après l'installation et la configuration du HTS, et une fois que les paramètres acoustiques, les fichiers d'étiquettes, les listes et les questions sont connus, nous pouvons commencer la détermination des HMM à travers plusieurs algorithmes d'estimation statistique (figure 5.10).

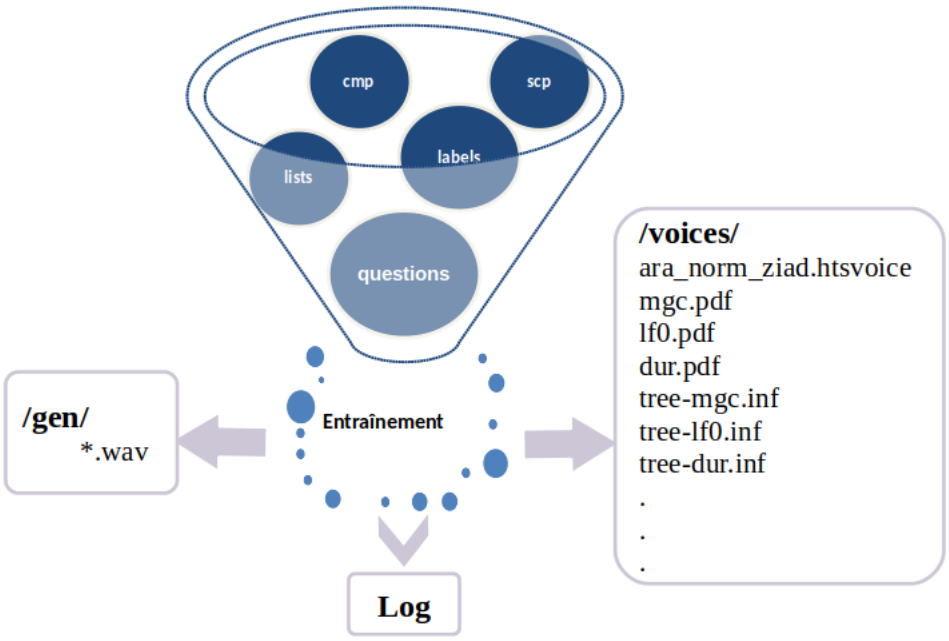


FIGURE 5.10 : La formation vocale.

Malgré leur complexité, les traitements phonétiques, prosodiques et linguistiques sont nécessaires pour effectuer de telles analyses de manière automatique.

L'étape de formation est effectuée par le script :

⇒ /usr/bin/perl ./scripts/Training.pl ./scripts/Config.pm > log 2>1

Cela génère d'abord des HMM indépendants du contexte, puis, en utilisant des méthodes d'estimation statistique et des algorithmes Viterbi et Baum-Welch, le script génère des HMM contextuels.[38]

Enfin, les fichiers suivants sont créés, mgc.pdf, lf0.pdf et dur.pdf. Ils contiennent respectivement les fonctions de densité et de probabilité des HMM contextuels (probability density function) pour le spectre, la fréquence de hauteur et la durée de l'état. Ces fichiers, ainsi que les fichiers d'arbre de décision tree-mgc.inf (Arbre du spectre), tree-lf0.inf (Arbre de fréquence de la hauteur) et tree-dur.inf (Arbre de durée d'état) permettent des performances de synthèse vocale, en utilisant l'API de son moteur.

Les paramètres et les étapes de l'entraînement sont mentionnés dans le fichier : /scripts/Config.pm

```
# Switch
$MKENV = 1; # preparing environments
$HCMPV = 1; # computing a global variance
```

```
$IN_RE = 1; # initialization & reestimation
$MMMMF = 1; # making a monophone mmf
$ERST0 = 1; # embedded reestimation (monophone)
$MN2FL = 1; # copying monophone mmf to fullcontext one
$ERST1 = 1; # embedded reestimation (fullcontext)
$CXCL1 = 1; # tree-based context clustering
$ERST2 = 1; # embedded reestimation (clustered)
$UNTIE = 1; # untying the parameter sharing structure
$ERST3 = 1; # embedded reestimation (untied)
$CXCL2 = 1; # tree-based context clustering
$ERST4 = 1; # embedded reestimation (re-clustered)
$FALGN = 1; # forced alignment
$MCDGV = 1; # making global variance
$MKUNG = 1; # making unseen models (GV)
$MSPF1 = 1; # training modulation spectrum-based postfilter (1mix)
$MKUN1 = 1; # making unseen models (1mix)
$PGEN1 = 1; # generating speech parameter sequences (1mix)
$WGEN1 = 1; # synthesizing waveforms (1mix)
$CONVM = 1; # converting mmfs to the hts_engine file format
$ENGIN = 1; # synthesizing waveforms using hts_engine
$MKDAT = 1; # making training data for deep neural network
$TRDNN = 1; # training a deep neural network
$MSPFD = 1; # training modulation spectrum-based postfilter (dnn)
$PGEND = 1; # generating speech parameter sequences (dnn)
$WGEND = 1; # synthesizing waveforms (dnn)
$TRJGV = 1; # trajectory training considering global variance
$MSPFT = 1; # training modulation spectrum-based postfilter (trj)
$PGENT = 1; # generating speech parameter sequences (trj)
$WGENT = 1; # synthesizing waveforms (trj)
$SEMIT = 1; # semi-tied covariance matrices
$MKUNS = 1; # making unseen models (stc)
$PGENS = 1; # generating speech parameter sequences (stc)
$WGENS = 1; # synthesizing waveforms (stc)
$UPMIX = 1; # increasing the number of mixture components (1mix -> 2mix)
$ERST5 = 1; # embedded reestimation (2mix)
$MKUN2 = 1; # making unseen models (2mix)
$PGEN2 = 1; # generating speech parameter sequences (2mix)
$WGEN2 = 1; # synthesizing waveforms (2mix)
```

Nous pouvons suivre la progression du processus dans le fichier Log, qui se trouve dans le répertoire principal. Ce fichier est constamment mis à jour au cours de la procédure.

En cas d'erreur, la première action à entreprendre est de consulter le fichier Log. Il fournit le code d'erreur et il est possible d'identifier à quel moment l'exécution

du programme a été abandonnée.

Le contenu des fichiers avec l'extension .inf peut être visualisé à l'aide d'un éditeur de texte commun. Les fichiers .pdf sont au format binaire et peuvent être lus en utilisant la commande suivante de SPTK [39] :

⇒ /swab /voices/ver1/nom_de_fichier.pdf | /dmp +i|less -> header

⇒ /swab /voices/ver1/nom_de_fichier.pdf | /dmp +i|less -> parameters

L'étape d'entraînement ne doit être effectuée qu'une seule fois, car dès que les HMM dépendants du contexte sont connus, il devient possible de synthétiser la parole à partir de n'importe quelle entrée de texte.[39][38]

La synthétisation peut être subdivisée en quatre parties :

1. La saisie de texte doit être transcrite phonétiquement et correctement étiquetée. Par la suite, il est possible de définir la séquence de modèles HMM contextuels qui décrivent l'énoncé.
2. Les durées d'état HMM sont définies statistiquement avec les données des fichiers dur.pdf et tree-dur.inf.
3. Des paramètres spectraux et d'excitation sont générés pour chaque phonème du texte.
4. La forme d'onde est synthétisée.

Pour tester la qualité de la voix, nous pouvons écouter les fichiers synthétisés qui se trouvent dans le répertoire : / gen/.

5.3.3 L'intégration de la voix créée au système de synthèse vocale Festival

Festival peut être installé en appliquant la commande :

⇒ sudo apt install festival

Ce logiciel contient des voix anglaises intégrées par défaut et d'autre bibliothèques pour la synthétisation, contrairement au logiciel utilisé dans la création de la voix.

Afin de synthétiser une nouvelle voix par le système Festival, il existe plusieurs façons pour la présenter selon la méthode utilisée.

Dans notre projet, nous allons synthétiser la voix arabe via les modèles HMM générés. Cela nécessite deux types de fichiers :

- Festvox : Ce sont les fichiers déjà configurés pour le traitement linguistique. Afin de synthétiser un texte écrit en langue arabe, il doit être d'abord converti en phonèmes. Les fichiers Scheme doivent être présentés dans un dossier nommé 'festvox', avec certaines modifications dans les chemins utilisés et un scripts pour les lier entre eux.
- HTS : Ce sont les fichiers générés par HTS dans l'entraînement, ils décrivent toutes les caractéristiques vocales de la voix. Ils se trouvent dans le répertoire "/voices", et incluent des fichiers avec l'extension .inf, .pdf, .win et .htsvoice.

Une fois que la voix est ajoutée, les utilisateurs peuvent synthétiser n'importe quel texte écrit en arabe en appliquant les commande suivantes :

⇒ festival
festival > (voice_nom_de_la_voix) # sélectionner la voix utilisée.
festival > (SayText " texte arabe ") # synthétiser le text.
ou
(tts " chemin d'un fichier qui contient un texte arabe " nil)

5.3.4 L'amélioration de la qualité vocale générée

La voix synthétisée peut toujours être améliorée en fonction de la naturalité et l'intelligibilité en comparant entre les résultats obtenus et les résultats voulus. Cela nécessite une connaissance approfondie de toutes les règles de la langue arabe, ainsi que des outils avancés pour l'extraction des caractéristiques acoustiques. Cela nécessite aussi des arbres de décision plus précis et un corpus avec une voix très claire et des phonèmes bien articulés.

5.4 Applications

Une fois que l'ordinateur est capable de communiquer verbalement, plusieurs applications intéressantes peuvent être développées, telles que l'utilisation d'ordinateurs pour les malvoyants, la lecture de courriels et de textes, les systèmes automatisés dans les centres de services et les applications de communication mobile pour les personnes handicapées.

5.5 Conclusion

Dans ce chapitre, nous avons présenté notre travail d'un point de vue technique, en déterminant les étapes de la création d'une voix arabe synthétique.

Nous avons d'abord introduit les outils nécessaires utilisés, puis présenté par un schéma globale, les démarches de la réalisation du projet. Ensuite, nous avons détaillé toutes ces démarches en commençant par le traitement linguistique, la production de la voix avec les modèles de Markov cachés, et son intégration au système Festival, et en terminant par une brève explication de la procédure d'amélioration et quelques applications du système.

Le chapitre suivant aborde la méthodologie du test et ses résultats, en comparant la voix construite avec d'autres voix arabes open-source pour l'évaluer, afin d'améliorer la qualité et d'atteindre les résultats voulus.

Chapitre 6

Test et évaluation

6.1 Introduction

La parole synthétisée est essentielle pour les programmes d'IHM vocale, et pour aider les personnes à lire des textes, utiliser l'ordinateur et communiquer avec les autres. De nombreuses personnes ont un revenu faible ou nul, ils n'ont donc pas accès à cette technologie, ainsi, une alternative open-source est nécessaire.

Nous avons utilisé le système HTS pour créer une voix arabe synthétique open-source proche de la voix humaine.

Cette voix est construite et intégrée dans le système de synthèse vocale Festival. Nous allons la tester avec d'autres voix arabes pour l'évaluer, afin d'améliorer la qualité et d'atteindre les résultats voulus qui répondent mieux aux besoins d'utilisation.

Dans ce chapitre, nous allons présenter la méthodologie du test, puis ses résultats.
Le test a été effectué à l'aide d'un questionnaire en ligne en comparant quatre voix arabes, y compris celle que nous avons construite. La comparaison sera faite selon l'identification des mots et la qualité de la voix. Nous terminerons par mentionner quelques commentaires constructifs qui montrent l'importance du projet.

6.2 Méthodologie

Pour le test, nous avons choisi dix phrases d'une histoire internationale traduite en arabe (بائعة الكبريت), et nous les avons synthétisées en utilisant les voix arabes open-source suivantes :

- Festival-ziad (notre voix) : Elle est créée en utilisant le système HTS basé sur les chaînes de Markov cachées, et synthétisée par le système de synthèse vocale Festival.
- Tacotron-ar (projet de Youssef Sharief[1]) : Elle est créée par le système Tacotron basé sur les réseaux de neurones.
- ESpeak-ar (projet du Dr.Taha Zerrouki[2]) : Elles est créée par le système de synthèse vocale eSpeak basé sur la synthèse par règle.
- Festival-asrajeh (projet de Abdullah Alrajeh[3]) : Elle est créée par la méthode statistique "clustergen", et synthétisée par Festival.

Nous n'avons pas fait la comparaison avec les voix commerciales pour les raisons suivantes :

- Notre travail a un but académique.
- La qualité des voix commerciales est relativement supérieur par rapport aux voix open-source.
- Les systèmes de synthèse vocale commerciaux ne sont pas documentés, ils ne sont donc pas favorables avec l'objectif des systèmes open source.
- Pour tester les voix commerciales il faut les acheter.

La qualité vocale dépend de la personne qui la juge. Elle est donc une notion complexe à définir du fait de sa forte subjectivité. Cela dépend de l'interprétation de chaque audio donné.

MOS "Mean Opinion Score" est une mesure de qualité subjective largement utilisée pour évaluer les performances des systèmes de la parole. Chaque auditeur évalue la qualité du signal reconstruit selon une échelle prédéterminée. Ensuite, les scores sont moyennés pour déterminer la valeur finale de l'évaluation.[45]

Nous avons utilisé la méthode MOS qui est basée sur l'évaluation humaine en raison de la difficulté d'établir une évaluation objective par manque de moyen d'évaluation métrique précis.

[1]https ://github.com/youssefsharief/arabic-tacotron-tts
[2]https ://github.com/linuxscout/espeak-ng
[3]https ://github.com/asrajeh/arabic-tts/

La création du questionnaire[4] en ligne est faite à l'aide de "Google Forms". Nous avons choisi deux types de tests :

- Le test d'identification : Le pourcentage de mots identifiés.
- Le test de qualité : La clarté et la proximité d'une voix naturelle humaine.

Après chaque écoute, l'auditeur donne une note sanctionnant la qualité qu'il a perçue. Dans chaque page il y a une phrase synthétisée par les quatre voix mentionnées ci-dessus. Pour éviter un test lent et ennuyeux, celui-ci a été divisé en trois partie, afin que les testeurs puissent s'arrêter dans n'importe quelle section. Le questionnaire ne comporte pas de questions relatives au niveau d'expertise des participants. Il s'adresse à toutes les catégories sociales.

Le formulaire commence par une introduction, puis les questions suivantes :

1. L'arabe est-il votre langue maternelle ? (oui/non).
2. Êtes-vous ? (voyant, non voyant, autre).
3. Évaluation des fichiers audios (Échelle de notation sur 10).
4. Si vous avez un commentaire laissez-le ici.
5. Si vous voulez rester à jour, laissez votre e-mail.

L'idée que les participants laissent leurs commentaires et leurs e-mails permet de découvrir l'importance de notre projet et son impact sur la société. Cela nous permet également de rencontrer des personnes intéressées par ce projet, et même celles qui peuvent contribuer à son développement.

Les voix sont anonymes. Le choix des fichiers audios dans chaque page se fait aléatoirement, afin d'effectuer un test fiable et crédible.

Les phrases utilisées dans le test :

يُحْكَى بِأَنَّهُ كَانَ هُنَاكَ فَتَاةٌ فَقِيرَةٌ جَمِيلَةٌ ذَاتِ شَعْرٍ أَشْقَرٍ طَوِيلٍ

خَرَجَتْ فِي لَيْلَةِ رَأْسِ السَّنَةِ الْمِيلَادِيَّةِ مِنْ أَجْلِ بَيْعِ الْكِبْرِيتِ

وَكَانَتْ اللَّيْلَةُ شَدِيدَةَ الْبُرودَةِ

حَيْثُ كَانَ الثَّلْجُ يَتَسَاقَطُ مِنَ السَّمَاءِ

[4]https ://forms.gle/XpGCj4cBjtvMF9NZ6

وَكَانَتْ الْفَتَاةُ تَرْتَجِفُ مِنْ شِدَّةِ الْبُرودَةِ

وَكَانَتْ لَا تَرْتَدِي أَيَّ شَيْءٍ فَوْقَ رَأْسِهَا

فَكَانَتْ حَبَّاتُ الثَّلْجِ تَتَسَاقَطُ فَوْقَهُ

وَكَانَتْ قَدْ فَقَدَتْ حِذَاءَهَا, وَأَصْبَحَتْ تَمْشِي حَافِيَةَ الْقَدَمَيْنِ

كَمَا كَانَتْ تَشْعُرُ بِالْجُوعِ الشَّدِيدِ

وَخَشِيَتْ الْعَوْدَةَ إِلَى الْبَيْتِ خَوْفًا مِنْ أَبِيهَا الَّذِي طَلَبَ مِنْهَا بِأَنْ تَبِيعَ أَعْوَادَ الْكِبْرِيتِ

6.3 Résultats

Question 1: L'arabe est-il votre langue maternelle.

- 95,20% ont répondu par oui.
- 4,80% ont répondu par non.

FIGURE 6.1 : Pourcentage de participants parlant l'arabe.

Question 2: Etes-vous ? (voyant, non voyant, autre).

- 98,6% ont répondu par voyant.
- 0,4% ont répondu par aveugle.
- 1% autre.

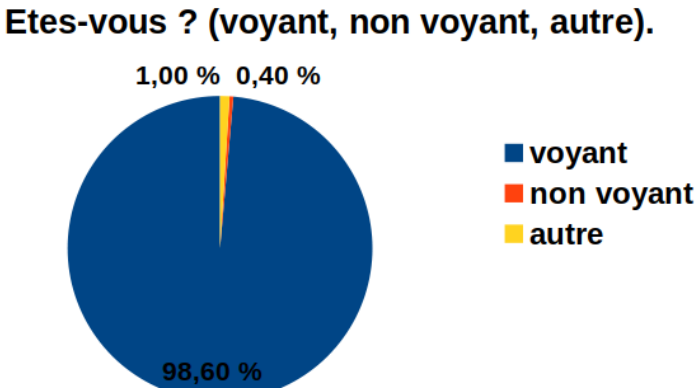


FIGURE 6.2 : Acuité visuelle des participants.

Question 3: Évaluation

Nombre de participants :

- Phrases 1,2,3 :290 réponses.
- Phrases 4,5,6 :89 réponses.
- Phrases 7,8,9,10 :28 réponses.

Test d'identification

La moyenne générale est donnée par :
Moyenne (voix) : $\bar{x} = \frac{\sum_1^n x_i \times f_i}{\sum_1^n f_i}$ où :
x_i : la moyenne de la phrase.
f_i : le nombre de participants qui évaluent la ième phrase.

TABLE 6.1 : Moyennes du test d'identification

Voix	1	2	3	4	5	6	7	8	9	10	Moy.
Festival-Ziad	9,18	8,7	8,9	8,92	9,13	9,21	8,82	9,33	9,14	9	8,97
Tacotron-ar	8,49	7,76	8,45	8,84	8,89	8,2	8,21	8,11	9,03	8,61	8,34
Espeak-ar	7,57	6,36	6,89	6,01	6,79	6,74	6,14	7,03	7,33	6,77	6,84
Festival-asraJih	9,1	8,45	8,66	8,11	9,16	9,2	9,21	9,21	9,14	8,57	8,78
Participants	290			89			28				

La moyenne la plus élevée est obtenue par la voix Festival-Ziad, suivi par Festival-asraJih, puis Tacotron-ar. Les trois premières voix ont des valeurs proches et élevées grâce à l'utilisation de méthodes statistiques paramétriques dans leur construction. En revanche, la voix Espeak-ar est en dernier au classement à cause de l'utilisation de la synthèse par règle qui affecte négativement le pourcentage de mots identifiés.

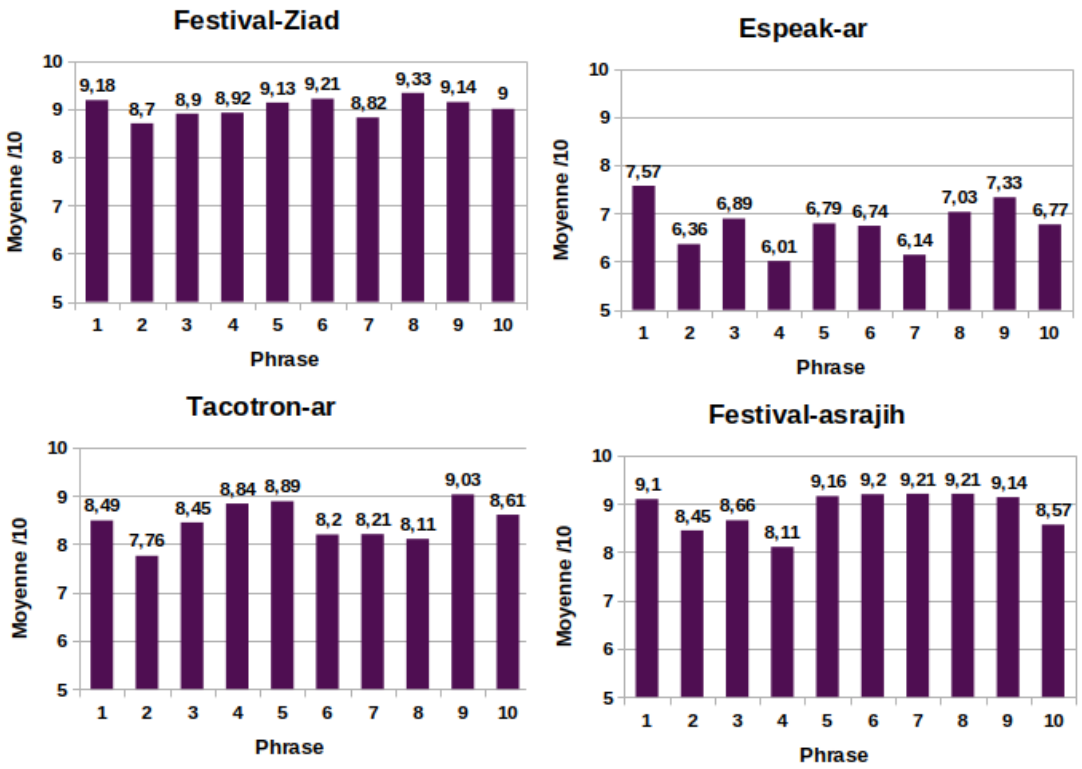


FIGURE 6.3 : Représentation graphique des moyennes du test d'identification.

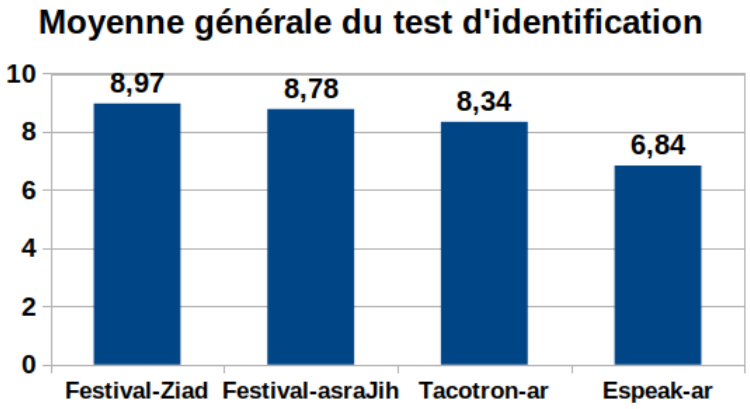


FIGURE 6.4 : La moyenne générale du test d'identification.

Test de qualité

La moyenne a été calculer de la même façon que le test d'identification.

TABLE 6.2 : Moyennes du test de qualité

Voix	1	2	3	4	5	6	7	8	9	10	Moy.
Festival-Ziad	7,46	6,54	7,13	7,13	7,69	7,97	7,14	7,77	7,67	7,44	7,20
Tacotron-ar	6,65	5,81	6,72	7,36	7,75	6,63	6,81	6,74	7,67	7,00	6,64
Espeak-ar	4,81	3,79	4,33	3,64	4,56	4,32	4,38	4,85	4,84	4,03	4,30
Festival-asraJih	7,49	6,88	7,19	5,94	7,61	7,13	7,89	7,96	7,68	7,86	7,18
Participants	290			89			28				

La moyenne la plus élevée dans ce test de qualité est obtenue par la voix Festival-Ziad, suivi par Festival-asraJih. Les deux premières voix ont des valeurs proches et élevées grâce à l'utilisation d'un corpus de la parole naturelle pour créer une voix synthétique proche de la voix humaine. La voix Tacotron-ar utilise aussi une base de donné de la parole naturelle, et a obtenu une moyenne qui n'est pas très faibles par rapport à celle de Festival. Finalement, la voix Espeak-ar est en dernier au classement relatif à la clarté et la proximité d'une voix naturelle, à cause de la parole robotique générée.

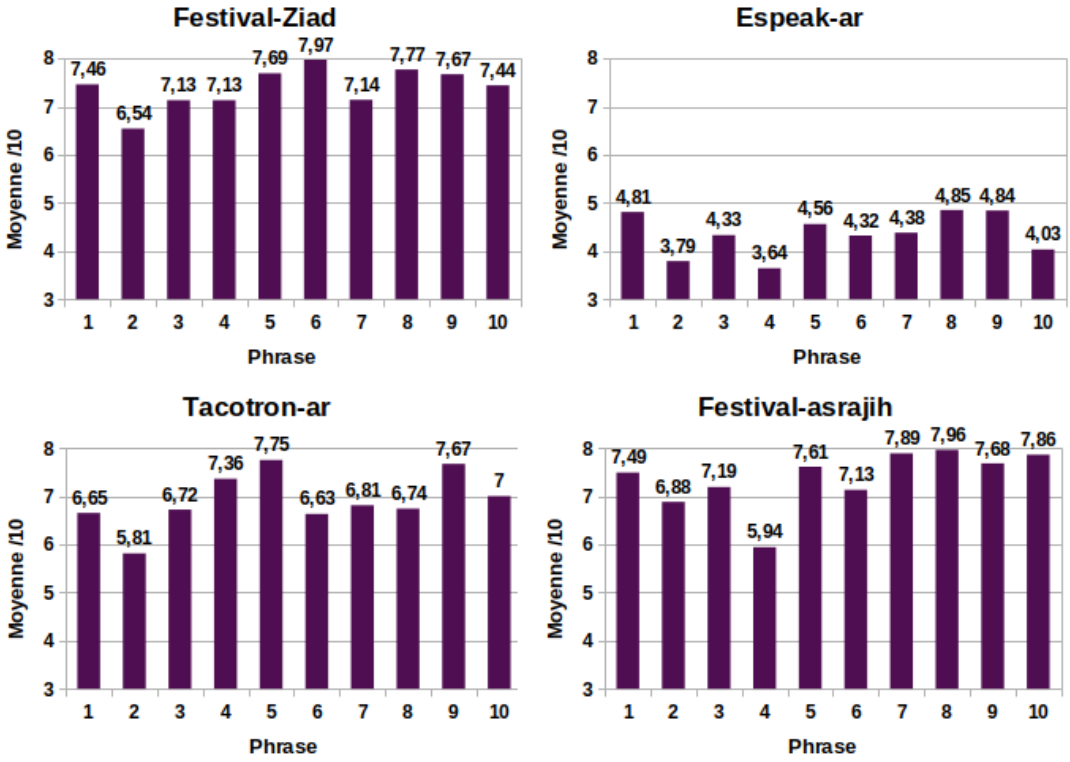


FIGURE 6.5 : Représentation graphique des moyennes du test de qualité.

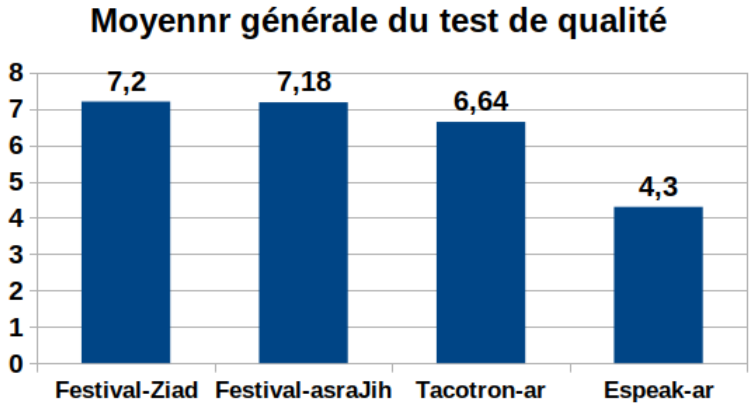


FIGURE 6.6 : La moyenne générale du test de qualité.

Question 4: Si vous avez un commentaire laissez-le ici.

Parmi les 90 commentaires, certains sont utiles et d'autres encourageants.

FIGURE 6.7 : Nombre de tests avec et sans commentaire.

En voici quelques aperçus :

- Ajouter une voix féminine.
- Conseils d'experts pour l'amélioration de la voix et l'enregistrement d'une autre.
- Un administrateur de Wikipédia nous a demandé d'essayer la voix sur des articles de l'encyclopédie.
- Des volontaires veulent participer par leurs propre voix.
- Quelques suggestions de projets similaires.
- Améliorer la vocalisation.

Question 5: Si vous voulez rester à jour, laissez votre email.

113 participants ont laissé leurs e-mails, y compris ceux qui nous ont contactés pour l'amélioration de la voix et l'intégration dans d'autres projets.

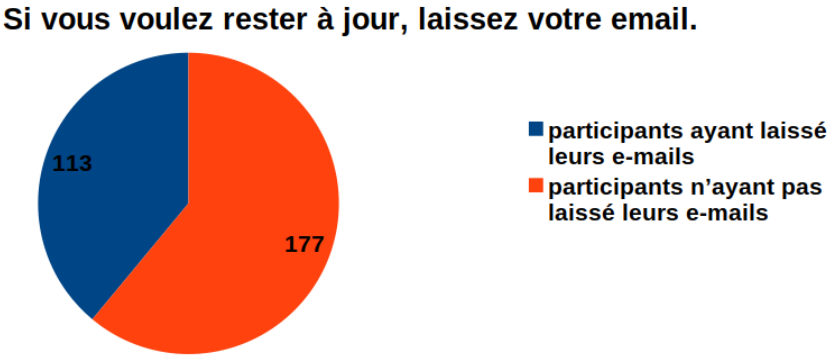


FIGURE 6.8 : Nombre d'e-mails reçus.

6.4 Discussion

Dans cette partie, nous avons testé différentes voix arabe open-source, y compris celle que nous avons construite. Les participants ont évalué les voix et donné une note selon la qualité des phrases synthétisées par les quatre systèmes sélectionnés.

Les voix synthétisées par Festival ont reçu la plus haute évaluation en terme d'identification et de qualité. La voix construite par HTS et la voix construite par la méthode statistique clustergen ont presque la même moyenne et chacune peut surpasser l'autre et s'améliorer pour donner des meilleurs résultats. Nous pouvons remarquer que l'évaluation de la voix de Tacotron est proche de celle de Festival, qui est une nouvelle technique et peut également donner de meilleurs résultats. En revanche, eSpeak génère une voix robotique loin d'être un bon choix pour les systèmes qui utilisent la synthétisation.

6.5 Conclusion

Dans ce chapitre, nous avons évalué la voix construite dans le projet, en la testant avec trois voix arabes open-source afin d'améliorer la qualité, et répondre mieux aux besoins d'utilisation.

Nous avons d'abord introduit la méthodologie du test, en mentionnant les systèmes à évaluer, la mesure de qualité (MOS) utilisé, les phrases synthétisées et

les points abordés dans le formulaire. Ensuite, nous avons présenté les résultats et les moyennes obtenus, en terminant par quelques commentaires de participants qui montrent l'importance du projet, et une brève discussion sur l'évaluation des systèmes.

Conclusion Générale

Dans ce projet, nous avons introduit le développement d'une voix arabe synthétique open source. Notre choix pour les technologies utilisées est basé sur deux facteurs, une technique de synthèse vocale pour une bonne qualité de la voix, et un système de synthèse vocale open source qui peut être adapté pour la langue arabe.

Nous avons mené une étude sur les quatre techniques existantes : articulatoire, par règle, par concaténation et statistique paramétrique. Parmi ces techniques, la plus appropriée était la méthode statistique paramétrique basée sur les modèles de Markov cachés. En ce qui concerne les systèmes, nous avons étudié quatre systèmes connus : eSpeak, Festival, Marry et Tacotron. Festival était le meilleure choix car il peut être utilisé avec des voix basées sur HMM implémentées par le système HTS. Nous avons également fait une étude sur la phonologie de la langue arabe, afin d'obtenir les informations phonologiques nécessaires pour l'implémentation de ce projet. Par la suite nous avons commencé l'implémentation de notre système par la définition, à l'aide de Festival, des scripts nécessaires à la création des caractéristiques linguistiques du corpus arabe crée par Nawar Halabi. Ces caractéristique sont utilisées pour produire la voix par le système HTS, en commençant par l'extraction des caractéristiques acoustiques des fichiers audios, puis l'étiquetage, la définition des nœuds de l'arbre de décision, en terminant par l'entraînement. Une fois la voix créée, nous l'avons intégrée dans le système de synthèse vocale Festival pour l'utilisation.

Cette voix n'est pas la première dans ce domaine, mais elle est naturelle, disponible pour le téléchargement et l'utilisation et ne nécessite pas de ressources puissantes. La version actuelle est principalement destinée aux développeurs et aux programmeurs, et elle peut être installée facilement. La voix arabe est un package indépendant, il suffit d'installer Festival, puis de mettre les fichiers à leur place. Cette version ouvrira la porte à de nombreuses applications qui nécessitent une sortie vocale. Elle peut être convertie à un format adapté au système Flite (travail en cour), qui est capable de fonctionner dans les environnements mobiles.

Nous avons rencontré quelques limites dans ce projet. Le manque de corpus de la parole arabe nous a empêché de construire d'autres voix. De plus, il n'est pas aisé d'utiliser ces systèmes en raison du manque de documentations suffisantes. Heureusement que nous avons travaillé sur le domaine d'open source, où ils existent quelques travaux similaires pour d'autres langues. En ce qui concerne l'entraînement, nous ne possédons pas de machine puissante, ce qui a augmenté le nombre d'heures pour ce processus (environs huit heurs).

D'autre part, nous avons réussit à créé une voix par HTS en utilisant le même corpus et un autre outil pour l'extraction des caractéristiques acoustiques (WORLD). La qualité de cette voix est supérieure à celle que nous avons présentée dans ce projet, mais malheureusement elle n'est pas compatible pour qu'elle soit utilisée dans Festival pour la synthétisation.

Nous projetons poursuivre et améliorer ce travail. Nous pensons essentiellement créer un corpus de la parole naturelle arabe, afin de générer d'autres voix différentes, y compris celle féminine, utiliser d'autres outils comme "hts-engine-api" et "flite", qui permettent l'utilisation de notre voix dans les environnements mobiles, chercher d'autres moyens pour l'extraction des caractéristiques acoustiques, afin de générer une voix de qualité meilleure et trouver une solution pour utiliser la voix que nous avons créée avec WORLD comme outil de traitement de signal vocal.

Bibliographie

[1] T. Dutoit, *An introduction to text-to-speech synthesis*, 1st ed. Dordrecht, Netherlands : Kluwer Academic Publishers, 1997.

[2] S. Lemmetty, "Review of speech synthesis technology," Master's thesis, Dep. of Electrical Engineering and Communications Technology University of Helsinki, Finlande, 1999.

[3] J. Benesty, M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*, 1st ed. Secaucus, New Jersy : Springer-Verlag, 2008.

[4] M. Assaf, "A prototype of an arabic diphone speech synthesizer in festival," Master's thesis, Dep. of Linguistics and Philology, Uppsala University, Sweden, 2005.

[5] P. Taylor, *Text-to-speech synthesis*, 1st ed. New York, USA : Cambridge University Press, 2009.

[6] T. Zerrouki, M. M. A. Shquier, A. Balla, N. Bousbia, I. Sakraoui, and F. Boudardara, "Adapting espeak to arabic language : converting arabic text to speech language using espeak," *International Journal of Reasoning-based Intelligent Systems*, vol. 11, no. 1, pp. 76–89, 2019.

[7] N. Halabi, "Modern standard arabic phonetics for speech synthesis," Ph.D. dissertation, University of Southampton, UK, 2016.

[8] D. Alabbad, A., "An investigation into approaches to text-to-speech synthesis for modern standard arabic," Ph.D. dissertation, Faculty of Science and Engineering, University of Manchester, UK, 2019.

[9] H. Xuedong, A. Alex, and H. Hsiao-whuen, *Spoken Language Processing ; A Guide to Theory, Algorithm and System Development*, 1st ed. New Jersey, USA : Prentice Hall, 2001.

[10] Y. Fan, Y. Qian, F. K. Soong, and L. He, "Speaker and language factorization in dnn-based tts synthesis," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5540–5544.

[11] H. Ze, A. Senior, and M. Schuster, "Statistical parametric speech synthesis using deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, BC, 2013, pp. 7962–7966.

[12] A. A. Yousef, S. Sid-Ahmed, and C. Wladyslaw, "Investigating emphatic consonants in foreign accented arabic," *Journal of King Saud University-Computer and Information Sciences*, vol. 21, pp. 13–25, 2004.

[13] T. Redouane, "La reconnaissance en-ligne du manuscript arabe," Ph.D. dissertation, Université des schiences et de la technologie d'Oran -Mohamed Boudiaf-, Oran, Algérie, 2012.

[14] V. Prakāśaṃ, *Encyclopaedia of the Linguistic Sciences : Issues and Theories*, 1st ed. New Delhi, India : Allied Publishers, 2008.

[15] S. Usama, *Introduction to Arabic Linguistics*, 1st ed. Middlebury, Vermont, USA : Middlebury Language School, 2014.

[16] M. Salameh and A.-R. H. Abu-Melhim, "The phonetic nature of vowels in modern standard arabic," *Advances in Language and Literary Studies*, vol. 5, pp. 60–67, 2014.

[17] S. Imed and B. Fateh, "Amélioration d ' un outil de synthèse vocale open source pour la langue arabe," Engineering thesis, Ecole nationale supérieure d'informatique, Alger, Algérie, 2015.

[18] J. C. Watson, *The phonology and morphology of Arabic*, 1st ed. united states : Oxford University Press, 2002.

[19] A. A. a.-R. Salman, "An acoustic phonetic analysis of fortis- lenis consonants in english and arabic," in *al-Adab Journal*, no. 108. Iraq : University of Baghdad College of Arts, 30 jun 2014, pp. 33–76.

[20] "espeak text to speech, offciial website," [en ligne]. Disponible : http://espeak.sourceforge.net/, consulté le : 16-07-2020.

[21] M. Hamiti and R. Kastrati, "Adapting espeak for converting text into speech in albanian," *International Journal of Computer Science Issues (IJCSI)*, vol. 11, no. 4, p. 21, 2014.

[22] "Cstr, « festival manual », 2014. [en ligne]. disponible sur :," http://www.cstr.ed.ac.uk/projects/festival/manual/, consulté le : 16-07-2020.

[23] A. Black, P. Taylor, and R. Caley, "The festival speech synthesis system," in *System documentation. Edition 1.4, for Festival version 1.4.3.* University of Edinburgh, Scotland, UK, 2002.

[24] A. W. Black and K. A. Lenzo, "Building synthetic voices," *Language Technologies Institute, Carnegie Mellon University and Cepstral LLC*, vol. 4, no. 2, p. 62, 2003.

[25] A. W. Black and K. A. Lenzo, "Multilingual text-to-speech synthesis," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, Montreal, Que, 2004, pp. iii–761.

[26] M. Hamad and M. Hussain, "Arabic text-to-speech synthesizer," in *2011 IEEE Student Conference on Research and Development*, Cyberjaya, Malaysia, 19-20 Dec 2011, pp. 409–414.

[27] "The mary text-to-speech system (marytts), offciial website," [en ligne]. Disponible : http://espeak.sourceforge.net/, 2018, consulté le : 16-07-2020.

[28] M. Rashad, H. M. El-Bakry, and I. R. Isma'il, "Diphone speech synthesis system for arabic using mary tts," *International Journal of Computer Science & Information Technology*, vol. 2, no. 4, pp. 18–26, 18-26 August 2010.

[29] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, "Tacotron : Towards end-to-end speech synthesis," in *Interspeech 2017*, Stockholm, Sweden, 20-24 August 2017, pp. 4006–4010.

[30] P. Näslund , "Artificial neural networks in swedish speech synthesis," Master's thesis, School of Electrical and Computer Engineering, Royal Institute of Technology KTH, Sweden, 2018.

[31] "Hhmm/dnn-based speech synthesis system (hts)," [en ligne]. Disponible : http://hts.sp.nitech.ac.jp/, 2017, consulté le : 16-07-2020.

[32] "Site officel de festvox. [en ligne]. disponible sur :," http://festvox.org/, consulté le : 16-07-2020.

[33] Y. A. El-Imam, "Phonetization of arabic : rules and algorithms," *Computer Speech Language*, vol. 18, no. 4, pp. 339–373, October 2004.

[34] H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A. W. Black, and K. Tokuda, "The hmm-based speech synthesis system (hts) version 2.0." in *SSW*. Citeseer, 22-24 August 2007, pp. 294–299.

[35] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech parameter generation algorithms for hmm-based speech synthesis," in *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, vol. 3, 2000, pp. 1315–1318.

[36] O. Abdel-Hamid, S. M. Abdou, and M. Rashwan, "Improving arabic hmm based speech synthesis quality," in *9th International Conference on Spoken Language Processing*, Pittsburgh, PA, USA, 17-21 sep 2006, pp. 1332–1335, [En ligne] Disponible sur : https://www.isca-speech.org/archive/interspeech_2006/i06_1693.html.

[37] K. M. Khalil and C. Adnan, "Optimization of arabic database and an implementation for arabic speech synthesis system using hmm : Hts_arab_talk," *International Journal of Computer Applications*, vol. 73, no. 17, pp. 11–17, 11-17 july 2013.

[38] K. Tokuda, Y. Nankaku, T. Toda, H. Zen, J. Yamagishi, and K. Oura, "Speech synthesis based on hidden markov models," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1234–1252, Mai 2013.

[39] M. Alexsandro, Rodrigues, *Courses on Speech Prosody*, 1st ed. Lady Stephenson Library, UK : Cambridge Scholars Publishing, 2015.

[40] M. K. Krichi and A. Cherif, "Improvements of arabic database and noise reduction of speech signal using wavelet for arabic speech synthesis system using hmm : Hts-arab-talk," *J. Inf. Hiding Multim. Signal Process*, vol. 6, no. 1, pp. 123–130, jan 2015.

[41] S. Raymond, Eric, *The Art of Unix Programming*, 1st ed. Boston, Massachusetts, USA : Addison-Wesley, 2003.

[42] V. Valdis, "Hmm-based speech synthesis system (hts)," [en ligne]. Disponible : https://odo.lv/Recipes/HTS/, 2017, consulté le : 16-07-2020.

[43] "Creating .utt files for english," [en ligne]. Disponible : http://www.cs.columbia.edu/~ecooper/tts/utt_eng.html/, consulté le : 16-07-2020.

[44] B. Alan, W., "The festival speech synthesis system documentation," [en ligne]. Disponible : http://festvox.org/docs/manual-2.4.0/, 2014, consulté le : 16-07-2020.

[45] S. Alencar M. and C. Da Rocha V., *Communication systems*, 1st ed. USA : Springer US, 2005.