الجمهورية الجزائرية الديمقراطية الشعبية
République algérienne démocratique et populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة عين تموشنت بلحاج بوشعيب
Université –Ain Temouchent- Belhadj Bouchaib
Faculté des Sciences et Technologie
Département de l'électronique et de télécommunications

Projet de Fin d'Etudes
Pour l'obtention du diplôme de Master en : Instrumentation
Domaine : Science et technologie
Filière : Electronique
Spécialité : Instrumentation
Thème

Système de détection et congestion de la parole en texte pour la langue Arabe

Présenté Par :
1) Melle. Fekih Yousra

Devant le jury composé de :

Pr C.Ayache	UAT.B.B (Ain Temouchent)	Président
Dr S.Bentayeb	UAT.B.B (Ain Temouchent)	Examinatrice
Mm M.Boutkhil	M A A UAT.B.B (Ain Temouchent)	Encadrante
Dr H.Mekami	M C B UAT.B.B (Ain Temouchent)	Co-Encadrante

Année Universitaire 2022/2023


بسم الله الرحمن الرحيم

Remerciements

Nous remercions Dieu le tout puissant, pour nous avoir donné la force, le courage, la santé et la patience pour pouvoir accomplir ce travail.

Chaleureux remerciement à nos familles qui nous ont soutenues moralement.

Notre sincère gratitude s'adresse à notre encadrante , Madame " Boutkhil Malika", Pour ses précieux conseils, ses orientations et sa patience.

Un grand merci pour notre Co-encadrante Madame "Mekami Hayet" pour sa contribution à la réussite de notre travail.

Nos remerciements les plus sincères et les plus profonds sont adressés aux membres du jury pour l'honneur qu'ils nous ont accordé en évaluant ce travail.

Nous profitons de cette occasion pour remercier tous les professeurs du département d'électronique et télécommunication de l'université Bouchaïb Belhadj qui nous ont aidées à construire un savoir et un savoir-faire.

Enfin nous remercions tous ceux qui ont participé de près ou de loin à l'achèvement de ce mémoire.

DEDICACE

Je dédie ce travail :

A mes chers parents

Je remercie mes très chers parents, qui m'ont toujours apporté le meilleur, qui ont su me guider et me conseiller tout au long de mon parcours.

Merci mes chers parents, qu'ALLAH les bénisse et leur accorde une longue et heureuse vie.

A mes chers frères et sœurs

Pour leur amour et leur soutien inestimable.

A tous mes amis et ma famille

Et à tous ceux qui nous aiment et nous souhaitent le bonheur et la réussite

RESUME

Le traitement de la parole a toujours été l'un des domaines les plus passionnants du traitement du signal. La technologie de reconnaissance vocale permet aux ordinateurs de suivre les commandes vocales humaines et de comprendre le langage humain. Un objectif majeur dans le domaine de la reconnaissance vocale est de développer des techniques et des systèmes qui permettent de parler à des machines.

L'arabe est l'une des langues les plus parlées au monde, se classant au cinquième rang après le mandarin, l'espagnol, l'anglais et l'indien. Malgré son importance, la recherche sur la reconnaissance vocale automatique en arabe demeure insuffisante.

Ce mémoire propose et décrit une application Windows basée sur un système de reconnaissance automatique arabe standard, et traduit le texte extrait en plusieurs langues utilisant le langage de programmation Python.

Pour réaliser notre projet, nous avons utilisé le modèle de Google pour la reconnaissance vocale automatique, qui est basé sur des techniques d'apprentissage en profondeur.

Les résultats obtenus à l'aide de Google Speech Recognition sont souvent de haute qualité et sont largement utilisés dans de nombreuses applications et services.

Mots clés : Reconnaissance automatique de la parole arabe

ABSTRACT

Speech processing has always been one of the most exciting areas of signal processing. Speech recognition technology enables computers to follow human voice commands and understand human language. A major goal in the field of speech recognition is to develop techniques and systems that enable us to talk to machines.

Arabic is one of the most widely spoken languages in the world, ranking fifth after Mandarin, Spanish, English and Indian. Despite its importance, research into automatic speech recognition in Arabic remains insufficient.

This thesis proposes and describes a Windows application based on a standard Arabic automatic recognition system and translates the extracted text into several languages using the Python programming language.

To realize our project, we used Google's model for automatic speech recognition, which is based on deep learning techniques.

The results obtained using Google Speech Recognition are often of high quality and are widely used in many applications and services.

Keywords: Arabic automatic speech recognition

الملخص

لطالما كانت معالجة الكلام واحدة من أكثر المجالات إثارة في معالجة الإشارات. تسمح تقنية التعرف على الصوت لأجهزة الكمبيوتر بتتبع الأوامر الصوتية البشرية وفهم اللغة البشرية. أحد الأهداف الرئيسية في مجال التعرف على الكلام هو تطوير تقنيات وأنظمة تسمح بالتحدث إلى الآلات.

اللغة العربية هي واحدة من أكثر اللغات المنطوقة في العالم، حيث تحتل المرتبة الخامسة بعد الماندرين، والإسبانية والإنجليزية والهندية. على الرغم من أهميتها، لا تزال الأبحاث حول التعرف التلقائي على الكلام باللغة العربية غير كافية.

هذا العمل يقترح ويصف تطبيق ويندوز استنادًا إلى نظام التعرف التلقائي العربي القياسي، ويترجم النص المستخرج إلى عدة لغات باستخدام لغة برمجة بايثون.

لتنفيذ مشروعنا، للتعرف التلقائي على الكلام، استخدمنا نموذج قوقل والذي يعتمد على تقنيات التعلم العميق.

غالبًا ما تكون النتائج التي تم الحصول عليها من خلال التعرف على الكلام من قوقل عالية الجودة وتستخدم على نطاق واسع في العديد من التطبيقات والخدمات.

الكلمات المفتاحية: التعرف التلقائي على الكلام العربي

Sommaire

Liste de figure

Liste de tableaux

INTRODUCTION GENERALE

On voit ces dernières années la place de plus en plus grande de la communication vocale dans nos appareils intelligents. Aujourd'hui on peut activer notre smartphone via une simple commande vocale, on peut dicter à sa voiture la destination souhaitée et on peut même demander à un assistant vocal de commander à manger. Et ce n'est que la partie émergée de l'iceberg car les applications sont très nombreuses. Côté business, les avancées technologiques permettent de transcrire automatiquement une réunion ou un compte rendu vocal, d'estimer le niveau de satisfaction d'un client à travers les traits de son expression, de vérifier l'identité d'un client qui appelle son conseiller bancaire… Toutes ces prouesses sont les fruits de l'exploitation, par les capacités de l'intelligence artificielle, des informations offertes par notre voix.

En effet, la parole est un moyen de communication primordial chez l'homme. Elle est tellement riche en informations que les scientifiques essayent sans cesse de l'analyser afin d'en comprendre les différents aspects. Depuis les années 1950, de nombreuses équipes de chercheurs (informaticiens, phonéticiens, mathématiciens, linguistes…) se sont penchées sur un objectif commun : automatiser les processus d'interprétation de la parole, mais aussi de sa production. La reconnaissance automatique de la parole (RAP), la reconnaissance du locuteur et la synthèse de la parole ont particulièrement intéressé les académiques ainsi que les entreprises. Les résultats de ces problématiques représentent maintenant l'interface d'échange avec beaucoup de nos appareils intelligents, notamment avec nos assistants vocaux. La reconnaissance automatique de la parole est un domaine qui fascine le public et de nombreux chercheurs depuis près de trois décennies.

La reconnaissance de la parole consiste à transcrire automatiquement un contenu parlé afin obtenir la séquence de mots correspondante. Les premiers systèmes étaient capables de transcrire uniquement des mots isolés avec un vocabulaire réduit.

Malheureusement, malgré les incroyables avancées de l'informatique et des connaissances, la reconnaissance automatique de la parole reste un sujet de recherche toujours actif, et les résultats obtenus sont encore loin des idéaux auxquels on aurait pu s'attendre il y a vingt ans.

Cependant, si le système de reconnaissance idéal n'existe pas encore, des applications concrètes voient peu à peu le jour. La reconnaissance vocale automatique commence à équiper certains téléphones portables ou GPS, en reconnaissant certains mots clés, la tâche demandée peut être effectuée. Les systèmes de reconnaissance de la parole sont également utilisés pour indexer de vastes bases de données audiovisuelles, effectuer des recherches de termes dans des flux audio et servir d'interface homme-machine. Dans des conditions d'utilisation adéquates, ces systèmes se révèlent efficaces en pratique. Cependant, les limitations principales des systèmes actuels concernent leur robustesse : les conditions d'utilisation doivent être similaires à celles utilisées lors de l'entraînement du système, l'environnement sonore doit être peu bruyant et les locuteurs ne peuvent pas parler simultanément. Souvent, les utilisateurs doivent s'adapter pour utiliser les logiciels.

De plus en plus, la technologie de reconnaissance de la parole progresse vers des applications réelles. Actuellement, un changement qualitatif dans l'état de l'art est en cours, promettant d'offrir des capacités de reconnaissance accessibles à tous.

Le signal de la parole est l'un des signaux les plus complexes, ce qui rend difficile sa caractérisation par un modèle simple. La variabilité du signal est l'un des problèmes majeurs de la reconnaissance de la parole. Les pionniers du traitement de la parole pensaient initialement que la parole était une composition linéaire d'éléments distinctifs appelés phonèmes, et qu'en utilisant des échantillons représentant ces phonèmes, il serait possible de reconstruire ou reconnaître n'importe quelle phrase. Cependant, cette idée théorique ne s'applique pas toujours dans le cas de la reconnaissance de la parole continue en raison du phénomène de coarticulation.

Pour surmonter ces difficultés, de nombreuses méthodes et modèles mathématiques originaux ou adaptés d'autres domaines ont été développés. Parmi ceux-ci, on peut citer la comparaison dynamique, les systèmes experts, les réseaux de neurones, les modèles stochastiques, notamment les modèles de Markov cachés, et bien d'autres encore. Ces approches ont été utilisées pour traiter la reconnaissance de la parole en prenant en compte la variabilité du signal et le phénomène de coarticulation.

Notre travail s'inscrit dans le domaine de la reconnaissance automatique de la parole et vise à développer un Système de Reconnaissance Automatique de la Parole (SRAP) basé sur Python. Ce système sera basé sur les modèles de l'API de reconnaissance de la parole de Google pour la langue arabe standard.

Notre mémoire comporte trois chapitres :

- Dans le premier chapitre, nous présentons une brève description de la reconnaissance vocale, qui contient les sujets suivants : technologie vocale, ses avantages, difficultés et applications de la parole, type de reconnaissance de la parole et c'est approches, illustrer la langue arabe standard et particularité de cette langue et une vue basic sur l'architecture de système de reconnaissance de la parole.
- Le deuxième chapitre traite la technique d'extraction de caractéristiques, les approches de la modélisation acoustique et les modèles de la traduction.
- Dans le dernier chapitre nous allons décrire les étapes essentielles à l'élaboration d'un système la reconnaissance de la parole arabe Ensuite, nous présentons les résultats des expérimentations menées afin de tester l'efficacité de notre système proposé.

Chapitre 1

La reconnaissance automatique de la parole

1.1 Introduction

La parole est une forme de communication très efficace et naturelle utilisée par les humains. Ils rêvaient depuis longtemps de pouvoir s'adresser à des machines de la même manière, ce qui les rendrait encore plus intelligentes.

Cependant, malgré d'énormes efforts de recherche pour essayer de créer une telle machine intelligente capable de reconnaître le langage parlé et d'en comprendre le sens, nous sommes encore loin de l'objectif souhaité, ce qui est dû aux limites du système de reconnaissance vocale. Alors, que signifie la reconnaissance automatique de la parole (RAP) ?

1.2 Reconnaissance automatique de la parole

La reconnaissance vocale automatique (RAP) est une technologie qui permet aux ordinateurs de reconnaître ce qu'une personne dit dans un microphone ou un téléphone. Il a un large éventail d'applications : reconnaissance de commandes (interface utilisateur vocale avec des ordinateurs), dictée, réponse vocale interactive, et peut être utilisé pour apprendre des langues étrangères. L'RAP peut également aider les personnes handicapées à interagir avec la société. C'est une technologie qui facilite la vie.

Elle a développé de nombreux systèmes, notamment : Dragon Naturally Speaking, IBM via la voix, Microsoft SAPI. De nombreux systèmes de reconnaissance vocale open source sont également disponibles, basés sur des modèles de Markov cachés (HMM) (Haton, Cerisara, Fohr, Laprie, & Smaïli, 2006).

Figure 1.1 Mécanisme du système de reconnaissance vocale (Huang, Alleva et al. 1993)

1.3 Histoire du développement de l'RAP

Les premières expérimentations de reconnaissance automatique de la parole sur les machines ont débuté dans les années 1950.

- En 1952, Davis, Biddulph et Balashek ont développé le premier système RAP significatif aux laboratoires Bell. Ce système était capable de reconnaître des chiffres isolés et permettait la reconnaissance d'un seul locuteur. (K. H. Davis, Biddulph, & Balashek, 1952)

- Dans les années 1960 et 1970, de nombreuses techniques fondamentales ont émergé dans le domaine de l'RAP. Parmi celles-ci, on compte la transformée de Fourier rapide (FFT)(Cooley & Tukey, 1965), l'analyse cepstrale (Bogert, 1963) et le codage prédictif linéaire (LPC) (Markel, Gray, & Wakita, 1973) pour extraction des coefficients. De plus, la technique de déformation temporelle dynamique (DTW) a été développée pour mesurer la similarité entre les séquences qui peuvent varier au fil du temps, tandis que la technique des modèles de Markov cachés (HMM) a été utilisée pour la reconnaissance.

- Dans les années 1980, la problématique des mots connectés a attiré une attention particulière. De plus, l'approche de la reconnaissance des formes a évolué des méthodes basées sur les modèles vers les méthodes de modélisation statistique. En particulier, l'approche basée sur les modèles de Markov cachés (HMM) a été largement étudiée et implémentée par différents laboratoires tels que Bell (Rabiner & Juang, 1993), CMU et IBM. L'approche HMM est devenue la technique clé de cette période. De plus, une autre technique qui a été réintroduite à la fin des années 1980 est celle des réseaux de neurones artificiels. Bien que les réseaux de neurones aient été initialement introduits dans les années 1950 (Fitch, 1944) ils n'ont pas connu de résultats notables à leurs débuts.

- À partir des années 1990, la reconnaissance de la parole continue à grand vocabulaire (Large Vocabulary Continuous Speech Recognition) est devenue un sujet d'intérêt majeur pour les chercheurs. Pendant cette période, de nombreuses techniques ont été développées, notamment les réseaux de neurones récurrents basés sur des cellules (Long Short-Term Memory : LSTM), qui ont été proposés en 1997 par Sepp Hochreiter et Jürgen Schmidhuber (Hochreiter & Schmidhuber, 1997).

- À partir des années 2000, le domaine de l'apprentissage profond (Deep Learning : DL) a été introduit, relançant l'utilisation des réseaux de neurones dans le traitement automatique de la parole. En 2007, les réseaux LSTM ont commencé à révolutionner la reconnaissance automatique de la parole, surpassant les modèles traditionnels dans

certaines applications (Fernández, Graves, & Schmidhuber, 2007). En 2014, Kyunghyun Cho a proposé une variante simplifiée appelée réseaux de neurones récurrents à portes (Gated Recurrent Unit : GRU) (Cho et al., 2014).

1.4 Les avantages de la reconnaissance automatique de la parole

Les avantages de la reconnaissance vocale sont nombreux. Il libère complètement l'usage de la vision et des mains, permettant aux utilisateurs de se déplacer librement. Dans ARP, les informations sont transmises plus rapidement que ne le permettrait l'utilisation du clavier. En fin de compte, presque tout le monde sait parler et peu de gens sont à l'abri de choses comme les fautes de frappe et d'orthographe. Ces avantages sont si importants qu'il existe déjà sur le marché des appareils dont l'utilisation est limitée mais qui fonctionnent toujours. Voici quelques applications qui ont vu le jour (Cerisara, 1999) :

- Saisie de données vocales ;
- Donner des ordres en conduisant une voiture ou un avion ;
- Aider les personnes handicapées ;
- Commande vocale de machines ou de robots ;
- Commande vocale des montres portables, etc.

1.5 Difficultés de la reconnaissance de la parole

Les signaux de parole sont parmi les plus complexes à caractériser et à analyser en raison de leur grande variabilité. Cette complexité est liée à la génération des signaux de parole ainsi qu'aux aspects techniques. Les signaux vocaux varient non seulement avec la voix de celui qui parle, mais aussi avec le locuteur, l'âge, l'humeur, la santé, l'environnement. De plus, la mesure des signaux de parole est fortement influencée par la fonction de transfert du système de reconnaissance (équipement d'acquisition et de transmission) ainsi que par le milieu environnant. Par conséquent, le principal obstacle à l'amélioration des performances des systèmes RAP provient de la grande complexité du signal de parole due à la combinaison de plusieurs facteurs, principalement la redondance du signal acoustique, l'énorme variabilité intra/interlocuteur, l'effet d'articulation articulaire de la parole continue et conditions d'enregistrement.

La redondance : Les signaux vocaux ont des caractéristiques redondantes. Il contient plusieurs types d'informations : le son, la syntaxe et la sémantique de la phrase, l'identité du locuteur et son état émotionnel. Bien que cette redondance assure une certaine capacité anti-bruit du message, il devient plus difficile d'extraire des informations pertinentes pour RAP du fait de la nature multimodale de la source d'information.

La variabilité : Les signaux sonores de deux énoncés ayant le même contenu de parole sont différents pour un même locuteur (variabilité intra-locuteur) ou différents locuteurs (variabilité interlocuteur). En effet, lorsqu'une même personne prononce deux fois le même énoncé, il y a un changement notable de la signature acoustique dû à :

L'état physique, par exemple, la fatigue ou le rhume.

- Les conditions psychologiques, comme le stress.
- Les émotions du locuteur.
- Le rythme lié à la durée des phonèmes (façon dont s'exprime le locuteur) et l'amplitude (voix normale, voix chuchotée, voix criée).

Cependant la variabilité interlocuteur est a priori la plus importante. Elle s'explique par :

- Les différences physiologiques entre locuteurs.
- Les habitudes acquises en fonction du milieu social et géographique comme les accents régionaux.

Cette variabilité rend très difficile la définition d'invariants, et admettre. Par conséquent, il est nécessaire de pouvoir distinguer les caractéristiques des phonèmes des aspects propres à chaque intervenant.

Continuiste et coarticulation : En prévision de la posture d'articulation, la production sonore est fortement influencée par les sons avant et arrière. Il est parfois impossible de localiser correctement un discours isolé de son contexte. Apparemment bonne reconnaissance des mots isolés Il est plus facile d'être séparé par le silence que

de reconnaître des mots liés. En effet, dans ce dernier cas, non seulement les frontières entre les mots ne sont plus connues, mais les mots deviennent nets et distincts.

Conditions d'enregistrement : L'enregistrement du signal de parole dans des conditions défavorables rend l'extraction des informations pertinentes nécessaires à la reconnaissance des mots contenues dans ce signal difficile. En effet, les perturbations causées par le microphone (en fonction de son type, de sa distance et de son orientation) ainsi que l'environnement (bruit, réverbération) compliquent considérablement le problème de la reconnaissance.

Pour illustrer l'ensemble de ces difficultés, un système de RAP doit être en mesure de déterminer que "la prononciation d'un « a » par un adulte masculin est plus similaire à celle d'un « a » prononcé par un enfant, dans un mot différent, dans un environnement différent et avec un autre microphone, qu'à celle d'un « o » prononcé dans la même phrase par le même adulte masculin (MARIANI, 1990)

1.6 Domaine d'application de la reconnaissance automatique de la parole :

Bien que toute tâche impliquant une interface avec un ordinateur puisse utiliser RAP, les applications suivantes sont de loin les plus courantes :

- **La dictée :** Aujourd'hui, la dictée est l'utilisation la plus courante des systèmes RAP. La transcription, la dictée juridique et commerciale et le traitement de texte général sont les exemples les plus importants d'utilisation de la dictée. Dans certains cas, des vocabulaires spéciaux sont utilisés pour améliorer la précision du système
- **Le contrôle et commande :** Les systèmes RAP conçus pour exécuter les fonctions et les actions du système sont définis comme des systèmes de commande et de contrôle. Des déclarations comme "open chrome" feront exactement cela.
- **Téléphonie :** Certains systèmes vocaux permettent aux appelants de prononcer des commandes au lieu d'appuyer sur des boutons pour envoyer des tonalités spécifiques.
- **Médical/handicap :** Les contraintes physiques telles que les microtraumatismes répétés ou la dystrophie musculaire, entre autres, peuvent rendre la tâche de saisie difficile pour de nombreuses personnes. Pour illustrer, les individus malentendants

pourraient bénéficier d'un système connecté à leur téléphone qui convertit automatiquement la parole de l'appelant en texte.

- **Applications militaires :** Telles que le contrôle vocal de certaines des fonctions des appareils militaires (avions de chasse).
- **Production :** Où la reconnaissance de la parole peut être utilisée pour la commande de contrôle vocal de processus de fabrication (par exemple, pour l'accès aux systèmes de contrôle de qualité) et apporter une aide au tri et envoi de paquets.

Ou encore le sous-titrage et la traduction automatique de vidéo, l'indexation et l'extraction d'information dans les documents audiovisuels, et les interfaces vocales homme-machine.

1.7 Modes de fonctionnement

Il est possible d'utiliser un système de reconnaissance selon plusieurs modes.

1.7.1. Dépendant du locuteur (mono-locuteur)

Dans cette situation spécifique, le système de reconnaissance de la parole peut être utilisé par un seul locuteur à la fois. Les principaux systèmes de dictée vocale actuels nécessitent une phase d'apprentissage recommandée avant toute utilisation, afin d'adapter les paramètres à la voix de l'utilisateur.

1.7.2 Multi-locuteur

Le système de reconnaissance peut être utilisé par un groupe restreint de personnes, permettant la transition d'un locuteur à un autre au sein du même groupe sans nécessiter d'adaptation supplémentaire.

1.7.3 Indépendant du locuteur

Le système de reconnaissance peut être utilisé par n'importe quel locuteur (Khelil, Berrah, & Amiar, 2022).

1.8 Types de reconnaissance vocale

Les systèmes RAP peuvent être regroupés en différentes catégories en fonction de leur capacité à reconnaître différents types d'énoncés. Ces catégories sont définies en tenant compte de la difficulté que rencontre l'RAP à déterminer le début et la fin d'un énoncé prononcé par un locuteur (Singh, Nath, & Kumar, 2018).

1.8.1 Mots isolés : Les systèmes de reconnaissance des mots isolés fonctionnent en acceptant un seul mot à la fois. Ils sont souvent équipés de modes "Écouter/Ne pas écouter", ce qui signifie qu'ils requièrent une pause entre les mots pour reconnaître chaque mot individuellement. Ce type de reconnaissance convient dans les situations où l'utilisateur doit fournir une seule réponse au système RAP ou utiliser des mots isolés pour donner des commandes.

1.8.2 Mots connectés : Les systèmes de reconnaissance des mots enchaînés sont capables de traiter des mots qui sont séparés par des pauses. Ils présentent des similitudes avec les systèmes de reconnaissance des mots isolés, mais ils permettent que des énoncés distincts soient exécutés ensemble avec une pause minimale entre eux.

1.8.3 Parole continue : Les systèmes de reconnaissance de la parole continue permettent aux utilisateurs de s'exprimer de manière quasi naturelle, en traitant la parole où les mots sont connectés sans être séparés par des pauses. Cependant, ces systèmes sont confrontés à des défis tels que la coarticulation, la vitesse d'élocution et les limites des mots inconnues, ce qui impacte leurs performances. La création de tels systèmes est considérée comme complexe, car ils nécessitent l'utilisation de méthodes spéciales pour déterminer les limites entre les mots.

1.8.4 Parole spontanée : La parole spontanée se réfère à une forme naturelle de communication orale où le contenu n'est pas préalablement connu. Un système RAP qui traite de la parole spontanée doit être capable de gérer une diversité de caractéristiques propres à la parole naturelle, telles que les mots prononcés simultanément (légers bégaiements) et l'utilisation de non-mots comme "um" ou "ah".

1.9 Approches de la reconnaissance de la parole

Il y a deux approches pour aborder la reconnaissance de la parole : l'approche globale et l'approche analytique. Ces deux approches se différencient principalement par la nature et la taille des unités abstraites qu'elles cherchent à aligner avec le signal de parole.

1.9.1 Approche globale

Dans l'approche globale, l'unité de base est souvent le mot considéré comme une entité globale non décomposée. L'idée derrière cette méthode est de fournir au système une représentation acoustique de chaque mot, afin qu'il puisse l'identifier ultérieurement. Cette opération est réalisée lors de la phase d'apprentissage, où chaque mot est prononcé une ou

plusieurs fois. L'avantage de cette méthode est qu'elle évite les effets liés à l'articulation. Cependant, elle est limitée à un vocabulaire restreint et à un nombre restreint de locuteurs.

1.9.2 Approche analytique

L'approche analytique (structure du mot) contrairement à l'approche globale, se base sur la structure linguistique des mots. Elle cherche à détecter et à identifier les composants de base tels que les phonèmes et les syllabes. Ces éléments constituent les unités fondamentales à reconnaître. L'avantage de cette méthode est sa simplicité, car seules les caractéristiques des unités de base, et non des mots entiers, doivent être enregistrées en mémoire. En réalité, les deux approches sont fondamentalement similaires, la différence réside dans l'entité à reconnaître : le mot pour l'approche globale et le phonème ou la syllabe pour l'approche analytique (Douib, 2018).

1.9.3 Principe général de la méthode globale et analytique

Les deux méthodes de reconnaissance vocale sont mises en œuvre en deux phases, la phase d'apprentissage suivie de la phase de reconnaissance.

- **La phase d'apprentissage**

L'utilisateur dicte tout le vocabulaire utilisé dans la commande vocale pour créer une signature audio de référence de la commande. Mais pour l'analyse, les utilisateurs ne dictent que quelques mots spécifiques qui contiennent un grand nombre de phonèmes consécutifs.

- **La phase de reconnaissance**

L'utilisateur prononce la commande vocale réelle contenant le vocabulaire stocké. Ensuite, le système de reconnaissance de texte est un problème typique de reconnaissance de formes. Tout système de reconnaissance de formes se compose toujours des trois parties suivantes :

- des capteurs (dans notre cas des microphones) qui permettent de comprendre le phénomène physique considéré ; - étape de paramétrage de forme (par exemple analyseur spectral) ;

- La phase de décision chargée de classer une forme inconnue dans l'une des classes possibles(Douib, 2018).

1.10 Architecture du système de reconnaissance automatique de la parole

Un système de reconnaissance automatique de la parole comporte typiquement 4 modules :

- Le prétraitement acoustique, qui va identifier les zones de parole dans l'enregistrement à transcrire et en extraire des séquences de paramètres acoustiques.
- Extraction de caractéristiques (features), qui extraire les informations caractéristiques du signal de parole en éliminant au maximum les parties redondantes.
- Le décodeur, qui va combiner les prédictions des modèles de prononciation, acoustiques et linguistiques pour proposer la transcription en texte la plus probable pour un énoncé de parole donné. Ou le modèle de prononciation associe les mots connus par le système à leurs représentations phonétiques, le modèle acoustique, servant à prédire les phonèmes les plus probablement prononcés dans un énoncé audio et le modèle linguistique, servant à prédire la séquence de mots la plus probable pour un texte donné.
- Le Post-traitement, c'est une sélection finale du mot à partir de ceux issus du décodeur.

1.11 La langue arabe

La langue arabe est considérée comme l'une des langues officielles dans vingt-deux pays situés au Moyen-Orient, en Afrique et dans le Golfe. Elle est classée comme la cinquième langue la plus utilisée dans le monde (Al-Anzi & AbuZeina, 2022) et est utilisée par plus de 422 millions de personnes, qu'elles soient natives ou non. Selon (Abdelhamid, Alsayadi, Hegazy, & Fayed, 2020) la langue arabe peut être classée en trois catégories principales, à savoir:

- L'arabe classique représente la forme la plus formelle et la plus standard de l'arabe, car il est principalement utilisé dans le Saint Coran et les instructions religieuses de l'Islam;
- L'arabe standard moderne (ASM) représente la norme linguistique formelle actuelle de la langue arabe. Il est généralement utilisé dans la communication écrite et les médias, et est enseigné dans les établissements d'enseignement ;
- L'arabe dialectal (AD), également appelé arabe familier, est une variante de la même langue spécifique à des pays ou à des groupes sociaux, utilisée dans la vie de tous les jours. Il existe plusieurs dialectes de l'arabe et, parfois, plusieurs DA peuvent être utilisés dans un même pays (Abushariah, 2017).

1.11.1 Particularités de la langue arabe

Absence des voyelles

La langue arabe est une langue sémitique qui s'écrit et se lit de droite à gauche. Son alphabet comprend deux types de symboles pour former des mots : les lettres et les signes diacritiques. Contrairement aux langues latines, l'arabe ne fait pas de distinction entre les lettres minuscules et majuscules. Les 28 lettres de l'alphabet arabe représentent les sons consonantiques de la langue. Chaque lettre peut prendre jusqu'à quatre formes différentes en fonction de sa position dans un mot : initiale, médiane, finale ou isolée. Les lettres sont généralement connectées entre elles. Pour des raisons phonétiques, les lettres de l'alphabet arabe sont classées en deux groupes : les lettres lunaires.

Le deuxième type de symboles dans l'alphabet arabe est celui des signes diacritiques. Il existe trois types de signes diacritiques dans l'écriture arabe : les voyelles, la nunation et le shadda.

- **Les voyelles** : au nombre de trois, sont appelés aussi voyelles courtes : la damma◌ُ se représente comme une virgule (,) qui apparaissent ur le dessus d'une consonne, la fatha/◌َ se représente comme un trait d'union (-) qui apparaît sur le dessus d'une consonne et la kasra / ◌ِse représente comme un trait d'union (-) qui apparaissent en dessous d'une consonne
- **Nunation :**/Altnwyn/ peut seulement survenir dans la position finale d'un mot dans les nominales (noms, adjectifs et adverbes), où ils indiquent l'indétermination. Ils représentent la combinaison d'une voyelle courte et le marqueur non écrit /n/.
- **Shadda :** /Al$dap/ est un signe diacritique ressemblant à la lettre minuscule. Elle sert principalement à indiquer qu'une consonne est géminée, ce qui est l'équivalent d'un doublement de consonne. Elle est placée au-dessus de la consonne en question. Elle est aussi employée dans les textes où les diacritiques sont absents pour limiter l'ambiguïté

En langue arabe, les lettres sont toujours écrites, tandis que les signes diacritiques sont facultatifs. Ainsi, l'écriture arabe peut être totalement voyellée, partiellement voyellée ou entièrement dépourvue de voyelles. Cependant, l'absence de voyelles dans les textes arabes génère plusieurs cas d'ambiguïtés et pose des problèmes lors de l'analyse automatique. En effet, l'ambiguïté grammaticale augmente lorsque les mots ne sont pas voyellés. Cela est dû au

fait qu'un mot non voyelle peut avoir plusieurs voyellations possibles, et chaque voyellation est associée à une liste différente de catégories grammaticales (Belguith, 1999).

Agglutination

Contrairement aux langues latines, en arabe, « les articles 2 », « les prépositions 3 », « les pronoms 4 », etc. collent aux adjectifs, noms, verbes et particules En comparaison avec le français, un mot arabe peut parfois avoir la signification d'une phrase complète en français. Exemple : le mot arabe أتتـذكرونا correspond en français à la phrase « Est ce que vous vous souvenez de nous ? » (Souissi, 1997)

Cette caractéristique peut entraîner une ambiguïté morphologique, car il peut parfois être difficile de distinguer entre une proclitique (attaché au début du mot) ou un enclitique (attaché à la fin du mot) et un caractère original du mot lui-même.

1. Les articles : par exemple « ال »
2. Les prépositions sont : « ب, ك,ل,من,حتى,مع,في,لن »
3. Le pronom personnel en arabe est isolé ou affixé. Isolé, il correspond en français à : moi, toi, etc.

1.12. Conclusion :

Dans le présent chapitre nous avons défini la reconnaissance automatique de la parole, ses avantages, ses difficultés et ses applications. Nous avons vu aussi les approches de la RAP et les architectures et dans le prochain chapitre nous allons présenter les techniques de conversion de la voix ou de la parole en texte.

Chapitre 02

Extraction de l’information textuelle

à partir un signal acoustique

2.1. Introduction

L'extraction de l'information textuelle partir de la parole, par une machine, à fait l'objet de plusieurs travaux de recherche. Le but est d'aboutir à un concept très proche de la réalité c'est-à-dire réduisant le maximum possible le taux d'erreur. Il est essentiel d'avoir une bonne vision d'ensemble du processus permettant de passer d'un signal audio à sa transcription textuelle. L'objectif de ce chapitre est donc de dresser un panorama théorique et pratique du fonctionnement d'un système de reconnaissance de la parole SRAP.

2.2. Les Systèmes de reconnaissance automatique de la parole

Un Système de Reconnaissance Automatique de la Parole (SRAP) a pour objectif la transcription textuelle d'un signal de la parole. Une fois les signaux audios sont numérisés un système de reconnaissance vocale commence à traiter ce signal à travers différentes étapes pour générer le texte le plus probable.

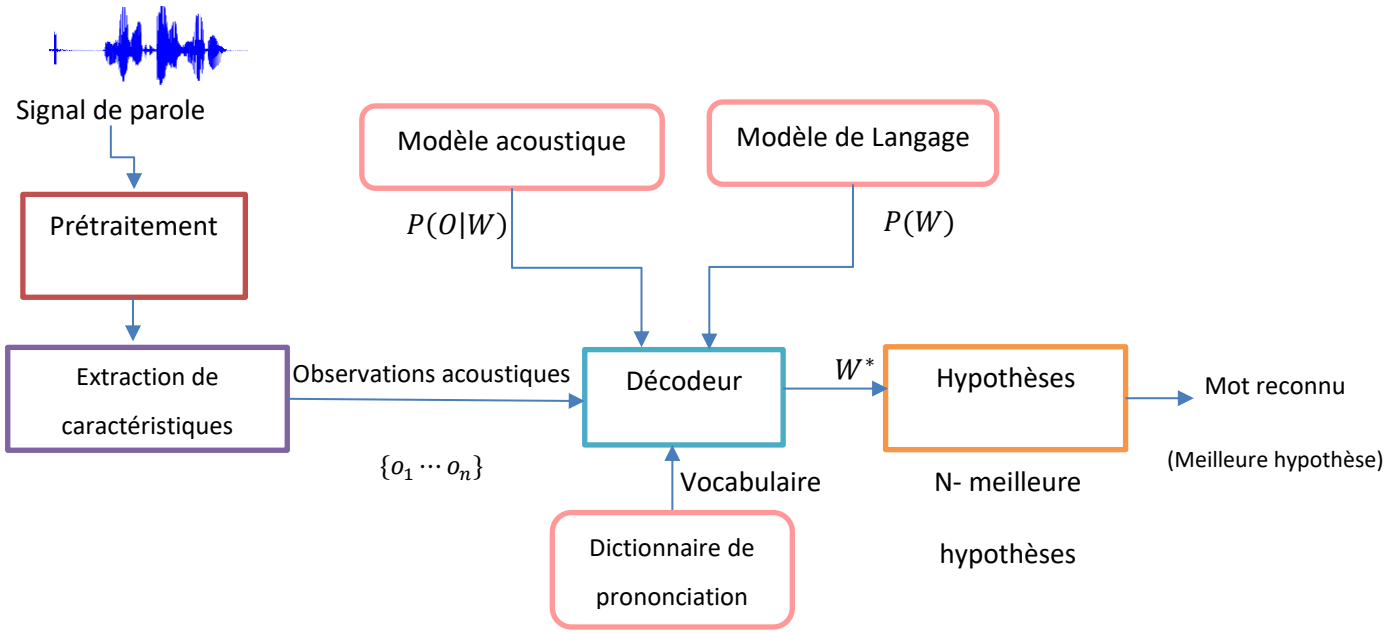


Figure 2.1 : Architecture d'un système de reconnaissance vocale typique

A l'architecture illustrée sur la figure 2.1, le signal vocal est reçu, ce signal est soumis à des opérations de prétraitement pour l'améliorer, telles que l'application de filtres de préaccentuation et la suppression ou la réduction du bruit. Ensuite, les caractéristiques sont extraites pour obtenir des caractéristiques discriminatives utiles pour l'étape suivante. Les paramètres obtenus sont transmis au décodeur. Dont, le but de la phase de décodage est de trouver la correspondance la plus probable entre toute séquence de mots et le signal vectoriel

de caractéristique en appliquant des modèles linguistiques et acoustiques considérés au cœur des systèmes de reconnaissance de la parole. Le but de la dernière étape est d'obtenir une hypothèse parfaite parmi les n-meilleures hypothèses (Gruhn, Minker, & Nakamura, 2011).

2.2.1 Prétraitement

Cette étape améliore le signal de parole, par exemple en appliquant un filtre de pré-focalisation et en supprimant ou en réduisant le bruit pour obtenir une meilleure différenciation du signal. Différentes techniques et opérations peuvent être réalisées à cette étape :

- Convertit un signal audio analogique en signal numérique en échantillonnant le signal avec un taux d'échantillonnage approprié, tel que 16 kHz ou 8 kHz, et en appliquant un processus de quantification.

- Traitement de préaccentuation : Le but d'un filtre de préaccentuation est d'augmenter l'amplitude des signaux haute fréquence et de réduire l'amplitude des signaux basse fréquence (par exemple 1 Hz).

- Processus de segmentation parole/non-parole : ce processus supprime des parties de l'enregistrement, telles que la partie entre le début de l'enregistrement et le moment où le discours commence, et la fin de l'enregistrement lorsque le discours se termine. Le but est d'éliminer le bruit qui peut se produire pendant l'enregistrement.

2.2.2. Extraction de fonctionnalités

L'extraction de caractéristiques est la première étape d'un système ASR. Elle convertit la forme d'onde du signal de parole en un ensemble de vecteurs de caractéristiques dans le but d'obtenir une discrimination élevée entre les phonèmes. L'extraction de caractéristiques effectue toutes les mesures nécessaires sur le segment sélectionné qui sera utilisé pour prendre une décision (Doukas, Bardis, & Markovskyi, 2017). Les caractéristiques mesurées peuvent être utilisées pour mettre à jour les mesures statistiques à long terme afin de faciliter l'adaptation du processus à des conditions environnementales variables (principalement en arrière-plan) (Doukas & Bardis, 2017). L'extraction de caractéristiques déterminera les zones vocales de l'enregistrement à écrire et en extraira des séquences de paramètres acoustiques.

Pour ce faire, le signal est tout d'abord découpé en trames. Chaque trame est considérée comme une fenêtre de 10 à 20 ms de signal. Dans cet intervalle, on suppose que le signal vocal est suffisamment stable. Par la suite, un vecteur de paramètres acoustiques est extrait

pour chacune de ces trames. Suite à ces prétraitements, on obtient une séquence d'observations acoustiques $\boldsymbol{O}$, où $O = \boldsymbol{o_1}\ \boldsymbol{o_2}...\boldsymbol{o_n}$, et chaque vecteur $\boldsymbol{o_i}$ représente quelques millisecondes (typiquement 10 ms) (Dammak, 2016).

Il existe de nombreuses techniques d'extraction de caractéristiques, comme indiqué dans rang et Gupta 2015), notamment :

- Analyse par codage prédictif linéaire (Linear Predictive Coding :LPC) (O'Shaughnessy, 1988) : Le LPC est un modèle paramétrique du signal de parole pris du modèle humain de la production de la parole. Cette technique s'appuie particulièrement sur l'hypothèse que la parole peut être modélisée par un processus linéaire, qui cherche à prédire le signal $\boldsymbol{s(n)}$ à un instant $\boldsymbol{n}$ à partir des $\boldsymbol{p}$ échantillons précédents ;
- Analyse spectrale relative (RelAtive SpecTrAl : RASTA)(Hermansky, Morgan, & processing, 1994) RASTA est conçu pour réduire l'impact du bruit et améliorer la qualité de la parole. Cette technique est largement utilisée pour les signaux vocaux bruités ;
- L'analyse discriminante linéaire (LDA) et la LDA probabiliste (Ioffe, 2006) : Cette technique utilise les variables dépendantes de l'état du modèle de Markov caché (HMM) sur l'extraction des i-vecteurs. Le i-vecteur est un vecteur de faible dimension et de longueur fixe qui contient des informations pertinentes. ;

Analyse par coefficients cepstraux à échelle de Mel (Mel Frequency Cepstral Coefficients : MFCC)(S. Davis, Mermelstein, & processing, 1980) : Il s'agit de la technique la plus couramment utilisée, avec un décalage de trame et une longueur généralement comprise entre 20 et 32 ms. Cette technique est peu complexe et son taux de reconnaissance est élevé. En raison de leur importance, ils seront décrits dans la sous-section suivante.

2.2.2.1Analyse des Coefficients cepstraux à échelle Mel

Les MFCC (Mel Frequency Cepstral Coefficients) sont utilisés pour les tâches de reconnaissance de la parole, du langage ou du locuteur. Bien que le grand nombre d'ouvrages proposent d'autres méthodes, les MFCC restent les plus utilisées dans le domaine du traitement de la parole. Le processus d'extraction des MFCC est décrit par figure 2.2.

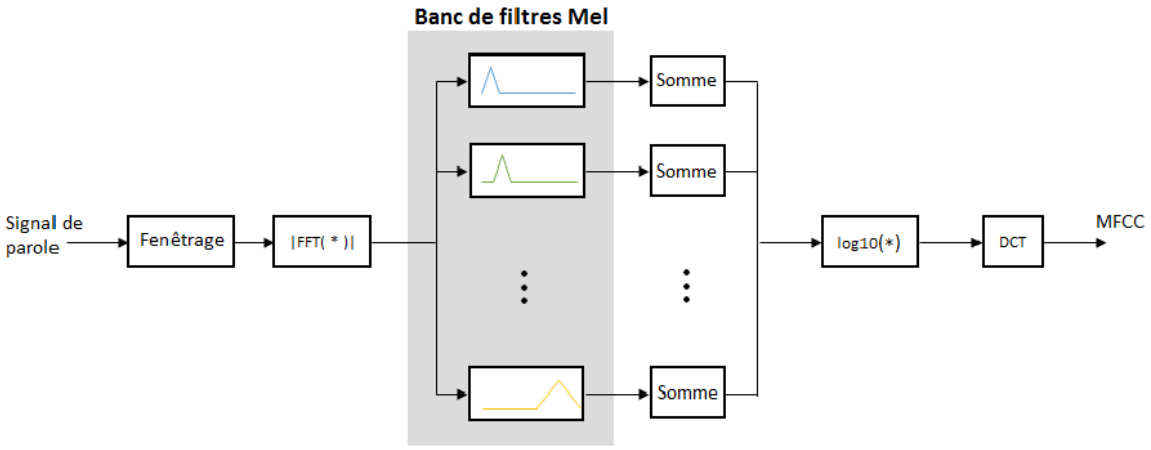


Figure 2.2. Schéma fonctionnel des étapes de calcul des MFCCs.

Le signal de parole est analysé localement en utilisant le fenêtrage temporel (souvent Hanning ou Hamming) afin de réduire les effets de bord causés par la troncature (fenêtre rectangulaire). La longueur de la fenêtre glissante utilisée (entre 20 et 30 millisecondes) est choisie pour respecter la stationnarité. Le décalage des fenêtres temporelles utilisées pour extraire deux segments consécutifs du signal est choisi de sorte que ces fenêtres chevauchent en partie le segment du signal tramé et ensuite appliqué à une transformation de Fourier rapide (Fast Fourier Transform - FFT).

Le module du spectre obtenu est filtré par un banc de filtres qui permet de réduire la taille du vecteur spectral en calculant la moyenne du spectre sur la bande de fréquence correspondant à chacun des filtres.

Les fréquences centrales de chaque filtre sont fixées par l'échelle MEL. Le logarithme de ces valeurs est calculé et multiplié par 20 pour obtenir l'enveloppe spectrale en décibels.

Les coefficients acoustiques ainsi obtenus à ce stade peuvent être directement utilisés pour les prochaines étapes de traitement. Dans ce cas, on parlera dans ce cas de banc de filtres logarithmique ou log filter-bank (FB) en anglais.

La dernière étape de la paramétrisation consiste à appliquer une transformée en cosinus discrète (DCT) à partir de laquelle résultent les coefficients Cepstraux (MFCC).

La transformée en cosinus discrète est utilisée ici pour sa capacité à décorréler les données.

Des informations dynamiques sont ajoutées à ces coefficients en les concaténant avec leurs dérivées première et seconde temps inférieures à celles des basses fréquences.

Le principe de calcul des coefficients MFCC repose sur des recherches psychoacoustiques sur la tonie et la perception de différentes bandes de fréquences par l'oreille humaine. La FFT passe dans un banc de filtres à l'échelle de Mel. Cette échelle non linéaire tient principalement compte du fait que la perception des intervalles change en fonction de la zone du spectre à laquelle appartiennent les hauteurs qui les composent. L'intérêt principal de ces coefficients est d'extraire des informations pertinentes en nombre limité, à la fois sur la base de la production (théorie de Cepstale) et sur la perception de la parole (échelle de Mels).

Le calcul se déroule comme suit :

- La FFT est calculée sur un fragment (trame).
- Cette dernière est filtrée par un banc de filtres triangulaires répartis le long de l'échelle de Mel.
- Le logarithme du module de l'énergie de sortie du banc de filtres est calculé.
- Une transformation en cosinus discrète inverse (équivalente à la transformée FFT inverse pour un signal réel) est appliquée. Seuls les premiers coefficients sont retenus.

2.2.3 Modélisation acoustique

Le modèle acoustique est un élément central de le SRAP ; il est construit à partir de la modélisation d'une unité élémentaire qui est le phonème ou une variante. Le choix de cette unité est motivé par le fait que le phonème est la plus petite unité sonore identifiable et que tout mot se décline en une ou plusieurs séquences phonétiques, représentatives de sa prononciation. Cela permet d'inclure d'une manière évolutive des nouveaux mots au vocabulaire existant, sans avoir à disposer de la prononciation de tous les mots d'une langue disponible dans les données d'apprentissage. L'estimation des paramètres du modèle acoustique se fait uniquement sur la prédiction des sorties phonétiques possibles pour chaque séquence de mots.

Cette approche phonétique permet de construire un modèle acoustique probabiliste pour la suite de mots, par concaténation de phonèmes et prise en compte des différentes prononciations de chaque mot et des coarticulations possibles.

Les phonèmes de contexte sont modélisés par un modèle de Markov caché gauche et droit à trois états (HMM).

2.2.3.1. Modèles de Markov Cachés :

Les modèles de Markov cachés (Hidden Markov model, HMM) sont aujourd'hui utilisés dans un très grand nombre des systèmes de reconnaissance automatique de la parole. Ces modèles furent introduits par (Baker, 1975).

Un HMM est une machine à état fini, où la transition d'un état ***i*** vers un état ***j*** est effectuée à chaque pas temporel avec une probabilité a_{ij}. Contrairement à une chaîne de Markov classique, nous n'avons pas une connaissance directe de l'état courant. Les états sont alors dits "cachés" et ont une probabilité d'émettre une observation. Ainsi, en entrant dans l'état ***j***, un vecteur d'observation y_t est généré en suivant la densité de probabilité $b_j(y_t)$.

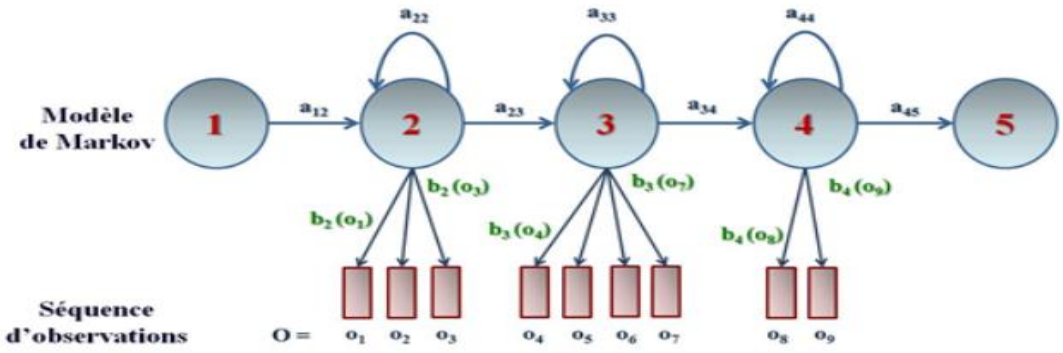


Figure 2.3. Exemple de HMM à 5 états gauche-droit dont 3 émetteurs

Le HMM de la figure 2.3 est définie par 5 états de gauche à droite. Les états 1 et 5, situés respectivement au début et à la fin de la chaîne, représentent l'entrée et la sortie. Ils n'ont pas de probabilité d'émission et permettent d'isoler le phonème pour en faciliter la concaténation. un mot est modélisé alors par la concaténation des modèles HMM de chaque phonème du mot. Les états 2, 3, 4 ont une probabilité d'émission (b_i). Par chaque état émetteur, le HMM peut rester sur l'état courant ou bien passer au suivant. Cette particularité du HMM lui permet de prendre en compte des dictions plus ou moins lentes du phonème concerné.

Un HMM est défini par un ensemble de paramètres $\Phi = (A, B, \pi)$, avec :

- $A = (a_{ij})$ la matrice stochastique de transition.
- $B = (b_j(y))$ la matrice stochastique des probabilités d'émission avec y le vecteur descripteur.

- $\pi = (\pi_i)$ la matrice d'initialisation des états.

Il y a donc un ensemble $M = \{M_1, M_2, \ldots, M_U\}$ d'unités possibles associées à un ensemble de paramètres $\Phi = \{\Phi_1, \Phi_2, \ldots, \Phi_U\}$. L'objectif du modèle acoustique est d'estimer la probabilité que le modèle ***M*** génère la séquence de vecteur acoustique ***Y***.

Pour atteindre cet objectif, trois problèmes majeurs sont à résoudre (Romdhani, 2015) :

- L'évaluation/estimation : déterminer $P(Y|\Phi)$.
- Le décodage : trouver la séquence d'état $S = (S_1, S_2, \ldots, S_T)$ la plus probable qui génère la séquence observée $Y = (Y_1, Y_2, \ldots, Y_T)$.
- L'entraînement : maximiser le produit $\prod_Y P(Y|\Phi)$ en ajustant les paramètres du modèle.

2.2.4 Modélisation linguistique

Généralement, les êtres humains n'éprouvent aucune difficulté à identifier ces mots car ils ont déjà une connaissance préalable des termes appropriés dans leur contexte. Cependant, contrairement à l'ordinateur, qui ne possède pas la capacité humaine de reconnaissance des sons correspondants, et le modèle de langage statistique est utilisé à cette fin pour le SRAP.

Les modèles de langage impliquent l'estimation d'une distribution de probabilité sur une séquence de mots. En d'autres termes, s'il existe une séquence de mots $\boldsymbol{w = (w_1, w_2 \ldots, w_K)}$, le modèle de langage utilise la distribution de probabilité de $\boldsymbol{P(w)}$ sur cette séquence. Selon (Plátek, 2014), le but principal de la modélisation du langage est de minimiser et simultanément de prioriser les hypothèses du modèle acoustique. Ces résultats de probabilité sont additionnés à ceux du modèle acoustique pour calculer la probabilité finale pour l'ensemble de la transcription.

Différents types de modélisation du langage sont utilisés dans le domaine RAP, mais les n-grammes sont considérés comme l'une des approches les plus courantes. n-gram utilise les mots précédents (n-1) pour estimer le mot suivant.

Ce processus d'estimation est appelé Markov, du nom du mathématicien Andrei Markov, qui a inventé le processus pour montrer que la probabilité de chaque mot est basée sur le mot précédent.

Les bi-grammes et les tri-grammes sont des types couramment utilisés dans la modélisation du langage en utilisant des n-grammes dans le domaine de la reconnaissance vocale. Un bi-gramme est créé lorsque la valeur de n dans l'expression "n-gramme" est égale à deux (n = 2), tandis qu'un tri-gramme est créé lorsque la valeur de n est égale à trois (n = 3). Ainsi, un n-gramme permet de prendre en compte le mot précédent (n-1) dans le contexte.

2.2.4. Dictionnaire de prononciation

Le lien entre la modélisation acoustique et la modélisation linguistique est fait par un dictionnaire de prononciation peut être appelé aussi dictionnaire de prononciation. Un tel dictionnaire doit contenir un vocabulaire qui peut être défini comme l'ensemble des mots qu'un SRAP est capable de reconnaître, ainsi que leursprononciations.

Autrement dit, il est nécessaire d'associer chaque entrée du dictionnaire à une suite de phonèmes qui lui est propre. Un phonème peut correspondre à un ou plu- sieurs lettres (graphèmes) différents, cela signifie qu'il est nécessaire de disposer de toutes les séquences de phonèmes correspondant à un mot dans le dictionnaire. Pour convertir les graphèmes en symboles phonétiques, un système appelé conversion graphème en phonème (G2P) est utilisé. Dans la littérature, il existe trois approches qui ont été utilisées pour la transcription phonétique d'un mot donné à savoir, l'approche manuelle, l'approche à base de règles où la connaissance linguistique et phonétique des experts est utilisée pour développer un ensemble de règles et l'approche guidée par les données.

2.2.5. Décodeur

La dernière étape dans le processus de reconnaissance automatique de la parole consiste à intégrer les résultats de la modélisation acoustique et ceux des modèles de langage dans un seul processus de décision permettant de retrouver un message prononcé en entrée. Pour ce faire, un composant principal d'un SRAP est mis en place, il s'agit d'un "décodeur".

Partant des informations contenues dans le dictionnaire de prononciation et des modèles acoustiques et linguistiques, la question de décodage d'un signal de la parole consiste à parcourir l'espace de recherche que représente l'intégralité des séquences de mots possibles à partir du vocabulaire du système et à trouver le meilleur chemin qui donnera la séquence de mots la plus probable. Clairement, cet espace de recherche, est très grand, et représenté sous forme de graphe appelé "graphe de recherche". En plus, il intègre certaines informations

utilisées pour générer les hypothèses telles que les unités acoustiques (phonèmes) associées à leurs scores acoustiques.

Etant donné que l'espace de recherche est très grand et dans le but de n'explorer qu'un espace restreint et suffisant pour trouver la meilleure solution, des algorithmes de recherches sont employés pour choisir à chaque instant un nombre limité d'hypothèses. La stratégie de recherche, dans cet espace d'hypothèse, diffère d'un algorithme à un autre. En effet, il y a des algorithmes qui incluent la stratégie de recherche en "profondeur d'abord" alors que d'autres incluent la stratégie en " largeur d'abord". Ces deux types de stratégies sont relatifs à des parcours différents d'une arborescence de possibilités. Le premier type explore l'arbre en suivant toujours l'hypothèse la plus "prometteuse", tandis que le deuxième examine parallèlement toutes les hypothèses d'un seul niveau.

2.2.6. Post-traitement

Les systèmes de reconnaissance fournissent un texte représentant la transcription d'un signal sonore. Outre la transcription finale, un SRAP peut trouver les N meilleures hypothèses de transcription possibles triées selon leur score total. En pratique, cette liste est souvent limitée aux cinq ou dix meilleures hypothèses, également appelée liste "n-best". Une approche couramment utilisée pour améliorer la précision de la reconnaissance consiste à enrichir cette liste en utilisant des sources d'informations supplémentaires. Cela permet de revoir et d'affiner le résultat de la meilleure hypothèse notée (Gruhn et al., 2011).

2.3. Modèles de bout en bout

Dans les approches basées sur le modèle de Markov caché, trois composants sont cruciaux pour le développement des systèmes de reconnaissance de la parole, à savoir : le modèle acoustique, modèles de langage et dictionnaires (le modèle basé sur ces trois modules est un modèle statistique). Ces composants sont modélisés et entraînés indépendamment, et le décodeur les combine pour générer des séquences de mots. Depuis quelques années, une approche alternative a été proposée qui remplace les trois composantes de la modélisation précédente par un seul modèle basé sur les réseaux de neurones, ce sont le modèle de bout en bout, de bout en bout.

Le principe d'un modèle de bout en bout est d'intégrer tous les modèles dans un seul composant basé sur un réseau de neurones récurrent. Cette nouvelle architecture tire parti de la grande quantité de données disponibles pour rendre l'apprentissage plus efficace, En optimisant mieux l'ensemble du système. Deux méthodes principales sont utilisées pour développer un système de bout en bout (End2End) : l'approche basée sur la classification temporelle connexionniste (CTC) et les approches basées sur les modèles séquence-à-séquence (seq2seq) (Menacer, 2020).

2.3.1 Modèles basés sur CTC

La classification CTC est employée pour entraîner des réseaux de neurones récurrents à étiqueter des séquences sans nécessiter un alignement explicite entre la séquence d'entrée et celle de sortie. Cette approche sans segmentation se rapproche des méthodes d'entraînement des modèles HMM, où l'alignement entre la séquence d'observation et les unités acoustiques (souvent des caractères dans ce cas) repose sur des algorithmes de programmation dynamique tels que l'algorithme de Baum-Welch ou celui de Viterbi. L'avantage par rapport aux modèles HMM est que les réseaux de neurones récurrents permettent de capturer simultanément plusieurs informations, notamment la classification acoustique et la structure linguistique, d'où le terme "modèles de bout en bout" (Menacer, 2020).

2.3.2 Modèle séquence-à-séquence

Modèle séquence-à-séquence (Seq2Seq) permet également au même titre que le CTC de calculer tous les alignements possibles et les regrouper. Le RNN-Transducer, à l'inverse du CTC, ne fait pas l'hypothèse de l'indépendance des sorties, la définition de chemin s'en trouve différente pour prendre en considération la dépendance entre sorties.

Dans un modèle séquence à séquence, il y a deux composants principaux : un encodeur et un décodeur. L'encodeur prend en entrée une séquence d'observations O = o1 :::oT et génère une représentation intermédiaire de cette observation basée sur plusieurs blocs d'un réseau de neurones récurrent.

La valeur intermédiaire de l'entrée est exactement le vecteur produit par le dernier bloc du RNN, que le décodeur utilise pour prédire la séquence de caractères qui correspond le mieux à l'observation O. Cette architecture est décrite par la figure suivant :

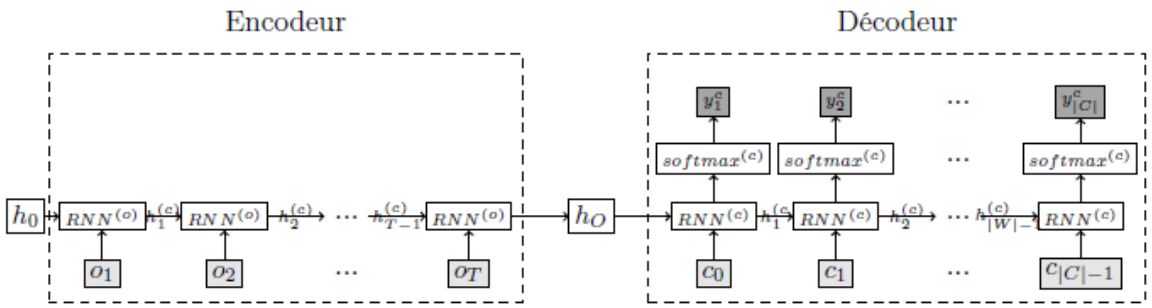


Figure 2.4.Utilisation du mécanisme d'attention dans les modèles séquence-à-séquence.

La modélisation basée sur les deux composants encodeur-décodeur rencontre des difficultés lorsqu'il s'agit de généraliser à de nouvelles séquences d'observations. Cela s'explique par le fait qu'elle repose sur une seule représentation de la séquence d'entrée pour générer la séquence de sortie. Cette représentation intermédiaire est limitée car elle ne contient aucune information explicite sur les parties les plus importantes du signal sur lesquelles on peut se baser pour reconnaître la parole de manière plus précise.

Pour remédier à ce problème, plutôt que d'encoder la séquence d'entrée en un seul vecteur, on utilise plusieurs vecteurs, appelés vecteurs de contexte, pour prédire chaque caractère de la séquence de sortie. C'est là que les techniques d'attention (Bahdanau, Cho, & Bengio, 2014) interviennent. Chaque vecteur de contexte vct est calculé comme illustré dans la figure

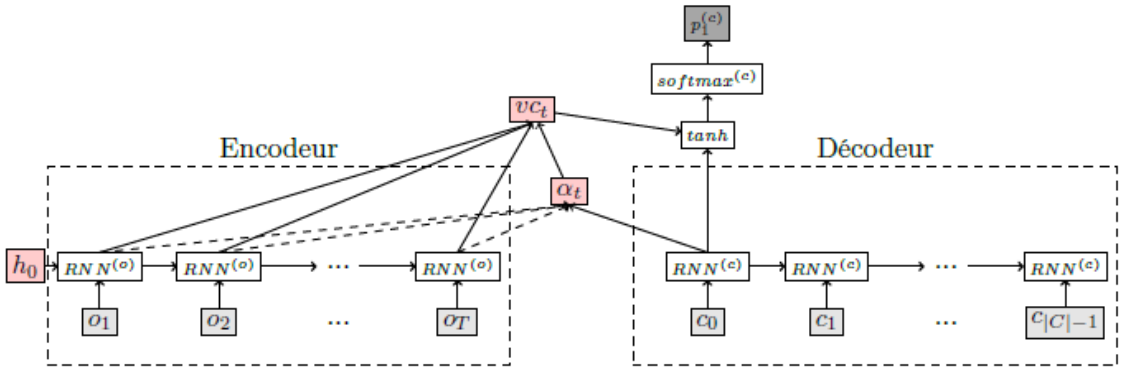


Figure 2.5 . Architecture du modèle séquence-à-séquence.

Avec cette approche, l'alignement entre la séquence d'entrée et celle de sortie est modélisé de manière explicite. Le terme t dans la figure 1.12 représente la probabilité d'alignement du caractère ct avec les observations de la séquence

d'entrée. Ainsi, pour chaque caractère de la séquence de sortie, le réseau de neurones prête attention aux parties les plus importantes du signal qui peuvent contribuer à la génération de cette unité. C'est pourquoi cette technique est appelée "technique d'attention".

L'avantage de ces approches est qu'elles sont indépendantes de la langue, leur mise en place dépend d'une collection de données de la langue à reconnaître, en particulier des données orales et textuelles.

2.4. Modèles Hybrides

Afin d'améliorer les performances des SRAP des modèles hybrides sont proposés, soit deux modèles statistique sont combinés comme le cas des modèles GMM-HMM (Gaussian Mixture Model GMM), soit par la combinaison d'un modèle HMM et un réseaux de neurones profond (HMM-DNN).

2.4.1. Modèles GMM-HMM

Une idée efficace pour optimiser les HMM consiste à intégrer un modèle de mélange gaussien (GMM) au modèle HMM pour créer un modèle "GMM-HMM". L'idée consiste à alimenter le GMM avec les trames audio comme d'habitude afin de regrouper les états, en identifiant les phonèmes ; ces états de regroupement sont transmis aux modèles HMM, chacun d'entre eux essayant alors de créer un modèle statistique

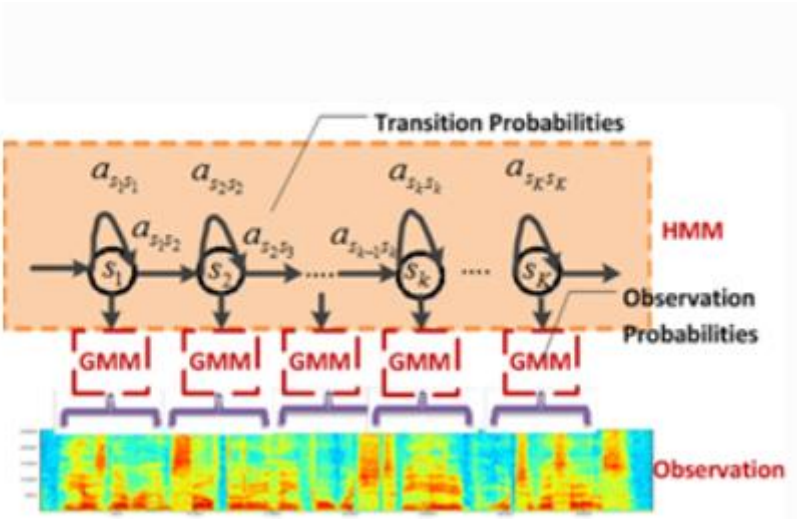


Figure 2.6 – Architecture d'un système GMM- HMM

Les modèles GMM-HMM ont longtemps été considérés comme étant les meilleurs. Cela est dû à leur capacité d'adaptation à des signaux acoustiques variables et à l'efficacité de l'algorithme d'Expectation-Maximization pour l'estimation de ses paramètres. Cependant, ces modèles considèrent une distribution de données gaussienne. Il s'agit là d'une approximation grossière. De ce fait, pour modéliser des données hautement non-linéaires, les GMM requièrent un nombre gigantesque de paramètres.

2.4.2. Modèle HMM - Deep Neural Network

Une solution pour contourner les limites du HMM est le réseau de neurones. En effet, le DNN possède une capacité de classification bien supérieure à celle du GMM. Ainsi, le DNN modélise des frontières non-linéaires entre les différentes classes de sorties, qui sont ici les états cachés du HMM.

Une approche supervisée est utilisée pour entraîner le réseau de neurones. Les données d'entraînement labélisées correspondent aux alignements du modèle HMM-GMM, c'est-à-dire la relation entre les états cachés (que l'on veut déterminer) et les frames émises (que l'on observe)

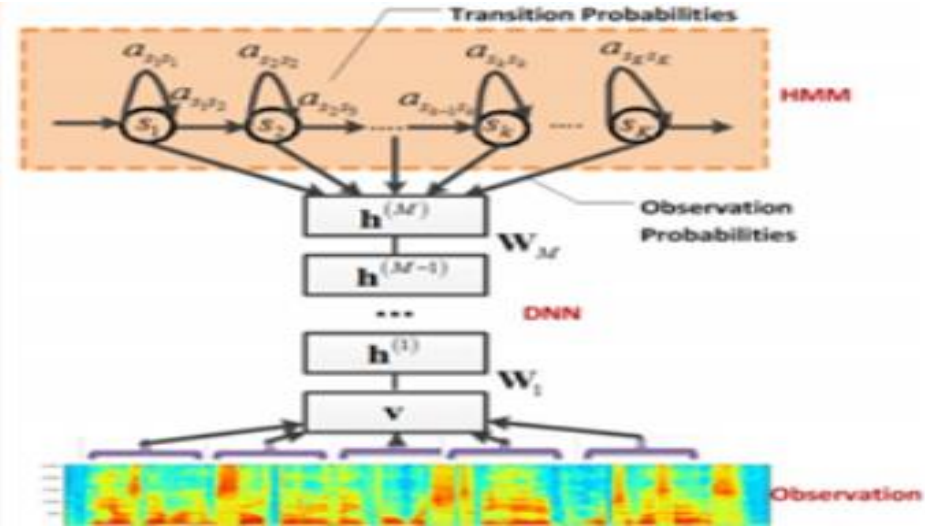


Figure 2.7 – Architecture d'un système HMM-DNN

2.5. Evaluation des systèmes RAP

La performance des systèmes RAP peut être évaluée en termes de différentes mesures, telles que l'erreur, la précision et l'exactitude. En général, les systèmes RAP ont trois types d'erreurs courants : les substitutions, les insertions et les suppressions. La substitution a lieu dans Le système reconnaît les mots qui diffèrent des mots prononcés. L'insertion se produit lorsque les hypothèses de sortie contiennent des mots qui n'ont pas été prononcés par le locuteur. La suppression se produit lorsque des mots parlés sont manqués par les résultats reconnus (Font, 2005). . Une des mesures les plus répandues pour évaluer les performances d'un système de reconnaissance automatique de la parole est le taux d'erreur mot (Word Error Rate WER). Le *WER* consiste à comparer les hypothèses de transcription et la transcription de référence. Pour ce faire, un alignement mot à mot est réalisé entre les deux transcriptions et la comparaison s'effectue selon les différents types d'erreurs sur les mots que peut commettre le système. Formellement, le *WER* est calculé comme suit :

$$WER = \frac{I+S+D}{N} \qquad (2.)$$

- N est le nombre de mots de référence,
- S est le nombre de substitutions (mots incorrectement reconnus),
- D est le nombre de suppressions (mots omis),

- I est le nombre d'insertions (mots ajoutés).

2.6. Revue de littérature sur les systèmes SRAP arabe

Par rapport à la recherche existante sur la SRAP pour la langue anglaise, la SRAP pour la langue arabe a reçu peu d'attention car elle est considérée comme une langue aux ressources limitées. Les principaux défis de la langue arabe restent spécifiques à l'existence d'énormes dialectes aux diverses prononciations, à la complexité morphologique et à la difficulté d'acquérir une transcription diacritisée des textes parlés, qui ne sont pas souvent de open source, etc. Pour construire un système robuste du SRAP en arabe, il est fortement recommandé et plus précis d'utiliser de vastes collections de paroles.

La plupart travaux sont focalisés à l'arabe standard, 89,47% des études retenues portent sur l'arabe standard moderne, alors que 26,32% d'entre elles sont consacrées à différents dialectes de l'arabe. MFCC et HMM ont été présentés comme les techniques d'extraction et de classification de caractéristiques les plus utilisées, respectivement : 63 % des articles étaient basés sur MFCC et 21 % étaient basés sur HMM. (Dhouib, Othman, El Ghoul, Khribi, & Al Sinani, 2022).

Les réseaux de neurones récurrents (RNNs) comptent parmi les meilleurs modèles appliqués aux données séquentielles. Ils permettent à la fois la propagation avant et la propagation arrière, ce qui est bien adapté aux données de la parole, qui peuvent être considérées comme des séquences temporelles. Dans l'étude (Alotaibi, 2004),un SRAP basé sur les RNNs a été conçu et testépour reconnaitre les dix chiffres arabes (de zéro à neuf). L'architecture RNN proposée a atteint 99,5% de reconnaissance correcte des chiffres dans le cas du mode multi-locuteurs et 94,5% dans le cas du mode indépendant du locuteur.

Une autre application des RNNs est proposée dans (El Choubassi, El Khoury, Alagha, Skaf, & Al-Alaoui, 2003), où les auteurs ont présenté une nouvelle approche pour implémenter un SRAP pour la parole isolée. Ils ont utilisé

un RNN Elman modulaire, c'est-à-dire que pour chaque mot de l'ensemble du vocabulaire, un RNN séparé est appliqué. La modularité adopte une approche "diviser pour régner" en divisant le problème complexe en plusieurs problèmes bien plus simples. Le vocabulaire utilisé est composé de 6 mots arabe : "manzel" (maison), "hirra" (chat), "chajara" (arbre), "tariq" (route), "ghinaa" (chant), "zeina" (zeina étant un nom propre). L'apprentissage est divisé en deux étapes : un apprentissage cohérent composé de 48 énoncés et un apprentissage discriminant possédant 20 énoncés. Les résultats obtenus, entre 85% et 100%, ont été comparés avec ceux des HMMs.

Les réseaux de neurones profonds (DNNs) sont des modèles d'apprentissage machine récents et extrêmement puissants. Récemment, ils sont employés dans le domaine de le SRAP.

Dans (AbdAlmisreb, Abidin, & Tahir, 2015), les auteurs ont proposé une architecture DNN entièrement connectée avec des unités Maxout. La technique MFCC a été utilisée pour l'extraction des caractéristiques du signal de parole. L'apprentissage et le test du DNN ont été effectués sur un jeu de données composé de phonèmes arabe consonantiques enregistrés à partir de 20 locuteurs Malais. Les résultats de l'architecture proposée a été comparés avec la machine Boltzmann restreinte (Restricted Boltzmann Machine : RBM), le réseau de croyance profond (DBeN), le réseau neuronal convolutif (CNN), le ANN et à l'auto-encodeur convolutif (Convolutional Auto-Encoder : CAE).

Les auteurs dans (Elaraby & Abdul-Mageed, 2018) ont testé 6 techniques DNN différentes sur un SRAP, en comparant les performances à plusieurs modèles d'apprentissage machine classiques selon le type de problème à classifier (classification binaire et multi-classes). Les

résultats expérimentaux ont montré que les variantes des RNNs bidirectionnels ont atteint la meilleure précision sur le jeu de données commentaire arabe en ligne (Zaidan & Callison-Burch, 2011). Celui-ci représente un référentiel à grande échelle de dialectes arabes avec des étiquettes manuelles pour 4 variétés de dialecte. Les résultats obtenus ont surpassé de manière significative toutes les méthodes de base concurrentielles.

Dans l'étude (Zerari, Abdelhamid, Bouzgou, & Raymond, 2019), les auteurs ont proposé un SRAP basé sur un DNN bout en bout qui s'appuie sur des architectures LSTM/GRU pour reconnaitre des commandes TV vocales en langue arabe. Les caractéristiques extraites par MFCC ont été introduites aux différentes structures profondes à savoir : avant, arrière et bidirectionnelle. Ensuite un MLP a été adopté pour classifier les commandes. Le système proposé a donné des précisions individuelles de reconnaissance correcte allant de 95.98% jusqu'à 99.64 %.

2.7. Les outils de développement

Aujourd'hui, il existe beaucoup d'outils de développement pour les systèmes de reconnaissance automatique de la parole. Il y a des outils avec code fermé, que sont présentés sur le tableau 2.1. D'autres outils sont en open source comme le HTK a été mis en œuvre à la fin des années 1980 et est maintenu par le Speech Vision and Robotics Group du Cambridge University Engineering Department (CUED) et l'une des boîtes à outils les plus populaires est CMU Sphinx, conçue pour les applications mobiles et les applications serveur. (Fendji, Tala, Yenke, & Atemkeng, 2022); Kaldi est également un outil de reconnaissance vocale open-source, écrit en C++ (Povey et al., 2011) et Wit qu' est gratuit même pour un usage commercial. Il prend en charge plus de 130 langues.

D'après les études comparatives été faites par Këpuska et.al (Këpuska & Bohouta, 2017) et Filippidou et al (Filippidou & Moussiades, 2020), L'API Google Speech est plus performant

par rapport aux autres outils ou Google a amélioré sa reconnaissance vocale en utilisant un DNN dans ses applications, atteignant un taux d'erreur de 8% en 2015 contre 23% en 2013. Et le taux de erreur WER avec sphinx-4 est de 37 % , et par API Microsoft et 18%.

Boîte à outils /1ére version	Langage de programmation	Langage cible	Applications	Technologie appliquée
Microsoft Speech API, 1995	C#	Anglais, chinois, japonais	Reconnaissance vocale, synthèse vocale dans les applications Windows	DNN dépendant du contexte, HMM
SiriKit, 2011	Objective-C	Anglais, allemand, Français	Reconnaissance vocale	DNN et ML
Dragon Mobile SDK, 2011	C++	Anglais, espagnol, français, allemand, italien, japonais	Reconnaissance vocale et synthèse vocale « texte-parole »	Processus de Markov
Yandex Speechkit, 2014	C, Python	Russe, anglais, ukrainien, turc	Reconnaissance vocale, synthèse vocale, identification musicale et activation vocale, « texte-parole »	Réseau de neurones
Google Speech Recognition API, 2016	C#	Anglais, plus de 120 langues	Parole -texte, reconnaissance de la parole	DNN

Tableau 2.1 Liste non exhaustive de travaux/outils de reconnaissance vocale à code source fermé (Fendji et al., 2022).

2.8. La traduction automatique de la parole

2.8.1 Définition

La traduction automatique est le processus de conversion automatique d'un texte donné dans une langue source en texte dans une langue cible, tout en préservant le sens du texte d'entrée et en produisant un texte fluide dans la langue cible. Le processus n'est pas une simple substitution mot à mot, c'est beaucoup plus compliqué. Le traducteur doit interpréter et analyser tous les éléments du texte source et savoir comment chaque mot affecte un autre mot pour finalement générer le texte cible

2.8.2 Les approche de la traduction automatique de la parole

Une fois la reconnaissance de la parole arabe terminée, le texte correspondant est utilisé pour la traduction automatique. La littérature propose plusieurs approches pour automatiser ce processus, parmi lesquelles on trouve la modélisation statistique et la modélisation basée sur les réseaux neuronaux.

2.8.2.1 Modélisation statistique de la traduction automatique

L'approche statistique de la traduction automatique s'inspire de la modélisation statistique utilisée dans les systèmes de reconnaissance automatique de la parole. Elle repose sur deux modèles principaux : le modèle de traduction, qui permet de trouver la traduction d'un mot ou d'une expression de la langue source vers la langue cible, et le modèle de langage, qui garantit que la traduction est correctement formulée dans la langue cible. Ces deux modèles sont utilisés conjointement par le décodeur, qui explore l'espace de recherche pour trouver la meilleure traduction possible.

Cependant, pour les langues présentant une morphologie complexe ou un ordre de mots relativement flexible, telle que la langue arabe, ces deux modèles ne sont pas suffisants. Des modèles complémentaires sont intégrés dans le processus de décodage afin de capturer d'autres caractéristiques linguistiques. Parmi ces modèles complémentaires, on trouve le modèle de réordonnancement, qui prend en compte les variations d'ordre des mots entre les langues, et le critère de longueur de traduction, qui pénalise les traductions trop longues en fonction du nombre de mots utilisés (Menacer, 2020).

2.8.2.2 Modélisation basée sur les réseaux de neurone

Le modèle était basé sur une architecture séquence-à-séquence, mais il y a une différence dans la représentation des entrées et des sorties. Dans le cas de la traduction automatique, le modèle

prend la phrase source en entrée. Chaque mot de la phrase est transformé en une représentation numérique dans un espace vectoriel, qui est ensuite utilisée comme entrée pour l'encodeur.

Le travail de (Menacer, 2020) n'était pas compétitif par rapport à l'approche statistique, en raison d'une modélisation insuffisante des alignements entre les mots de la phrase source et ceux de la phrase cible. En effet, la génération de la phrase cible se basait uniquement sur une représentation fixe de la phrase source, sans prendre en compte les parties les plus importantes de la phrase source qui devraient guider la production des mots de la phrase cible. Ce problème a été résolu grâce aux modèles d'attention proposés par Bahdanau et al (Bahdanau et al., 2014).

Au fil des années, de nombreuses avancées ont été réalisées dans ce domaine. Dans une de ces avancées, Google (Menacer, 2020) a proposé son propre modèle basé sur l'architecture séquence-à-séquence. Leur modèle utilise huit couches de réseaux de neurones récurrents pour encoder la phrase source et la décoder. L'utilisation du modèle d'attention est une caractéristique clé de ce modèle pour générer les mots de la phrase cible.

Une autre caractéristique importante de ce modèle est sa capacité à traduire les mots inconnus. Étant donné que le modèle est entraîné sur un vocabulaire limité, tous les mots qui ne font pas partie de ce vocabulaire sont remplacés par un mot spécial. Cependant, pour surmonter ce problème, les auteurs ont proposé de décomposer les mots inconnus en unités plus petites, telles que des caractères, et la traduction est ensuite réalisée au niveau de ces caractères. Ce modèle est utilisé dans la plate-forme de traduction en ligne de Google.

2.6. Conclusion

Nous avons présenté au cours de ce chapitre les modèles de reconnaissance et de traduction automatiques de la parole. En plus nous avons cité les services basés sur l'apprentissage profond avec RAP arabe. Le prochain chapitre sera consacré à la conception et l'implémentation de notre modèle de classification.

Chapitre 3

Mise en œuvre

3.1.Introduction :

Dans les chapitres précédents, nous avons présenté la procédure de reconnaissance de la parole, le principe de fonctionnement d'un système de reconnaissance de la parole et la conversion en texte éditable.

Le but de ce chapitre est de décrire de manière détaillée les étapes de mise en œuvre de l'application Windows suggérée pour la reconnaissance vocale. Nous avons créé une interface graphique pour l'extraction des informations acoustiques de différentes sources telles que le microphone (temps réel), les fichiers audio (mp3), et les vidéos (mp4).

Nous commençons par une représentation des principaux outils informatiques utilisées pour créer cette application. Ensuite nous décrirons en profondeur chaque étape d'exécution.

3.2. Présentation d'environnement de développement utilisé

D'abord, Nous commençons par présenter l'environnement de développement et les ressources utilisés, Ensuite, nous abordons la base de données qui a été utilisée pour l'entraînement et l'évaluation de notre modèle, suivie des différentes étapes de sa réalisation.

3.2.1. Environnement matériel

Afin de réaliser ce projet l'opération a été effectuée sur un ordinateur portable « HP » avec

Les caractéristiques suivantes

- **Processeur** : Intel(R) Core (TM) i3-4005U CPU @ 1.70GHz 1.70 GHz
- **Mémoire installé** :4,00 Go
- **Type du système** : Système d'exploitation 64 bits, processeur x64

3.2.2 Environnement logiciel

3.2.2.1 Langage de développement

Il existe une vaste gamme de langages de programmation, chacun ayant ses avantages et ses inconvénients, Il faut bien en choisir un, notamment dans notre cas de reconnaissance automatique de la parole avec transformation en texte (Speech-to-Text) plus la traduction, le langage de programmation approprié et récent c'est Python.

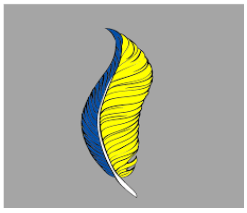
Python : Python est en effet un langage de programmation de haut niveau interprété, ce qui signifie qu'il ne nécessite pas d'étape de compilation. Il est largement utilisé par une vaste communauté de développeurs. Python se distingue par sa simplicité et sa facilité d'apprentissage. Les packages Python en bioéthique favorisent la modularité et la réutilisabilité du code. De plus, Python et ses bibliothèques sont disponibles gratuitement en source ou en binaires pour la plupart des plateformes et peuvent être redistribués sans frais Voici quelques-unes des fonctionnalités et avantages offertes par Python et ses

Bibliothèques :

- Faire circuler des informations au travers d'un réseau.
- Dialoguer d'une façon avancée avec le système d'exploitation.
- Produire de petits programmes très simples, appelés scripts, chargés d'une mission très précise sur un ordinateur.
- Créer des interfaces graphiques.
- Lisibilité des codes.
- Rapidité de développement.
- Auto-documentation

3.2.2.2 Bibliothèques utilisées

Dans notre programme Python, nous avons utilisé les bibliothèques suivantes **:**



Tkinter : Tkinter est une bibliothèque standard de Python pour créer des interfaces graphiques (GUI). Elle permet aux développeurs de concevoir des applications interactives et esthétiques. Tkinter fournit une interface Python pour Tk, un ensemble d'outils d'interface utilisateurs. Les développeurs peuvent créer des fenêtres, des boutons, des champs de saisie et bien plus encore. Avec Tkinter, les interfaces utilisateurs peuvent être personnalisées en utilisant des propriétés telles que la couleur et la taille. La documentation officielle de Python pour Tkinter offre une référence complète sur les classes et les méthodes. [Documentation officielle de Python pour Tkinter - https://docs.python.org/3/library/tkinter.html]

PIL : PIL (Python Imaging Library) est une bibliothèque Python largement utilisée pour le traitement et la manipulation d'images. Elle fournit un large éventail de fonctionnalités telles que le chargement, la modification, la création et l'enregistrement d'images dans différents formats. PIL permet de réaliser des opérations courantes sur les images telles que le redimensionnement, la rotation, le recadrage, l'ajustement des couleurs et bien plus encore. Elle offre également des fonctionnalités avancées comme le filtrage d'images, la création de miniatures et la manipulation des canaux de couleur. PIL est une bibliothèque populaire et largement documentée, ce qui facilite son utilisation dans les projets nécessitant le traitement d'images en Python. [Site officiel de Pillow (fork de PIL) - https://pillow.readthedocs.io/]

- **Speechrecognation** : Speech Recognition est une bibliothèque Python qui permet la reconnaissance vocale en convertissant les signaux audio en texte. Elle offre une interface simple pour capter et traiter des données audio à partir de différentes sources telles que des fichiers audio, des flux en direct ou des périphériques d'enregistrement. Speech Recognition prend en charge plusieurs méthodes de reconnaissance vocale telles que Google Speech Recognition, CMU Sphinx, Wit.ai et bien d'autres, offrant ainsi la flexibilité de choisir la méthode adapté à nos besoins. Elle permet également de configurer des paramètres de reconnaissance tels que la langue, le seuil de confiance et les limites de durée. Speech Recognition est une bibliothèque pratique pour intégrer la reconnaissance vocale dans les projets Python**.** [Documentation officielle de Speech Recognition - https://pypi.org/project/SpeechRecognition/**]**

MoviePy **MoviePy :** MoviePy est un module Python pour l'édition vidéo, qui peut être utilisé pour des opérations de base (comme les coupes, les concaténations, les

insertions de titres), la composition vidéo (aussi appelée édition non linéaire), le traitement vidéo, ou pour créer des effets avancés. Il peut lire et écrire les formats vidéo les plus courants, y compris le format GIF. [Documentation officielle de MoviePy - https://zulko.github.io/moviepy/]

Pydub

Pydub : Pydub est une bibliothèque Python conçue pour faciliter la manipulation et la conversion de fichiers audio. Elle offre une interface conviviale pour découper, fusionner, ajuster le volume, convertir le format et effectuer d'autres opérations sur les fichiers audio. Pydub prend en charge divers formats audio courants tels que MP3, WAV, FLAC, etc., et permet de lire et d'écrire des fichiers audios. Grâce à sa simplicité d'utilisation, Pydub permet aux développeurs de réaliser facilement des tâches d'édition audio, telles que la création de sonneries, l'extraction de segments audio, la conversion de formats, etc. [Documentation officielle de Pydub - https://pydub.com/]

- **GoogleTrans :** Googletrans est une bibliothèque Python qui permet la traduction de texte en utilisant les services de traduction de Google. Googletrans utilise l'API de traduction de Google pour effectuer les traductions et prend en charge de nombreuses langues. Cette bibliothèque permet également de détecter automatiquement la langue source du texte à traduire. Grâce à Googletrans, les développeurs peuvent facilement intégrer des fonctionnalités de traduction dans leurs projets Python, que ce soit pour des applications, des outils ou des scripts nécessitant la traduction automatique de texte. [Documentation officielle de Googletrans - https://py-googletrans.readthedocs.io/]

Os : La bibliothèque os (Operating System) est une bibliothèque standard de Python qui fournit des fonctionnalités pour interagir avec le système

d'exploitation. Elle permet d'accéder aux fonctionnalités spécifiques du système, comme la manipulation de fichiers et de répertoires, la gestion des processus, la manipulation des chemins de fichiers, et bien plus encore. La bibliothèque os fournit une interface portable pour interagir avec le système d'exploitation, ce qui permet aux développeurs d'écrire du code qui fonctionne de manière cohérente sur différentes plates-formes. Grâce à os, les développeurs peuvent créer, supprimer, déplacer, renommer des fichiers, exécuter des commandes du système, et gérer les environnements et variables système. [os: Documentation officielle de Python pour la bibliothèque os - https://docs.python.org/3/library/os.html]

3.3. Présentation de l'application de reconnaissance de la parole

L'objectif de notre travail est de développer une application Windows pour convertir le signal audio en texte éditable et de le traduire vers une autre langue.

Afin de réaliser notre application, nous avons utilisé le langage python avec la bibliothèque Tkinter qui nous a permet de créer l'interface graphique avec les différents éléments et boutons figure 3.1.

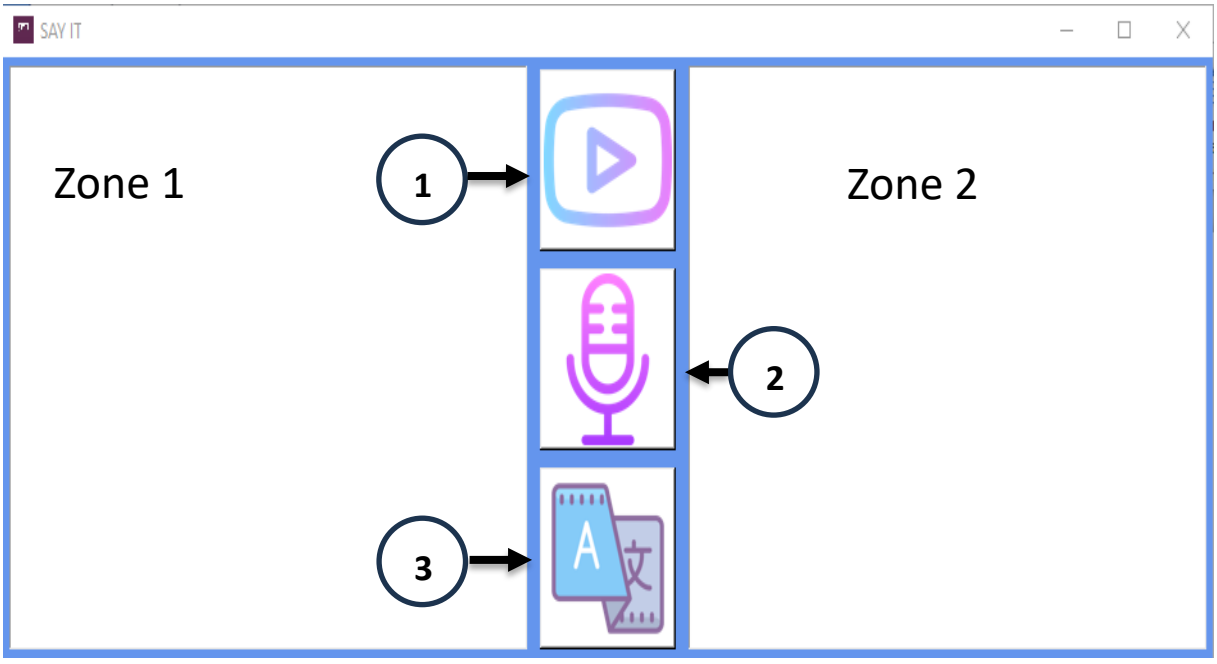


Figure 3.1. La fenêtre principale de notre application de RAP arabe

- Le premier bouton pour sélectionner un fichier audio (mp3) ou un fichier vidéo (mp4) comme une source ;
- Le deuxième bouton pour sélectionner le microphone ;

- Le troisième bouton est utilisé pour lancer la traduction du texte de l'arabe vers une autre langue ;
- La zone 1 : Sert à visualiser le texte extrait de la source (microphone, audio, vidéo) ;
- La zone 2 : Sert à visualiser le texte traduit de l'arabe vers la langue choisie.

3.1. Les actions des boutons de notre logiciel

Dans cette partie nous allons présenter les commandes python qui nous permettent de convertir la parole de chaque source de signal vocale en texte.

Pour effectuer la reconnaissance vocale en Python, il faut installer et importer le package de Speech Recognition.

Speech Recognition est installé depuis un terminal avec « pip » utilisant la commande suivante :

```
pip install SpeechRecognition
```

Pour convertir la parole en texte, la seule et unique classe utilisée du module speech_recognition est la classe Recognizer. Selon l'API cette classe inclue les méthodes suivantes :

recognize_bing(): Utilise l'API Microsoft Bing Speech

recognize_google(): Utilise l'API Google Speech

recognize_google_cloud() : Utilise l'API Google Cloud Speech

recognize_houndify(): Utilise l'API Houndify de SoundHound

recognize_ibm(): Utilise l'API IBM Speech to Text

recognize_sphinx(): Utilise l'API PocketSphinx

Nous avons mentionné dans le chapitre précèdent que la méthode API Google Speech et la plus robuste par rapport à d'autres méthodes. Pour cette raison nous l'avons choisi dans notre application.

3.1.1 Reconnaissance vocale à partir des fichiers audio

Généralement les fichiers audio sont enregistrés en format « mp3 » mais la bibliothèque speech recognation fonctionne qu'avec le format « wav », il faut donc transformer ces fichiers audio d'extension « mp3 » en fichiers d'extension « wav » (figure 3.2).

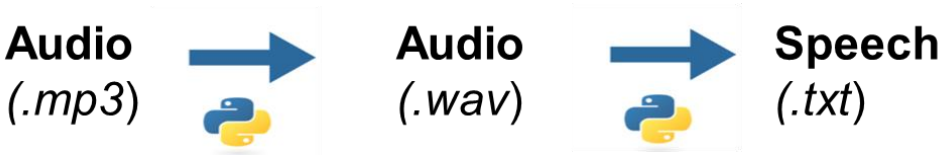


Figure 3.2 : schéma de conversion audio mp3 en texte

Nous avons utilisé la bibliothèque « pydub » pour effectuer cette conversion elle permet d'ouvrir plusieurs formats audio et vidéo multimédia et les manipuler.

Pour lire le fichier MP3 la fonction « form_mp3() » est utilisée. Ensuite, la fonction export() est appliquée pour exporter ce fichier, où le format « wav » est spécifiée dans l'argument format. Les commandes nécessaires pour cette tâche sont illustrées dans la figure 3.3.

```
from pydub import AudioSegment

# files
src = "transcript.mp3"
dst = "test.wav"

# convert wav to mp3
audSeg = AudioSegment.from_mp3("transcript.mp3")
audSeg.export(dst, format="wav")
```

Figure3.3 : programme Python pour la conversion du fichier mp3 en fichier « wav ».

L'étape suivante est la reconnaissance de la parole et la conversion en texte, où nous avons utilisé les lignes des commandes 11et 24 visualisées dans la figure 3.4.

- La fonction AudioFile('test.wav') est utilisé pour prendre le fichier "test.wav" comme source audio ;
- La fonction record(source) : permet l'extraction des données audio du fichier.

- `recognize_google()` est la méthode utilisée pour transcrire nos fichiers audio où les arguments en entrée sont les fichiers audio et la langue arabe dans notre cas (Language='ar-AR') (ligne 16 du script).

```
1  import speech_recognition as sr
2  from pydub import AudioSegment
3  # files
4  src = "transcript.mp3"
5  dst = "test.wav"
6
7  # convert wav to mp3
8  audSeg = AudioSegment.from_mp3("transcript.mp3")
9  audSeg.export(dst, format="wav")
10
11 r=sr.Recognizer()
12 audio=sr.AudioFile('test.wav')# utiliser "test.wav" comme source audio
13 with audio as source:
14     audio_file=r.record(source) #extraction des données audio du fichier
15 try:
16     t=r.recognize_google(audio_file,language='ar-AR')
17     print(t)
18     f=open('text.txt','a',encoding='utf-8')
19     f.writelines(t)
20     f.close()
21 except sr.UnknownValueError as u:
22     print(u)
23 except sr.RequestError as r:
24     print(r)
```

Figure 3.4 Script pour la conversion du fichier audio en fichier texte

3.1.2 Reconnaissance vocale à partir des fichiers vidéo

Le fichier vidéo nécessite un prétraitement avant de l'utiliser comme une source de signal acoustique.

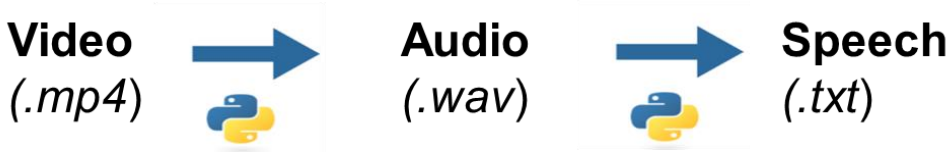


Figure 3.5: Schéma de conversion du mp4 en texte

En premier temps les données acoustiques sont extraites, nous avons utilisé la fonction « VideoFileClip » de la bibliothèque MoviePy pour les enregistrer en fichier audio de forme « wav » comme montré dans le programme de la figure 3.6 lignes 4 et ligne 5.
La suite du programme et qui concerne la reconnaissance de la parole reste inchangeable.

```python
import speech_recognition as sr
import  moviepy.editor as mp
#Extraction  du signal audio à partir du fichier vidéo
clip=mp.VideoFileClip(r'input.mp4')
clip.audio.write_audiofile(r'input.wav')
#Reconnaissance de la parole à partir de l'audio extrait
r=sr.Recognizer()
audio=sr.AudioFile('output.wav')
with audio as source:
    audio_file=r.record(source)
try:
    t=r.recognize_google(audio_file,language='ar-AR')
    print(t)
    f=open('text.txt','a',encoding='utf-8')
    f.writelines(t)
    f.close()
except sr.UnknownValueError as u:
    print(u)
except sr.RequestError as r:
    print(r)
```

Figure 3.6: Les lignes des commandes Python pour convertir une mp4 en texte

3.2.3 Reconnaissance vocale à partir d'un microphone

Maintenant, la source de l'audio à transcrire est un microphone. Pour capturer l'audio d'un microphone, nous avons utilisé la classe Microphone du module `Speach_Recogniton`, comme illustré dans la figure 3.7.

```python
import speech_recognition as sr
mic = sr.Microphone()
    with mic as audio_file:
    print("Dites quelque chose...")

    recog.adjust_for_ambient_noise(audio_file)
    audio = recog.listen(audio_file)
    print("Convertir Speech to Text: ")
```

Figure 3.7 Les linges des commandes pour capter l'audio d'un microphone

Donc la commande sr.Microphone() indique que la source du signal audio est le microphone, pour éliminer le bruit la commande « recog.adjust_for_ambient_noise » est appliquée. La fonction « listen » sert à extraire les données audio du signal provenant du microphone et qui peuvent ensuite être passées à la méthode `recognize_google()` afin de transcrire la parole.

3.2.4 Changement de langue

Nous avons installé l'API Google Translate à l'aide du terminal avec la commande :

pip install googltrans puis nous avons ajouté au programme la partie permettant la traduction du texte.

```
# fonction pour changer la langue de destination
def change_language():
    new_lang = tk.simpledialog.askstring("Change language", "Enter destination language code (e.g. fr for French):")
    if new_lang:
        translator = Translator()
        textmtarjam = translator.translate(text_box.get("1.0", tk.END), dest=new_lang).text
        translated_text_box.delete("1.0", tk.END)
        translated_text_box.insert("1.0", textmtarjam)
```

Figure3.8 : programme Python pour effectuer la traduction

- new_lang = tk.simpledialog.askstring("Changer la langue", "Entrez le code de la langue de destination (par exemple, fr pour le français) :") : Cette ligne ouvre une boîte de dialogue où l'utilisateur peut saisir le code de la langue de destination. Le message "Changer la langue" s'affiche et l'utilisateur doit saisir le code correspondant à la langue voulue. La réponse de l'utilisateur est stockée dans la variable new_lang.
- if new_lang: : Cette ligne vérifie si la variable new_lang contient une valeur (c'est-à-dire si l'utilisateur a saisi un code).

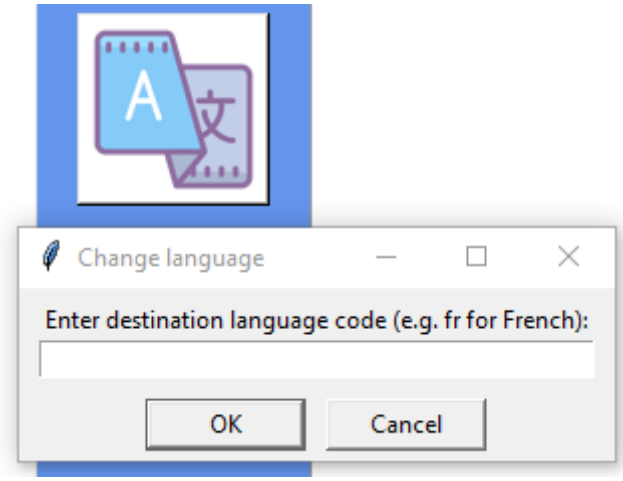


Figure3.9 : Sélection de la langue dans l'interface graphique

3.4. Conversion du programme en fichier exécutable

Les applications ou scripts développés avec le langage Python peuvent être convertis en exécutables pour le système d'exploitation Windows. De cette manière, ils sont utilisables

sans devoir installer Python et sont ainsi mis à la disposition du plus grand nombre. Il est possible de réaliser cette conversion avec différents modules créés dans ce but.

Notre code python a été convertie en fichier exécutable à l'aide de l'outil « Auto py to exe ». Cela permet de créer une version autonome d'une application qui peut s'exécuter sur n'importe quel ordinateur sans installer Python ou d'autres dépendances. Pour convertir l'interface on va suivre les étapes suivantes :

- Étape 1. installation

Pour installer l'application, nous utilisant l'instruction suivante :

pip install auto-py-to-exe

Pour ouvrir l'application, on lance cette ligne dans cmd :

auto-py-to-exe

- Étape 2. Conversion

Il y a quelques options principales que nous devons choisir :

a) Choisissons notre fichier python ensuite l'option **« One Directory » option.**

Assez simple. Lorsque nous choisissons " **One Directory** ", l'option "Auto PY vers EXE" mettra toutes les dépendances dans un dossier. Nous pouvons choisir le répertoire de sortie dans le menu "Avancé

b) Pour Consol Windows : nous choisissons l'option « **console Bassed** »

c) Pour Icon :permet d' insérer une icône au fichier exécutable (la photo en Icon obligé)

d) Pour additionnelle fil : nous sélectionnons les quatre photos utilisés.
(La Photo de microphone, photo d'audio, photo de traduction, logo Icône pour le fichier exécutable).

e) Pour Advanced et dans la case 'Name' nous donnons un nom à notre fichier exécutable « Say it ».

f) Dans settings nous mettons le dossier qui contient toutes les images plus le programme.

- Étape 3 : Exécution du programme.

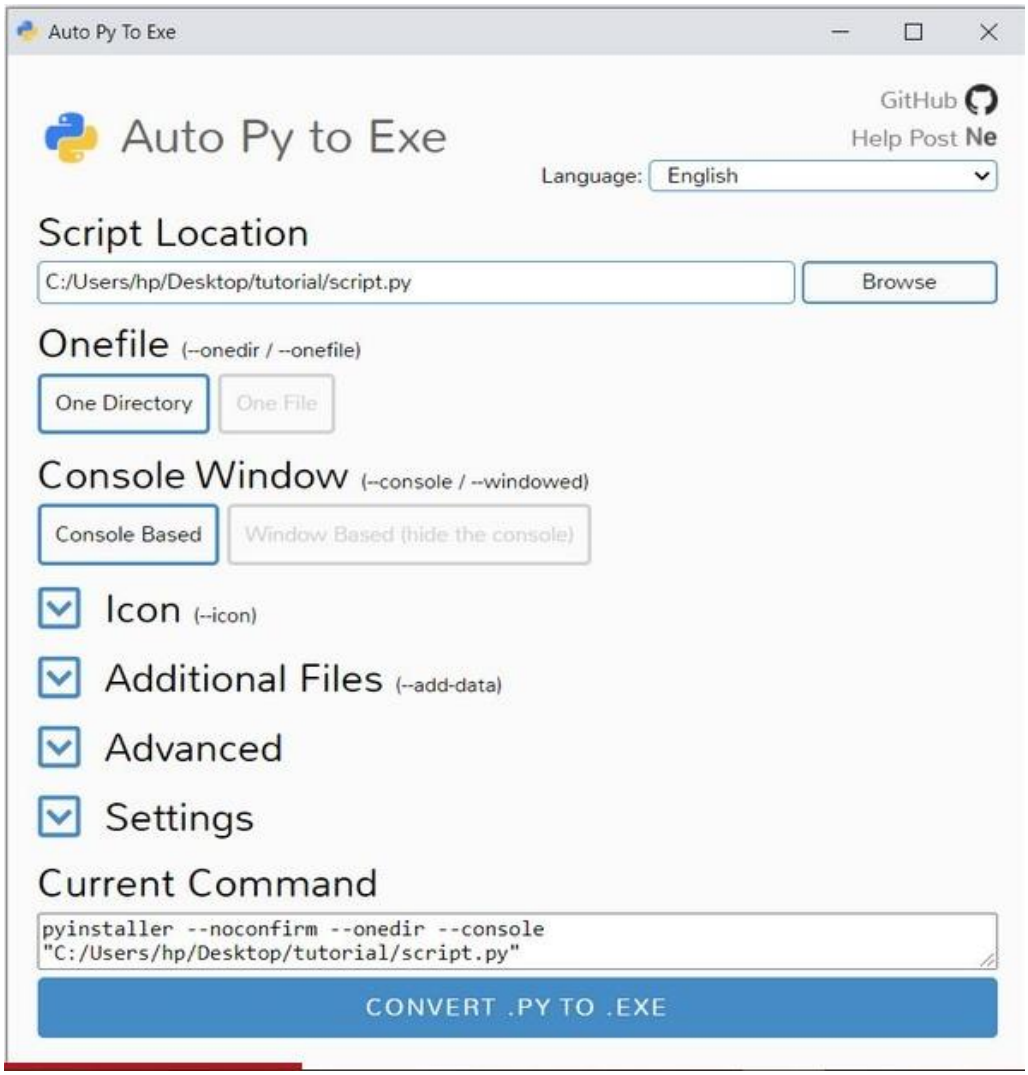


Figure 3.10 : l'outil Auto py to exe

L'exécution de l'application se fait simplement en double-cliq sur le fichier exécutable, ce qui ouvre une interface utilisateur similaire à celle de l'application Python. Les fonctionnalités clés, telles que la capture de la parole en temps réel, l'ouverture de fichiers audio existants, la conversion de la parole et des fichiers audio au format MP3 et des vidéos au format MP4, la conversion en texte, ainsi que la possibilité de changer de langue, sont toutes disponibles.

Cette version exécutable facilite le déploiement de l'application sur différentes machines sans nécessiter de configuration spécifique de Python. De plus, elle simplifie le partage de l'application avec d'autres utilisateurs qui peuvent l'exécuter sans aucune connaissance en programmation.

Figure3.11 : Icône Windows de notre application

3.5. Résultats et discussions

Nous allons convertir le message vocal à partir d'une vidéo, le résultat est illustré dans la Figure 3.11. Nous calculons par la suite le taux d'erreur.

Figure3.1.12 : Le texte extrait à partir d'une vidéo

- **Text de reference**

لم نكن نتخيل في تلك الساعه ان هذا سيكون اخر عهدنا بالعيش في هذا البيت وان العوده اليه او حتى مشاهدته ستكون حلما جزءا من الحلم الفلسطيني الطويل كنا نعتقد على نحو ما ان النكبه الكبرى لا يمكن ان تقع واذا وقع منها شيء فلن يكون غير رحله عابره يجب ان تكون موازين الكون قد اختلت لكي تضيع فلسطين عروبه فلسطين وموازين الكون معا هي القيامه الان.

- **Calcule de l'erreur**

WER=(***I***+***S***+***D***)/***N***

WER=(0+2+0)/67

WER=0.029 , soit 2,9%

- Dans ce deuxième cas, nous allons convertir une vidéo plus complexe contenant des mots qui se ressemblent, ‘Figure3.12’ puis nous comparons le texte après conversion avec le texte de référence pour savoir le taux d'erreur.

Figure3.13 : Le texte extrait d'une vidéo

- **Text de reference**

قيل الى للأمام الشافعي رحمه الله يا امام دلنا على واجب واوجب وعجيب واعجب وصعب واصعب وقريب واقرب فقال واجب ان الناس يتوبوا ولكن ترك الذنوب اوجب والدهر في تصرفه عجيب وغفله الناس عنه اعجب والصبر عند المصائب صعب ولكن فوات الثواب اصعب وكل ما تتمنى قريب والموت من دون ذلك اقرب

- **Calcule de l’erreur**

WER=(***I***+***S***+***D***)/***N***

WER=(1+2+1)/52

WER=0.057 soit 5,7%

- Dans ce troisième cas nous allons convertir un discours en arabe standard issu d'un microphone' Figure 3.13'. De la même manière nous calculons l'erreur.

Figure3.14 : texte en arabe standard extrait d'un signal acoustique issu d'un microphone

- **Text de reference**

انا الطالبة فقيه يسرى هذه واجهة برنامجي الذي يعمل على تحويل النص المنطوق الى نص منسوخ

- **Calcule de l'erreur**

WER=(***I***+***S***+***D***)/***N***

WER=(0+1+0)/16

WER=0.062 Soit 6,2%

Dans ce cas, nous utiliserons un microphone pour enregistrer un discours en dialecte algérien de l'arabe. Le résultat est représenté dans la Figure 2, l'erreur calculé est égale à 0.

Figure3.15 : texte en dialecte algérien à partir d'un microphone

- **Text de reference**

انا الطالبه فقيه يسرى هادي الواجهه نتاع البرنامج نتاعي اللي يخدم على تحويل النص اللي هضرناه الى نص مكتوب

- **Calcule de l'erreur**

WER=(***I***+***S***+***D***)/***N***

WER=(0+0+0)/19

WER=0

Figure3.16:texte traduir en français

3.6. Conclusion

Dans ce chapitre, nous avons présenté l'environnement de travail, à la fois du point de vue matériel et logiciel, nécessaire à la réalisation de notre système. Nous avons utilisé l'API de reconnaissance vocale de Google (Google Speech Recognition) pour développer un système de reconnaissance vocale à partir d'un microphone ainsi que pour la reconnaissance de fichiers audio et vidéo. Ce système réalisé pour Windows peut être développé pour Androïde. Nous avons également utilisé l'API Google Translate (Googletrans) pour ajouter une fonctionnalité de traduction à notre système. Une interface graphique était faite cela nous facilite la manipulation.

Dans chaque cas où nous changeons la source du signal acoustique nous avons calculé l'erreur qui représente le rapport entre le nombre des mots corrects additionné au nombre des mots erronés additionné au nombre des mots supprimés et le nombre total des mots du texte. Les taux calculés sont très faibles ou nuls cela prouve la fiabilité du système proposé.

Conclusion générale

La recherche sur la reconnaissance vocale est influencée par les avancées technologiques. Cela a commencé avec les systèmes analogiques, et les progrès rapides de l'informatique et de la microélectronique ont ouvert de nouveaux horizons pour le domaine en termes de technologie et d'applications.

Au cours des dernières décennies, la recherche en reconnaissance automatique de la parole a été activement menée dans le monde entier, encouragée par les progrès du traitement du signal, des algorithmes, des architectures et du matériel. Les systèmes ASR ont été développés pour une large gamme d'applications, allant de la reconnaissance de mots-clés de petit vocabulaire, aux systèmes interactifs vocaux de commande et de contrôle de vocabulaire de taille moyenne, à la dictée vocale à grand vocabulaire, à la compréhension spontanée de la parole et à la traduction vocale avec un domaine limité

La reconnaissance vocale couvre tous les aspects liés à l'interprétation automatique du langage humain. Les applications de cette technologie sont nombreuses : navigation sur les serveurs vocaux dans les téléphones, apprentissage des langues étrangères, commandes vocales dans les voitures, les téléphones ou encore dans les blocs opératoires, dictée vocale, sondages de reconnaissance vocale dans les zones sécurisées ou les environnements légaux, etc.

Le travail réalisé dans le cadre de ce mémoire s'agit d'une application Windows que nous avons nommé « SAY IT ». « SAY IT » à pour objective de convertir une parole en texte écrit en arabe. Pour cela, nous avons utilisé le modèle de reconnaissance vocale automatique de Google Speech API qui repose sur des techniques d'apprentissage profond.

Les tests sont effectués sur la parole issue d'un microphone »temps réels « et aussi sur des fichiers audio et vidéo déjà enregistrés.

Pour enrichir notre travail nous avons passé à la traduction du texte abouti en plusieurs langues, enfin et Pour connaitre l'efficacité de notre système nous avons calculé les taux d'erreur. Les résultats sont plus que satisfaisants ce qui justifie l'utilisation de l'API Google Speech Recognition dans de nombreuses applications et services.

Comme perspectives nous proposons le développement de l'application "SAY IT" pour Androïde.

Références

Abdelhamid, A. A., Alsayadi, H. A., Hegazy, I., & Fayed, Z. T. (2020). *End-to-end arabic speech recognition: A review.* Paper presented at the Proceedings of the 19th Conference of Language Engineering (ESOLEC'19), Alexandria, Egypt.

Abushariah, M. A. J. I. J. o. S. T. (2017). TAMEEM V1. 0: speakers and text independent Arabic automatic continuous speech recognizer. *20*, 261-280.

Al-Anzi, F. S., & AbuZeina, D. J. A. S. E. J. (2022). Synopsis on Arabic speech recognition. *13*(2), 101534.

Bahdanau, D., Cho, K., & Bengio, Y. J. a. p. a. (2014). Neural machine translation by jointly learning to align and translate.

Baker, J. J. S. R. (1975). Stochastic modeling for automatic speech recognition. 521-542.

Belguith, L. (1999). *Traitement des erreurs daccord de larabe basr une analyse syntagmatique ndue pour la vfication et une analyse multicrit pour la correction.* Thèse de doctorat, Facults Sciences de Tunis,

Bogert, B. P. (1963). *The quefrency alanysis of time series for echoes: Cepstrum, pseudoautocovariance, cross-cepstrum and saphe cracking.* Paper presented at the Proc. Symposium Time Series Analysis, 1963.

Cerisara, C. (1999). *Contribution de l'approche multi-bandes à la reconnaissance automatique de la parole.* Vandoeuvre-les-Nancy, INPL,

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. J. a. p. a. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation.

Cooley, J. W., & Tukey, J. W. J. M. o. c. (1965). An algorithm for the machine calculation of complex Fourier series. *19*(90), 297-301.

Dammak, A. M. (2016). *Approche hybride pour la reconnaissance automatique de la parole en langue arabe.* Le Mans Université; Université de Sfax (Tunisie),

Davis, K. H., Biddulph, R., & Balashek, S. J. T. J. o. t. A. S. o. A. (1952). Automatic recognition of spoken digits. *24*(6), 637-642.

Davis, S., Mermelstein, P. J. I. t. o. a., speech,, & processing, s. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *28*(4), 357-366.

Douib, O. (2018). *Reconnaissance automatique de la parole arabe par cmu sphinx 4.*

Doukas, N., & Bardis, N. G. (2017). *Current trends in small vocabulary speech recognition for equipment control.* Paper presented at the AIP Conference Proceedings.

Doukas, N., Bardis, N. G., & Markovskyi, O. P. (2017). *Task and context aware isolated word recognition.* Paper presented at the 2017 International Conference on Control, Artificial Intelligence, Robotics & Optimization (ICCAIRO).

Fernández, S., Graves, A., & Schmidhuber, J. (2007). *An application of recurrent neural networks to discriminative keyword spotting.* Paper presented at the Artificial Neural Networks–ICANN 2007: 17th International Conference, Porto, Portugal, September 9-13, 2007, Proceedings, Part II 17.

Fitch, F. B. J. T. J. o. S. L. (1944). Warren S. McCulloch and Walter Pitts. A logical calculus of the ideas immanent in nervous activity. Bulletin of mathematical biophysics, vol. 5 (1943), pp. 115–133. *9*(2), 49-50.

Font, M. F. J. B., California. (2005). Multi-microphone signal processing for automatic speech recognition in meeting rooms.

Gruhn, R. E., Minker, W., & Nakamura, S. (2011). *Statistical pronunciation modeling for non-native speech processing*: Springer Science & Business Media.
Haton, J.-P., Cerisara, C., Fohr, D., Laprie, Y., & Smaïli, K. (2006). *Reconnaissance automatique de la parole: Du Signal à son Interprétation*: Dunod.
Hermansky, H., Morgan, N. J. I. t. o. s., & processing, a. (1994). RASTA processing of speech. *2*(4), 578-589.
Hochreiter, S., & Schmidhuber, J. J. N. c. (1997). Long short-term memory. *9*(8), 1735-1780.
Ioffe, S. (2006). *Probabilistic linear discriminant analysis.* Paper presented at the Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part IV 9.
Khelil, A., Berrah, S., & Amiar, L. (2022). Reconnaissance des chiffres arabes parlés par les réseaux de neurones convolutionnels.
Manohar, V., Povey, D., & Khudanpur, S. (2017). *JHU Kaldi system for Arabic MGB-3 ASR challenge using diarization, audio-transcript alignment and transfer learning.* Paper presented at the 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU).
MARIANI, J. J. T. T. d. s. (1990). Reconnaissance automatique de la parole: progrès et tendances. *7*(4), 239-266.
Markel, J. D., Gray, A. H., & Wakita, H. (1973). *Linear prediction of speech: theory and practice*: Speech communications research laboratory.
Menacer, M. A. (2020). *Reconnaissance et traduction automatique de la parole de vidéos arabes et dialectales.* Université de Lorraine,
O'Shaughnessy, D. J. I. p. (1988). Linear predictive coding. *7*(1), 29-32.
Plátek, O. J. C. U. i. P., Prague. (2014). Automatic speech recognition using Kaldi.
Rabiner, L., & Juang, B.-H. (1993). *Fundamentals of speech recognition*: Prentice-Hall, Inc.
Romdhani, S. (2015). *Implementation of dnn-hmm acoustic models for phoneme recognition.* University of Waterloo,
Singh, A. P., Nath, R., & Kumar, S. (2018). *A survey: Speech recognition approaches and techniques.* Paper presented at the 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON).
Souissi, E. J. T. s. d. d., Universit~ de Paris 1/11, Octobre. (1997). Etiquetage grammatical de l'arabe voyellé ou non.