Republique Algerienne Democratique et Populaire

Ministere de l'enseignement Superieur et de la Recherche Scientifique

Universite Blida -1-

Faculte des sciences

Departement d'Informatique

Domaine Mathematiques et Informatique

Specialite : Traitement Automatique du Langage - TAL

Memoire de Master

Thème

Traduction automatique de la parole de l'anglais vers l'arabe

Présenté par :

Echikr Abdennour — **Encadreur** : Mr.Abbas Mourad

Abdelli Haithem — **Promotrice** : Mme.Tebbi Hanane

Soutenu le : 3/10/2021

Devant le jury Composé de

M. Bala Mahfoud

Mme. Hireche Célia

Année Universitaire : 2020/2021

Remerciements

Nous remercions Dieu le tout puissant de nous avoir données la santé, la volonté, le courage d'achever ce projet de fin d'étude.

Ces quelques lignes ne suffirent pas à exprimer toute la gratitude et tous les moments que nous avons partagés ensemble, mais parfois un simple merci vaut bien plus que tous les discours du monde, ce mémoire nous a permises de rencontrer également des personnes merveilleuses et d'une rare générosité.

Nous remercions notre promotrice Mme Tebbi pour son acceptation de diriger notre travail et ses conseils et remarques constructives.

que nous remercions énergiquement pour son soutien et sa générosité de nous aider à finaliser notre mémoire jusque-là, Pour bien vouloir diriger ce travail et pour ses remarques constructives. Nous la remercions particulièrement pour sa disponibilité, ses encouragements.

Nous remercions Mr ABBAS, notre encadreur au sein de Centre de Recherche Scientifique et Technique pour le Développement de la Langue Arabe (CRSTDLA), pour nous avoir accordées sa confiance pour la réalisation de ce projet à distance.

Nous remercions Mr LICHOURI, pour ses précieuses directives qu'il nous a prodiguées avec tout intérêt, ainsi que son appuie considérable dans notre démarche, et surtout son disponibilité a tout moment.

Résumé

Avec la mondialisation et les déplacements fréquents entre les pays, la différence de langue est une barrière qui inquiète touristes et hommes d'affaires. Mais grâce à la traduction automatique de la parole, cette barrière devrait être brisée pour stimuler le commerce international et le tourisme et égaliser les compétences en communication.

En 2021, la langue arabe compte plus de 422 millions de locuteurs, et elle se situe à la cinquième place dans le monde, et c'est ce qui nous a amené à concentrer notre projet sur la traduction automatique de la parole entre l'anglais et l'arabe, en créant une application web.

Nous avons pu réaliser ce projet en utilisant le modèle CMUsphinx pour implementer un système de reconnaissance de la parole qui a été apprentie avec le corpus fluent. La traduction automatique faite avec un système RBMT, et pour la synthèse on a utilisé pyttsx3, l'ensemble a crée translator

Mots-clés : traduction automatique de la parole, cmushpinx, RBMT Pyttsx3, synthèse vocale.

Abstract

Having summed up with globalization and the frequent displacements between countries, the difference of language is a barrier which worries tourists and businessmen. But grce in the automatic translation of word, this barrier should be broken to stimulate international trade and tourism and level competences in communication.

In 2021, the Arab language relies more than 422 million speakers, and it is on the fifth place in the world, and it is what led us to concentrate in our plan on the automatic translation of word between English and Arabic, by creating an application web.

We could accomplish this plan by using the model CMUsphinx for implementer a system of recognition of the word which was learner with the fluent corpus. The automatic translation made with a system RBMT, and for synthesis they used pyttsx3, group has creates translator

Key words: automatic translation of word, cmushpinx, RBMT Pyttsx3, vocal synthesis.

ملخص

مع العولمة والسفر المتكرر بين الدول ، يعد الاختلاف اللغوي حاجزا يؤرق السياح و رجال الاعمال. ولكن من خلال الترجمة التلقائية للكلام ، من المتوقع كسر هذا الحاجز لتعزيز الأعمال التجارية الدولية والسياحة وتحقيق المساواة في مهارات الاتصال.

في عام 2021 ، تضم اللغة العربية أكثر من 422 مليون ناطق وهي في المركز الخامس على مستوى العالم وهذا ماجعلنا نركز في مشروعنا على على الترجمة التلقائية للكلام بين الانجليزية و العربية و ذالك من خلال انشاء تطبيق يساعد على ذلك .

تمكنا من إنجاز هذا المشروع بالاستعانة ب cmushpinx لتدريب قاعدة البيانت الصوتية وتحصلنا من خلالها على نموذج للتعرف على اللغة الانجلزية حسب قاعدة البينات الصوتية المستعملة, قمنا بالترجمة إلى العربية بإستعمال نظام RBMT الذي يعتمد على الترجمة المباشرة من خلال قاعدة البيانات لكل من اللغتين وبعدها نستعين Pytssx3 للتوليف الصوتي للغة العربية, و هذا ما احتواه تطبيقنا TRANSLATOR.

الكلمات المفتاحية : الترجمة التلقائية للكلام ,Pyttsx3 RBMT ,cmushpinx ,التوليف الصوتي.

Table des matières

Chapitre I : Généralité sur le TAL

Chapitre II :Traduction automatique de la langue

Chapitre III:Conception et Modélisation

Chapitre IV: Réalisation et expérimentation

Liste des figures

Liste de tableaux

Liste des abréviations

- TALN : traitement automatique des langues naturelles
- IA : Intelligence Artificielle
- HMM : Modèle de Markov Cachés
- ANN : Artificial Neural Network
- RAP : Reconnaissance automatique de la parole
- GMM : gaussain mixture models
- EM : expeation and maximisation
- WER : word error rate
- SAP : synthese automatique de la parol
- TSS : text to speech
- MT : Machine Translation
- TL : langue cible
- SL : langue source
- RBMT : Rule-Based Machine Translation
- SMT : statistical machine translation
- NLP : Natural Language Processing
- EBMT : Example Based Machine Translation
- CBMT : Context Based Machine Translation
- BLEU : BiLingual Evaluation Understudy
- BP **:** la Peine de Brièveté
- PER : Position-independent word Error Rate
- TER : Translation Error Rate
- UML : Unified Modeling Language
- UP : unified process
- IHM : Interface Homme Machine
- SER : Sentence Error Rate
- MFCC : Mel-Frequency Cepstral Coefficients
- TAP : Traduction automatique de la parole

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Avec l'immense développement de la technologie et plus précisément le Traitement Automatique du Langage Naturel (TALN) qui a connu une vrai révolution ces dernières années au niveau des recherches avec le soutien de différent gouvernement dans le monde entier.

TALN faite interagir beaucoup de systèmes et d'application de notre vie de tous les jours dans le but est de facilité notre quotidien et parmi les plus grands succès du TALN c'est la traduction automatique de la parole.

Apres 65 ans de recherche l'idée du brésilien qui fait ses courses dans un supermarché en chine le plus normalement possible sachant qu'il ne connait aucun mot en chinois et devenue une réalité.

Malheureusement, malgré l'incroyable évolution des ordinateurs et des connaissances, la traduction automatique de la parole est loin d'être parfaite à cause de la variété linguistique et ces paramètres sémantiques et lexicals, ce qui fait un manque énorme de corpus, notamment de la langue Arabe.

Notre travail s'inscrit dans le cadre général de la Traduction Automatique de la Parole TAP offline. Il consiste à réaliser une application web basé sur trois systèmes qui sont : Le système de Reconnaissance Automatique de la Parole RAP, le système du Synthèse Automatique de la Parole SAP, et le système du la Traduction Automatique TAP,

Notre plateforme est basé sur le CMUSphinx qui est basé sur les Modèles de Markov Cachés (MMC), notre système se base sur un ensemble des règles qui utilisent une base de données sonore contient 10000 enregistrements réaliser par 101 locuteurs.

Notre mémoire comporte quatre chapitres qui sont organisé comme suit:

Le premier chapitre présente une généralité sur le TAL et ces technologies tel que la reconnaissance et la synthèse vocale.

- Le deuxième chapitre détaille la traduction automatique, ces différentes approches et ces mesures d'évaluation
- Le troisième chapitre présente la conception et la modélisation de notre application
- Le quatrième et dernier chapitre définit l'implémentation et l'évaluation de notre outil
- Et enfin nous terminons par une conclusion générale et quelques perspectives pour les futures étudiantes

Chapitre I :

Généralité sur le TAL

1. Introduction

Le traitement automatique du langage naturel, abrégé en TALN, est un domaine multidisciplinaire impliquant la linguistique, l'informatique et l'intelligence artificielle, qui vise à créer des outils de traitement de la langue naturelle pour différentes applications. Il ne doit pas être confondu avec la linguistique informatique, qui vise à comprendre les langues au moyen d'outils informatiques. C'est la discipline qui s'interroge au domaine de l'informatique et du langage. Les premiers travaux dans ce domaine commencent dans les années 1950 principalement aux États-Unis et puis au fur et à mesure ce domaine est devenue de plus en plus intéressant et ça se qui la fait grandir et devenue très vaste et couvre de très nombreuses disciplines de recherche qui peuvent mettre en œuvre des compétences aussi diverses que les mathématiques appliquées.

La TALN comporte plusieurs disciplines classées en fonction de leurs natures, par exemple en sémantique, on trouve : la traduction , le résumer, la reformulation des textes ,la correction orthographique, et même la génération automatique des textes, en syntaxe: La Lemmatisation et analyse syntaxique, en traitement du signal: la reconnaissance automatique de la parole, la synthèse vocale, en extraction d'informations: la recherche d'informations, l'analyse des sentiments. C'est grâce au traitement automatique du langage humain, on obtient une cohérence dans les textes en se basant sur le sens des phrases et formules. Cette cohérence linguistique des textes représente une nécessité absolue pour les ordinateurs afin de pouvoir résumer des textes longs, ou extraire des informations précises. Ces avancées présentent une révolution dans le domaine de traduction ,de l'exécution des ordres vocaux par les ordinateurs et robots, ce qui va faciliter a titre d'exemple la vie des personnes aveugles.

2. Definitions

Le traitement automatique du langage naturel (TALN), ou traitement automatique de la langue naturelle, ou encore traitement automatique des langues (TAL) Le traitement automatique du langage naturel est une discipline de l'ingénierie informatique permettant d'analyser et d'interpréter le langage humain, écrit et oral. Le TALN permet donc, par exemple, d'extraire la morphologie d'une phrase, de générer la traduction d'un texte, de créer une synthèse vocale ou de faire de l'extraction d'informations.[1]

On peut le définir aussi comme : une gamme de techniques de calcul motivées par la théorie pour analyser et représenter des textes naturels à un ou plusieurs niveaux d'analyse linguistique dans le but de réaliser un traitement du langage de type humain pour une gamme de tâches ou d'applications.[2]

Le traitement de langue naturelle est une région de recherche dans l'informatique et l'intelligence artificielle (AI) concerné avec le traitement des langues naturelles comme l'anglais ou l'arabe. Ce traitement implique généralement la traduction de la langue naturelle en données (les nombres) auxquels un ordinateur peut utiliser apprenez du monde. Et cette compréhension du monde est quelquefois utilisée à produire le texte de langue naturelle qui reflète cette compréhension.[3]

3. Signal de la parole (le signal vocal)

Le signal de la parole est présenté sous forme d'onde périodique créée au niveau du larynx, débit d'air à travers les cordes vocales en fonction du temps.

Afin de pouvoir reconnaître le contenu d'un signal de parole correctement, il est nécessaire d'en extraire des paramètres caractéristiques et pertinents pour la reconnaissance. Pour ce faire, plusieurs techniques d'analyse du signal et d'extraction de paramètres peuvent être utilisées.

3.1. Paramètres acoustiques d'un signal de parole

L'étude acoustique du signal de parole correspond à l'évaluation de ses paramètres acoustiques à savoir : la durée, la fréquence fondamentale et l'intensité voir Figure 1 ,1 .Les modifications apportées à l'un d'eux peuvent altérer indéniablement les autres paramètres. Cependant, si nous voulons étudier ces paramètres d'un point de vue acoustique, nous pouvons les considérer comme étant parfaitement indépendants.[4]

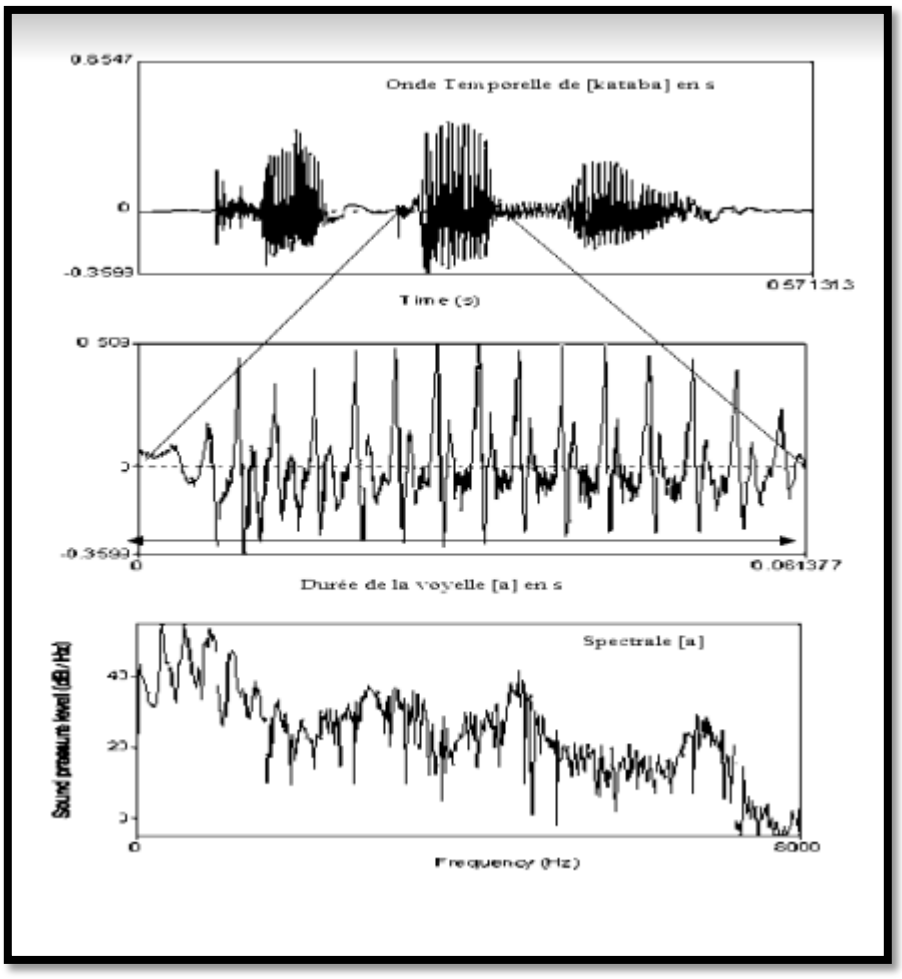

Figure 1. 1 paramètres du signale de la parole

3.1.1 Fréquence fondamentale

Résultant de commandes conscientes ou non, lors de la production de certains phonèmes (les sons voisés de la parole : les voyelles et certaines consonnes), la fréquence fondamentale (Fo) est la conséquence directe des variations de la pression subi-glottique (la tension des cordes vocales). Son corrélat acoustique appelée pitch est généralement linéaire aux basses fréquences d'où la supposition d'une relation linéaire entre lui et la fréquence fondamentale. Cette dernière correspond à la plus basse composante spectrale du spectre et s'inscrit pour la voix parlée dans un registre de 70 à 250 Hz chez l'homme et de 150 à 400 Hz chez la femme et de 200 à 600 chez les enfants.

Les algorithmes détectant le voisement et la fréquence fondamentale du signal de parole sont très nombreux. Les plus utilisés sont l'autocorrélation, AMDF (Aérage Magnitude Différence Function) et le CEPSTRE. L'évaluation de la mélodie relève des corrélats acoustiques des variations de Fo. Quant à sa correspondante perceptive qui est la hauteur, elle est calculée à partir d'une équation logarithmique de la variation de valeurs de la fréquence fondamentale. Sa formule est donnée comme suit :

$$H = 6 \times \text{ ech } \times \frac{\ln(F/F)}{\ln 2}$$

H : la valeur de la hauteur

F1, F2 : les fréquences fondamentales

Ech : coefficient.

Si ech = 1, 2, 4, ... → H sera donnée respectivement en : 1, 1⁄2, 1⁄4, ... tons.[4]

3.1.2 Intensité

L'intensité correspond au corrélat acoustique de la pression sub-glottique. Autrement dit, elle représente l'amplitude des vibrations des cordes vocales. L'énergie contenue dans une portion de signal échantillonné est définie par :

$$E = \sum_{t=1}^{T} S_l^2$$

Dans l'échelle perceptive, l'énergie est exprimée en décibels (dB):

$$\underset{db}{E} = 10 \times \log_{10}\left(\sum_{t=1}^{T} S_i^2\right)$$

L'intensité moyenne apporte une mesure globale de la force sonore de la voix (faible, forte,...). Cette dernière correspond au degré de noirceur des formants du sonagramme.[4]

3.1.3 Durée

La durée est le paramètre acoustique le plus délicat à évaluer. La difficulté de mesure réside dans sa grande variabilité due au contrôle quasi impossible du système phonatoire. Chaque phonème se caractérise par ses propres durées intrinsèques et co-intrinsèques de même que le facteur de compressibilité/expansion. En effet, si nous prenons pour exemple le phonème [a] produit une première fois individuellement puis dans le mot [kataba], nous obtenons des valeurs de durée (pour ce phonème) très différentes.[4]

4. La Reconnaissance Automatique de la parole (RAP)

4.1. Introduction

Simplement défini, le champ de traitement de langue naturelle est concerné avec les théories et les techniques qui adressent le problème de communication de langue naturelle avec les ordinateurs. Un des buts de cette recherche est de concevoir des programmes informatiques qui permettront aux gens de communiquer avec les ordinateurs dans les dialogues naturels de la conversation.

4.2. Definition

La Reconnaissance Vocale ou Reconnaissance Automatique de la Parole (RAP) est une technique informatique qui permet d'analyser un mot ou une phrase captée au moyen d'un microphone pour la transcrire sous la forme d'un texte exploitable par une machine. La reconnaissance vocale, ainsi que la synthèse vocale, l'identification du locuteur ou la vérification du locuteur, font partie des techniques de traitement de la parole. Ces techniques permettent notamment de réaliser des interfaces vocales c'est-à-dire ddes interfaces homme machine (IHM)

où une partie de l'interaction se fait en utilisant la voix. Parmi les nombreuses applications, on peut citer les applications de dictée vocale sur PC où la difficulté tient à la taille du vocabulaire et à la longueur des phrases, mais aussi les applications téléphoniques de type serveur vocal, où la difficulté tient plutôt à la nécessité de reconnaître n'importe quelle voix dans des conditions acoustiques variables et souvent bruyantes.

4.3. Composants d'un système de RAP :

Un système de reconnaissance automatique de la parole comporte typiquement 4 modules[5]:

le prétraitement acoustique : qui va identifier les zones de parole dans l'enregistrement à transcrire et en extraire des séquences de paramètres acoustiques..

le modèle de prononciation : qui associe les mots connus par le système à leurs représentations phonétiques.

le modèle acoustique-linguistique : servant à prédire les phonèmes les plus probablement prononcés dans un énoncé audio, ainsi que la séquence des mots la plus probable dans cet énoncé.

le décodeur : qui va combiner les prédictions des modèles acoustiques et linguistiques pour proposer la transcription en texte la plus probable pour un énoncé de parole donné.

4.4. Fonctionnement de RAP

L'apprentissage automatique réalise une association entre les segments élémentaires de la parole et les éléments lexicaux.

Le principe de fonctionnement d'un système RAP est simple figure 1.2. Une personne en parlant émet des variations de pression dans son larynx, les sons produits sont numérisés par le micro du logiciel afin d'être transmis sur le réseau. Ils sont ensuite transformés en vecteurs acoustiques. Le moteur de reconnaissance va alors analyser cette suite de vecteurs acoustiques en la comparant avec ceux qu'il a en mémoire (son modèle de langage) et proposer la suite qui lui paraît la plus probable. Il est donc nécessaire que la suite de vecteurs acoustiques se rapproche d'une de celles

qui est mémorisée par le moteur de reconnaissance. Afin de créer cette base, il est primordial de développer ce que l'on appelle une grammaire.

Cette association fait appel à une modélisation statistique entre autres par modèles de Markov cachés (HMM, Harden Markov Model) et/ou par réseaux de neurones artificiels (ANN, Artificiel Neural Networks)[6].

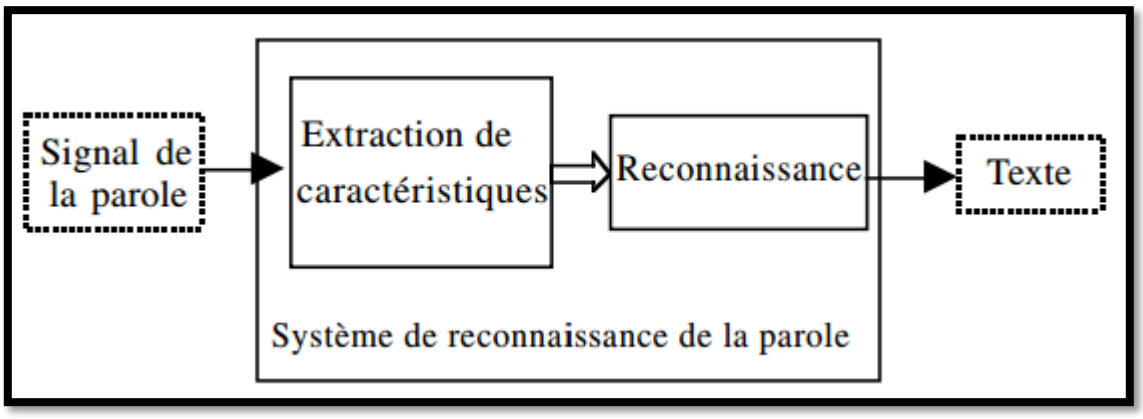

Figure 1. 2 Blocs de base composant le système RAP[6]

Un signal vocal est la parole du locuteur dirigée vers le système de reconnaissance vocale pour traitement et conversion en texte au moyen de l'intelligence artificiel

4.5. Les modèles acoustique et quelques approches de la RAP

Les modèles acoustiques sont des modèles stochastiques qui sont utilisés conjointement à un modèle de langage afin de prendre des décisions quant-à la suite de mots contenue dans la phrase.

Le rôle du modèle acoustique est de calculer la probabilité qu'un événement linguistique (phonème, mot, ...) ait généré une séquence de vecteurs de paramètres extraits d'un signal de parole.

Quelques caractéristiques importantes des modèles acoustiques doivent être prises en compte. D'un point de vue utilisabilité, les modèles acoustiques doivent être robustes puisque les conditions acoustiques de la tâche de reconnaissance sont souvent différentes des conditions d'entraînement. En effet, le signal de parole possède de nombreuses variabilités qui ont pour conséquence d'augmenter la disparité entre la réalisation acoustique et le contenu linguistique. D'un point de vue pratique, les modèles acoustiques doivent être efficaces. Pour que leur

utilisation soit acceptable, il est nécessaire qu'ils respectent certaines contraintes temporelles et donc proposer des temps de réponse relativement courts.

Les paramètres d'un modèle acoustique (F0,durée,intensité) sont estimés à partir d'un corpus d'entraînement. Ce corpus d'entraînement est généralement transcrit manuellement. Cela permet d'identifier les segments de parole correspondant à chaque événement linguistique.

Actuellement, on distingue deux types de modèles acoustiques couramment utilisés : les modèles de Markov Cachés (Hidden Markov Model - HMM) utilisant des mixtures de gaussiennes (Gaussian Mixture Models - GMM), et les modèles hybrides HMM utilisant des réseaux de neurones (Artificial Neural Network - ANN). D'autres techniques tel que les machines à support vectoriel, ont récemment fait leur apparition.

4.5.1 Le model de Markov Cachés (Hidden Markov Model - HMM)

Un HMM est un automate probabiliste contrôlé par deux processus stochastiques. Le premier processus, interne au HMM et donc caché à l'observateur, débute sur l'état initial puis se déplace d'état en état en respectant la topologie du HMM. Le second processus stochastique génère les unités linguistiques correspondant à chaque état parcouru par le premier processus.

4.5.1.1 Structure de Hidden Markov Model

Un HMM est défini par :

N : le nombre d'états composant le modèle.

A : la matrice des probabilités aij de transition entre les états, de taille NxN.La somme des probabilités de transitions entre un état i et tous les autres états doit être égale à 1, i.e. $\forall i, \sum_{j=1}^{N} a_{ij} = 1$.

π_i : la probabilité d'être dans l'état i à l'instant initial. La somme de ces probabilités doit également être égale à 1, i.e. $\sum_{i=1}^{N} \pi_i = 1$.

bi : la densité de probabilité de l'état i.

la Figure 1. 3 est un exemple des états de HMM

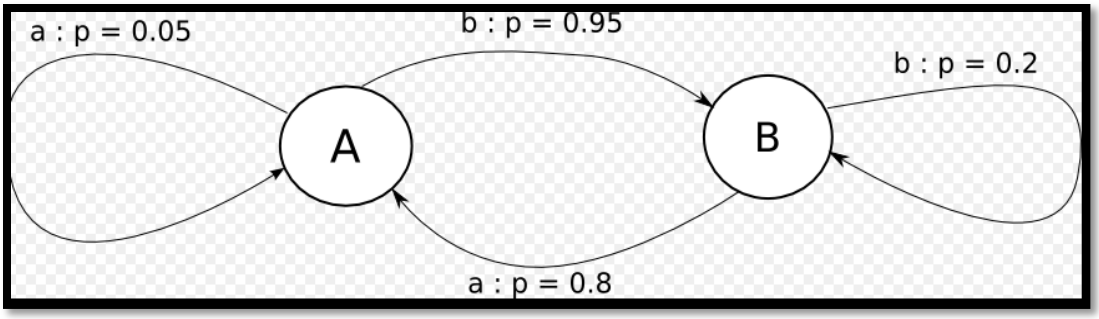

Figure 1. 3 Les états de HMM[7]

4.5.1.2 Apprentissage de Hidden Markov Model HMM

Afin d'utiliser un HMM pour la reconnaissance de la parole, il est nécessaire d'entraîner un modèle, ce qui signifie déterminer les paramètres optimaux des modèles de phonèmes constituant le HMM. Chaque état correspond à un (Gaussain Mixture Models) GMM dont on doit estimer les paramètres (moyenne et matrice de covariance). Ceci est fait grâce à un corpus d'apprentissage dont on connaît la suite de mots prononcée pour chaque phrase.

L'apprentissage du HMM consiste à déterminer les paramètres Θ! = {N, A, {πi}, {bi}} optimaux selon un critère de maximum de vraisemblance (Maximum Likelihood - ML).

Le critère de ML correspond à trouver le Θ! qui maximise la fonction de vraisemblance comme suit :

$$\widehat{\Theta}_{ML}(Y) = \underset{\Theta}{\operatorname{argmax}} f(Y \mid \Theta)$$

Y étant l'ensemble des données d'apprentissage.

La complexité du problème d'optimisation est grande en raison des données incomplètes Y à notre disposition. Les données du corpus d'entraînement sont appelées données incomplètes car elles ne contiennent pas l'information concernant le GMM qui les a générées. Il est donc nécessaire d'utiliser l'approche itérative « Expectation and Maximisation » (EM), présentée dans Rabiner (1989) permettant de converger vers le modèle optimal. Généralement, le processus est interrompu lorsque le gain en vraisemblance est inférieur à un seuil prédéfini, ou lorsque le nombre d'itérations désiré a été atteint.

Comme son nom l'indique l'approche EM comporte deux phases :

E ou expectation : dans cette étape, il faut compléter les données incomplètes Y en leur attribuant des données manquantes en fonction du modèle courant Θi.

M ou maximisation : trouver le nouvel ensemble de paramètres Θ^{i+1} qui maximise la vraisemblance des données complètes Zi connaissant le modèle Θi :

$$\Theta^{i+1} = \underset{\Theta}{\operatorname{argmax}} P(Z_i \mid \Theta)$$

Cet algorithme nécessite un modèle initial Θ0, à partir duquel on effectue plusieurs itérations des étapes E et M. À partir de ce modèle initial, EM converge vers un maximum local, il en découle que le choix des paramètres du modèle initial va influencer grandement la convergence de l'algorithme.

Dans la pratique, on utilise un système de reconnaissance existant pour effectuer l'alignement forcé par rapport aux états (avec l'algorithme de Viterbi par exemple). Á partir de cette segmentation forcée, on applique l'algorithme EM au niveau de chaque état pour déterminer les paramètres du GMM correspondant.

Plusieurs solutions sont possibles pour initialiser chaque GMM. La première consiste à estimer une première gaussienne (moyenne et variance). Ensuite, deux gaussiennes sont créées à partir de la première en faisant varier la moyenne de $\pm_{\epsilon}$" (déterminé en fonction de la variance de chaque paramètres). Puis on réestime les paramètres de chaque gaussienne avec les données qui ont le plus de vraisemblance avec elle. On réitère plusieurs fois pour atteindre le nombre de gaussiennes désiré. Une autre solution est d'utiliser l'algorithme des k-means (ou k-moyennes). Cet algorithme a pour but de partager l'espace en k parties en essayant de trouver les centres naturels de ces parties. L'objectif est double : minimiser la variance intra-classe ou l'erreur quadratique1 tout en maximisant la variance inter-classe. Une fois le modèle Θ0 déterminé, s'ensuivent plusieurs itérations d'EM. Les différents algorithmes se différencient par la manière de compléter les données incomplètes (étape E). On cherche à trouver le modèle Θ qui maximise la log-vraisemblance L (Z, Θ) des données complètes étant donné le modèle courant Θi.

$$\Theta^{i+1} = \underset{\curvearrowleft}{\operatorname{argmax}}\, E\left[\log(L(Z,\Theta)) \mid \Theta^{i}\right]$$

Cet algorithme utilise une procédure « avant-arrière » pour affecter les valeurs aux données manquantes et compléter les données incomplètes, d'où son autre nom de Forward-Backward Une autre manière de compléter les données incomplètes consiste à leur affecter leur valeur la plus probable comme dans l'algorithme de Viterbi[8]. Le lecteur pourra se référer à[9]; [10] [11]pour les informations complémentaires à ce sujet.

4.5.1.3 les inconvénients de Hidden Markov Model

Les modèles de Markov cachés reposent sur un ensemble d'hypothèses simplificatrices. Tout d'abord, les données à l'entrée d'un HMM sont supposées être statistiquement indépendantes, et donc la probabilité qu'un vecteur soit émis au temps t ne dépend pas des vecteurs précédemment émis. Cette hypothèse est irréaliste. En effet, les vecteurs de paramètres acoustiques sont calculés sur des portions de signal d'une durée très petite (en général 30 ms.), il est donc incorrect de penser que deux trames successives ne possèdent aucune corrélation statistique. L'utilisation des dérivées premières et secondes des paramètres acoustiques comblé en partie cette imprécision, mais à l'heure actuelle, les systèmes n'intègrent pas complètement la corrélation entre les trames de manière efficace. Les modèles segmentaux comme présentés dans[12] sont des modèles tentant de prendre en compte cette dépendance de manière intrinsèque lors des calculs des probabilités. Cependant, ils n'ont jamais montré de gain en performance probant et irrévocable au point de remplacer les modèles classiques. Une autre limitation réside dans la modélisation de la durée, qui est implicite dans un HMM. Elle est déterminée par le critère visant à maximiser la probabilité a posteriori. L'utilisation de modèles de trajectoire permettant de rendre compte de l'évolution temporelle du signal de parole a été proposé par [13]L'utilisation de HMMs de premier ordre2 repose sur l'hypothèse que la parole est également un processus de Markov de premier ordre. Des modèles d'ordre supérieurs ont été considérés [14]mais le compromis entre coût de calcul et gain en performance n'est pas évident.

4.5.2 Les réseaux de neurones artificiels:

Les modèles neuromimétiques sont constitués de cellules élémentaires, appelées neurones, fortement connectées entre elles [15] Ces neurones émettent en sortie une fonction non linéaire de la somme pondérée de leurs entrées, les plus fréquemment utilisées sont les sigmoïdes ou les fonctions de Heavyside. Une autre forme plus répandues de réseau de neurones c'est le perceptron multicouche (multilayer perceptron MLP) . Un perceptron est un réseau sans contre-réaction, ce qui signifie que les sorties des neurones de la couche i forment les entrées des neurones de la couche i+1.

La figure (1.4) montre un perceptron à trois couches dont une cachée, permettant de reconnaître N symboles (phonèmes ou autres).

L'ANN est alimenté avec des paramètres acoustiques dont il se chargera de trouver la combinaison optimale permettant d'obtenir la meilleure classification.

Les scores en sortie du réseau de neurones sont généralement normalisés par une fonction appelée « softmax » définie par :

$$pp_i = \frac{e^{s_i}}{\sum_{j=1}^{c} e^{s_j}}$$

Les probabilités ppi peuvent être interprétées comme des probabilités a posteriori des classes (ou symboles) i connaissant le signal de parole d'entrée. Ces probabilités sont ensuite normalisées avec la probabilité a priori des symboles pour obtenir des vraisemblances normalisées (scaled likelihoods) pouvant être exploitées comme probabilités d'émission des états d'un HMM classique.

D'un point de vue philosophique, les ANNs sont très différents des GMMs dans le sens où ils n'estiment pas uniquement la probabilité d'un symbole indépendamment des autres. Ils intègrent plutôt une approche discriminante qui vise non seulement à donner une grande probabilité pour le symbole réellement émis, mais aussi à donner une faible probabilité aux autres classes. L'apprentissage des ANNs se fait par un processus itératif dit de « rétro propagation du gradient d'erreur » qui modifie les paramètres des neurones des couches cachées en fonction d'un critère (par exemple les moindres carrés).

Les expériences de la première partie de nos travaux utilisent un système hybride HMM/ANN acceptant deux types de paramètres acoustiques en entrée (F0 ,intensité) .[16]

4.5.2.1 Structure de réseaux de neurones artificiels

Le Figure suivant 1.4 représenter un type de réseaux de neurones multicouches

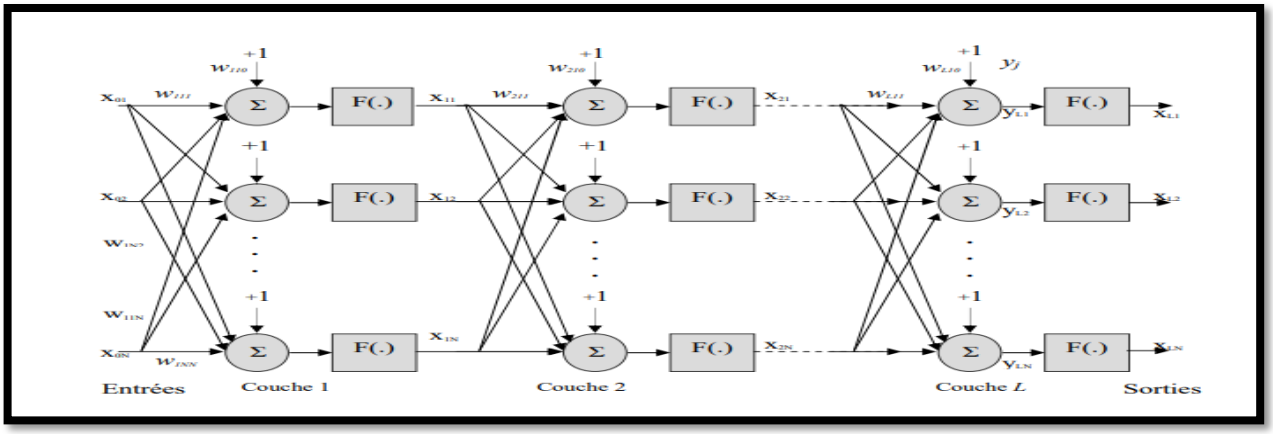

Figure 1. 4 Architecture de réseaux multi couche[17]

Un réseau de neurones peut prendre des formes différentes selon l'objet de la donnée qu'il traite et selon sa complexité et la méthode de traitement de la donnée.

Les architectures ont leurs forces et faiblesses et peuvent être combinées pour optimiser les résultats. Le choix de l'architecture s'avère ainsi crucial et il est déterminé principalement par l'objectif.

Les architectures de réseaux neuronaux peuvent être divisées en 4 grandes familles :

Réseaux de neurones Feed fowarded

Réseaux de neurones récurrents (RNN)

Réseaux de neurones à résonance

Réseaux de neurones auto-organisés

4.5.2.2 Apprentissage de réseaux de neurones artificiels

- **Apprentissage supervisé :**

L'apprentissage supervisé (supervised learning en anglais) est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés, au contraire de l'apprentissage non supervisé. On distingue les problèmes de régression des problèmes de classement1. Ainsi, on considère que les problèmes de prédiction d'une variable quantitative sont des problèmes de régression tandis que les problèmes de prédiction d'une variable qualitative sont des problèmes de classification.

Les exemples annotés constituent une base d'apprentissage, et la fonction de prédiction apprise peut aussi être appelée « hypothèse » ou « modèle ». On suppose cette base d'apprentissage représentative d'une population d'échantillons plus large et le but des méthodes d'apprentissage supervisé est de bien généraliser, c'est-à-dire d'apprendre une fonction qui fasse des prédictions correctes sur des données non présentes dans l'ensemble d'apprentissage.[18]

- **Apprentissage non supervisé** :

La différence majeure entre l'apprentissage supervisé et non supervisé peut être résumée dans le fait que le deuxième type d'apprentissage est autodidacte qui n'a pas besoin d'expert pour le guider à adapter ses paramètres qu'il ne dispose que des valeurs entrées. Remarquons cependant que les modèles d'apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision.

4.5.2.1 les inconvénients de réseaux de neurones artificiels

En utilisant le tronçonnage aide-mémoire d'un étendu chiffre de mots écrits à la main par des individus, on peut améliorer la théorie neuronal de relecture, Chaque arrangement peut postérieurement convenir enfiler dessous la dimension d'une miniature brute ; ayant une topologie spéciale qui doit s'entendre en rapport avec la fermeté du fait modélisé, et le chiffre d'exemples. Mais dans un logiciel ça ne peut pas être parfait à cause des exemples limités.

4.5.3 Les mixtures de gaussiennes :

Un Modèle de mélange gaussien est un mélange de distributions de probabilité qui suivent une loi gaussienne multivariée. Une fonction de densité de probabilité est estimée par la somme finie du gaussien composant Gaussian Mixture Model (GMM).

La figure 1.5 suivante montre un exemple de gaussienne bi-variée (multi-variée de dimension 2). Elle est définie par :

$$\mathcal{N}(\mu, \Sigma, x) = \frac{1}{2\pi \det(\Sigma)^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu)^T \mathbf{\Sigma}^{-1}(x-\mu)}$$

Avec μ est la moyenne et Σ est la matrice de variance-covariance.

Les GMMs sont à la base des systèmes de reconnaissance HMMs les plus couramment utilisés.[19]

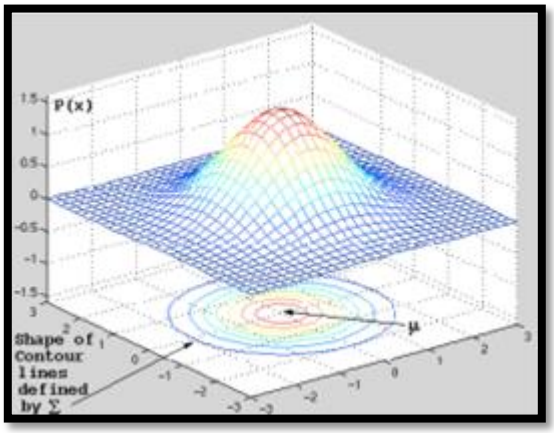

Figure 1. 5 Exemple de densité de probabilité d'une gaussiennes bi-variée

Un problème rencontré lors de la mise en œuvre des modèles de mélange concerne la taille du vecteur de paramètres à estimer. Dans le cas d'un mélange gaussien de g composantes de dimension p le paramètre est de dimension$\left(g \times (1 + p + p^2)\right) - 1$. La quantité de données nécessaire à une estimation fiable peut alors être trop importante par rapport au coût de leur recueil.

Une solution courante consiste à identifier les variables qui fournissent le plus d'informations pour l'analyse parmi toutes les variables disponibles et à éliminer celles qui ne présentent pas d'intérêt. Cette technique est largement utilisée pour distinguer les problèmes et rarement utilisée dans les problèmes de classification.

Une méthode alternative consiste à considérer des modèles dits parcimonieux dans lesquels on contraint le modèle initial de manière à n'estimer qu'un nombre plus restreint de paramètres. Dans le cas gaussien, la paramétrisation synthétique des lois de probabilités grâce à deux ensembles $\boldsymbol{\mu}_k$ et $\boldsymbol{\Sigma}_k$ de paramètres permet des ajouts de contraintes relativement simples. Le plus souvent, ces contraintes ont une signification géométrique en termes de volumes, d'orientation et de forme.

4.6. Mesures d'évaluation

En reconnaissance automatique de la parole, la mesure d'évaluation la plus répandue est le taux d'erreur mot (Word Error Rate - WER). Le WER consiste à comparer la phrase reconnue et la phrase de référence (celle qui a effectivement été prononcée).

Il est défini comme suit :

$$WER = \frac{I + D + S}{W} * 100$$

avec I le nombre d'insertions, D le nombre de suppressions, S le nombre de substitutions et W le nombre de mots dans la référence.

Une autre mesure d'évaluation que nous utiliserons est l'entropie conditionnelle moyenne ou "équivocation" apportée par le système. En considérant le système comme un canal de transmission ayant une source S, produisant des symboles f à son entrée, et un récepteur R, recevant des symboles g en sortie, alors l'équivocation 〖H 〗_R (s) est définie par :

$$H_R(S) = -\sum_{f,g} P(f,g)\log_2 P(g \mid f)$$

Cette mesure permet d'évaluer le degré de confusion apporté par le système de reconnaissance.[20]

4.7. Les logiciels de reconnaissance vocale

Le tableau suivant représente quelques les travaux les plus récents en reconnaissance vocal

Logiciel de reconnaissance vocale	**Plate-forme**
Dragon Professional	Windows OS
Google Now	Android & ios
Siri	iOS
Amazon Lex	Utilisé dans les applications

Tableau 1. 1 Les logiciels de reconnaissance vocale [21]

5. La synthèse Automatique de la parole (SAP)

5.1. Introduction

Au fil des années, la demande de discours générés par ordinateur n'a cessé de croître. Cela est en partie dû à la nature du texte, de la parole et de l'informatique. Bien sûr, la parole est la représentation de base du langage et existe dans toutes les cultures, il doit donc y avoir un moyen de communication entre les ordinateurs et leurs utilisateurs humains.

La synthèse vocale c'est de faire lire les ordinateurs à haute voix. Il s'agit donc d'environ trois choses : le processus de lecture, le processus de parole et les problèmes liés à l'obtention ordinateurs (par opposition aux humains) pour le faire. Ce domaine d'étude est connu à la fois sous le nom de discours synthèse, c'est-à-dire la génération « synthétique » (informatique) de la parole, et ou TTS ; le processus de conversion d'un texte écrit en discours. Il complète d'autres langues technologies telles que la reconnaissance vocale, qui vise à convertir la parole en texte, et la traduction automatique, qui convertit l'écriture ou la parole dans une langue en écriture ou discours dans un autre.[22]

5.2. Définitions

La synthèse vocale est une technique informatique de synthèse sonore qui permet de créer de la parole artificielle à partir de n'importe quel texte. Pour obtenir ce résultat, elle s'appuie à la fois sur des techniques de traitement linguistique, notamment pour transformer le texte orthographique en une version phonétique prononçable sans ambiguïté, et sur des techniques de traitement du signal pour transformer cette version phonétique en son numérisé écoutable sur un haut-parleur. Il s'agit, comme la reconnaissance vocale, d'une technologie permettant de construire des interfaces vocales.[23]

Et aussi, les systèmes de synthèse vocale, ou synthétiseurs vocaux, sont des programmes informatiques qui génèrent automatiquement de la parole, c'est-à-dire des systèmes qui permettent à l'ordinateur de « parler » ou de « parler » à l'utilisateur. Plus formellement, les systèmes de synthèse vocale sont définis comme des systèmes qui : permettent la génération de nouveaux messages [oraux], soit à partir de zéro (c'est-à-dire entièrement par une règle), soit en recombinant des unités pré-stockées plus courtes.[24]

5.3. Mécanisme de base d'un SAP

Un système de la parole ou du moteur est constitué de deux parties: frontal et Back-end. la partie frontal traite de la conversion de texte en symboles phonétiques tandis que le Back-end Il interprète les symboles phonétiques et « lit » les, en les transformant ainsi en voix artificielle voir figure suivant.

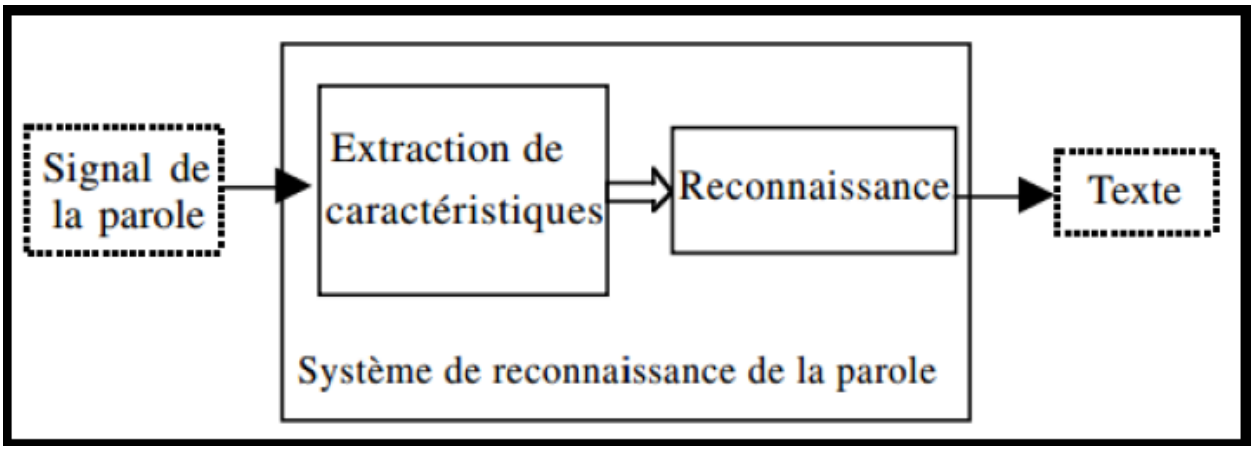

Figure 1. 6 Schéma d'un système vocal[25]

La frontal fournit deux caractéristiques principales: d'abord, une analyse du texte écrit est effectué pour convertir tous les numéros, acronymes et abréviations des mots en pleine (par exemple, le texte « 2 » est converti en « deux ».). Ce prétraitement est définie comme normalisation ou classification le texte (en anglais: tokenization). La seconde fonction consiste à convertir en son correspondant chaque mot symboles phonétiques et effectuer une analyse linguistique du texte révisé, divisant en unités prosodiques, des propositions-à-dire, des phrases et des phrases. Le processus d'attribution orthographe phonétique des mots est appelé texte à la conversion phonème ou graphème à phonème (En anglais text-to-phonème, TTP)[26]

La transcription phonétique et informations prosodie combinés constituent ensemble la représentation symbolique linguistique qui est utilisé par Back-end pour la conversion aux sons de ces informations est le processus de synthèse réel.

5.4. Les techniques de SAP

Les qualités les plus importantes d'une synthèse de la parole sont aspect naturel et un intelligibilité.

La naturalité exprime combien la voix synthétisée est proche de celle de l'homme alors que l'intelligibilité est la facilité de compréhension de la voix synthétisée. Un synthétiseur idéal est en même temps naturel et intelligible, en fait, les systèmes de synthèse de la parole rapprochent ce comportement en essayant d'optimiser les caractéristiques.

Les deux technologies principales utilisées dans la synthèse de la parole sont la synthèse par concaténation et la synthèse par règles. Chaque technologie a ses forces et ses faiblesses: le choix de l'utilisation de l'une ou de l'autre dépend du type d'utilisation finale typique de la synthèse vocale.[27]

5.4.1 La synthèse par concaténation

Comme son nom l'indique, il est basé sur enchaînement c'est à dire la combinaison des fragments de voix enregistrée. En général, cette méthode produit le résultat de la synthèse la plus naturelle, cependant, la différence entre la variation naturelle de la voix humaine et les techniques de fragmentation automatique formes d'ondes petit peut parfois générer du bruit audible.

5.4.1.1 Synthèse par concaténation unités acoustiques

Cette méthode base sure la concaténation des unités acoustique après la segmentation du signale vocale. Au premiers temps les chercheurs font la segmentation en phonèmes mais les expériences ont monté que les phrases de transition entre phonèmes entrainaient des discontinuités sur le signal reconstitué en raison de coarticulation, ce qui fait que les chercheurs ont choisis une autre segmentation en des endroits supposés stationnaires comme le milieu du phonème.

La synthèse diphone utilise une taille minimale des sons base de données contenant tous les diphones (transitions entre les différents sons) typiques d'une langue donnée. Le nombre de diphones dépend des caractéristiques phonétiques de la langue: par exemple, espagnol comprend environ 800 tandis que le diphones allemand a environ 2500. Avec cette technique est stockée dans la base de données un échantillon unique pour chaque diphone. Au cours du traitement en temps réel, aux diphones sélectionnés est superposé prosodie la phrase à synthétiser en utilisant des techniques de DSP (traitement numérique du signal) Telles que le codage prédictif linéaire,[28].[29] La qualité de la voix qui en résulte est généralement inférieure à celle obtenue

pour la synthèse articulatoire, mais il semble plus naturel que celui obtenu avec la synthèse sur la base des règles.

Les défauts de synthèse de diphones sont mineures déconnexion entre les sons, typique du mécanisme de la chaîne, et un effet de la voix métallique, comme dans le résumé fondé sur des règles. Par rapport à ces techniques, la synthèse de diphones ne présente pas d'avantages significatifs, en dehors de la taille réduite de la base de données de soutien. Pour cette raison, l'utilisation de cette technique pour des applications commerciales est en déclin car il continue d'être utilisé dans la recherche grâce à nombreuses implémentations logiciels libres disponibles.

La première application commerciale de la parole en italien, Eloquens, conçu CSELT et commercialisé par Telecom Italie depuis 1993, elle était fondée sur diphones. Il est encore très répandu, disponible en tant que logiciel libre (uniquement pour les systèmes d'exploitation Windows).

5.5. Mesures d'évaluation d'un SAP

Les systèmes de SAP ont été évalués sous différents aspects, tels que l'intelligibilité, la compréhension, le naturel et la préférence de la parole synthétique

5.5.1 Les facteurs influents sur l'intelligibilité et la compréhension

5.5.1.1 Mémoire à court terme

La mémoire à court terme est le plus grand facteur cognitif qui a la plus grande influence sur la tâche de compréhension. En effet, la mémoire à court terme est utilisée pour stocker temporairement des fractions d'informations jusqu'à ce que toutes les informations puissent être complètement comprises. Par conséquent, la technique est tout à fait essentielle lors de la tâche de compréhension. De plus, la charge de la mémoire à court terme doit également être prise en compte. Comme démontré à partir de l'expérience de la tâche simultanée par [30], la mémoire à court terme avait une capacité limitée. Goldstein [31] avait identifié deux niveaux différents de mémoire à court terme, qui étaient le niveau nominal et le niveau supra-nominal. Il a décrit que la mémoire à court terme de niveau nominal était impliquée dans les tâches d'intelligibilité, en se

concentrant sur l'évaluation qualitative. D'autre part, la mémoire à court terme de niveau supra-nominal était utilisée dans des tâches de compréhension, qui nécessitaient que l'information soit identifiée, traitée et comprise. Par conséquent, comme spécifié par des chercheurs précédents, il serait important de prendre en compte la mémoire à court terme dans cette étude.

5.5.1.2 Listeners' préférences

Un autre facteur qui peut influencer sur la performance de tâche est les préférences des auditeurs. [32] les préférences d'auditeurs jugés du feed-back d'auditeurs sur un discours naturel et deux synthétiseurs de discours : MITalk et Votrax. La mesure devait évaluer les mots d'adjectif du feed-back. Les chercheurs ont constaté que les gens ont préféré écouter le discours naturel qu'au deux synthétiseurs de discours et le système MITalk ont été préférés que le système Votrax. Aussi, l'intelligibilité dans le système MITalk a été évaluée pour être plus haute que le système Votrax . En plus [33] a soutenu que les préférences des d'auditeurs ont dépendu beaucoup de la qualité d'intelligibilité de discours. De plus, [34] et [35] déclare que si la qualité d'intelligibilité augmente alors le degré de préférence augmenterait aussi .

5.6. Quelques logiciels de SAP

Le tableau suivant (tableau 1.2) représente quelques les travaux les plus récents de synthèse vocal

Logiciel de synthèse vocale	Meilleur pour
Murf	L'établissement des caractéristiques puissantes pour créer les vidéos de voix-off.
iSpring Suite	Créez des cours eLearning, des classes de travaux dirigés vidéos et des présentations PPT avec les voix-offs.
Notevibes	L'utilisation commerciale, aussi bien que l'usage personnel et l'apprentissage
Natural Reader	L'utilisation personnelle et l'apprentissage, surtout pour les apprentis dyslexiques

Tableau 1. 2 Les logiciels de synthèse vocale[36]

6. Conclusion

Dans ce chapitre nous avons introduirte le domaine de traitement automatique de la langue naturel et on a détaillé deux principales disciplines : la reconnaissance automatique de la parole et la synthèse vocal.

Dans le chapitre suivant on va détailler la troisième approche du TAL qui est la traduction automatique de la parole.

Chapitre II :

Traduction automatique de la langue

1. Introduction :

Machine Traduction (MT) ou la Traduction Automatique de Parole TAP est défini comme : la traduction à partir d'une langue naturelle (langue source SL) vers une autre langue (langue cible TL) à l'aide de systèmes informatisés. La TAP peut être utilisée pour traduire de grands volumes de textes vite, qui est presque impossible de les traités on utilise des méthodes de traduction traditionnelles.[37] [38]

Ce concept est né au 17ème siècle après l'apparence de l'idée de la langue universelle ce qui est devenu incontournable pour faciliter les échanges commerciaux entre les peuples. il était basé sur la Traduction de mot à mot. Après le développement que le monde a connu au 20ème siècle, les chercheurs s'intéressaient beaucoup plus à la traduction automatique et au-dessus de années, elle a fait l'objet d'enquêtes par des linguistes, psychologues, philosophes, informaticiens et ingénieurs, ce qui fait que la TAP a contribué d'une manière très significative au développement de domaines tels que la linguistique informatique, l'intelligence artificielle et traitement du langage naturel orienté application.

Dans ce chapitre on va essayer de fournir une recherche cohérente, quoique nécessairement brève et incomplète sur ce domaine à travers les titres à venir ou on va détaillé les différentes approches développées (linguistiques et computationnelles) et les types de machines traduction.

2. Histoire de la traduction automatique :

Apres la 2eme guerre mondial et le développement de la technologie et surtout le domaine de l'informatique, les chercheur ont l'idée d'utiliser des ordinateurs dans la traduction et le premier qui a proposé cette idée était Warren Weaver dans son Memorandum de 1949, ensuit au fil des années jusqu'au 1960 beaucoup de travail était fait (1952 : Premier colloque de traduction automatique intitulé Conférence sur Traduction automatique, 1954 : Le développement du premier traducteur automatique (très basique) par un groupe de chercheurs de l'Université de Georgetown en collaboration avec IBM, 1954 : Victor Yngve publie la première revue sur la TA). En 1961 quand la linguistique computationnelle est née, grâce aux conférences hebdomadaires organisées par David G. Les foins à la Société de Rand à Los Angeles. Ces conférences seront incluses comme les papiers à la Première Conférence internationale de la Traduction automatique de Langues et d'Analyse de Langue Appliquée de Teddington en septembre de 1961 avec la participation de linguistes et d'informaticiens impliqués dans la traduction comme : Paul Garvin, M de Sydney. L'agneau, Kenneth E.

Harper, Charles Hockett, Martin Kay and Bernard Vauquois. Cette évènement était suivie par la création de comité ALPAC (la Langue Automatique Traitant le Comité consultatif) avec le gouvernement américain aux études les perspectives et les chances de traduction automatique en 1964, et en 1966 ALPAC a causé une chute libre à leur recherche (TA) après la publication de son rapport réputé dans lequel il a conclu que ses travaux sur la traduction automatique se perdent juste du temps et de l'argent. En 1970 les russes ont déclaré le début du projet REVERSO, Puis c'était le tour des japonais avec la création de système ATLAS2 par le FUJITSU en 1978, ce traducteur a été fondé sur les règles aussi il est en mesure de traduire du Coréen au japonais et vice versa. Au cours des années 1980 IBM a commencé le travail dans la traduction automatique statistique et dans les années 1990 la disponibilité de texte parallèle avaient augmenté l'intérêt pour la traduction automatique statistique. En 2006 un instrument de traduction automatique statistique open source appelé Moses était libéré et cela est actuellement le logiciel de traduction automatique statistique le plus complet disponible. En 2007 le lancement d'un système de traduction automatique hybride appelé MÉTIS-II, dans lesquelles les pénétrations de Statistique, l'Exemple la Traduction automatique basée et à base de Règle (STA (MT), EBTA (MT) et RBTA (MT) respectivement) sont utilisés. L'année 2009 connaitre l'apparition de la traduction speech-to-speech qui a été fournis dans le téléphone mobile pour l'anglais, le japonais et le chinois, et en 2013 Kalchbrenner et Blunsom a proposé la nouvelle approche qui a fait une révolution dans la traduction automatique appelé La traduction automatique Neuronale.

3. Definition :

La traduction automatique c'est l'énonciation faite par un système informatisé de ce qui a été énoncé dans une langue source par une langue cible[38].

4. Architectures de TAP :

Au fil des ans, les chercheurs ont adopté différentes approches de traduction automatique de la parole ,reflétait profondeur et la diversité linguistiques de la parole a cette différence eu un impact positif sur le domaine parce quelle l'enrichit tellement. Ces méthodes peuvent être divisées en deux types :

4.1. Approche de traduction automatique basée sur des règles (RBMT) :

Pour la traduction on a utilisé la traduction automatique basée sur des règles (RBMT)qui est un système basé sur des informations linguistiques sur les langues source cible essentiellement extraites de dictionnaires et de grammaires, il travaille sur la morphologie, la syntaxe et sémantique des deux langues. Alors on fait une analyse de syntaxique et sémantique de texte Source et une généré le texte dans la langue cible nous avons besoin de la génération syntaxique et la génération sémantique. Nous avons aussi besoin de la source dictionnaire bilingue et des deux langues

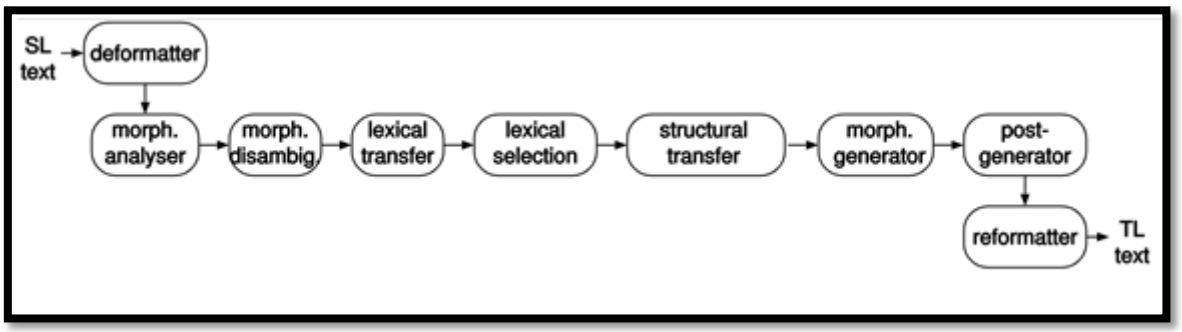

Figure 2. 1 Schéma de la traduction automatique basée sur des règles (RBMT)[40]

SL :langue source

TL :.langue cible

Et dans ce type on trouve trois (03) approches essentiels sont :

- **Approche directe :**

Dans cette approche les mots de la langue source sont traduits sans passer par une représentation intermédiaire supplémentaire, la traduction est fait directement du texte source vers la langue cible. Cette approche implique de prendre une ficelle de mots de la langue source, en enlevant le morphologique l'inflexion des mots pour obtenir les formes basées et le fait de les chercher dans un dictionnaire bilingue entre la source et les langues cible. C'est la traduction qui est faite par le remplacement de mot par mot et ces résultats sont mauvais[37]

- **Approche basée sur le transfert :**

Dans cela, la langue source est transformée dans une représentation abstraite. Une représentation équivalente (avec même niveau d'abstraction) est alors produite pour la langue cible en utilisant des dictionnaires bilingues et des règles de grammaire. Ces systèmes ont trois composantes importantes[41] :

- ✓ **L'analyse :**

L'analyse du texte source est faite basée sur les informations linguistiques comme la morphologie, la partie du discours, la syntaxe, la sémantique, etc. Aussi bien que les algorithmes sont appliqués pour analyser la langue source et en dériver.

- ✓ **Le transfert :**

La représentation syntaxique de langue source est convertie en forme syntaxique de langue cible.

- ✓ **La synthèse :**

Cette approche fortement dépendante de la grammaire et de la structure de la phrase et les modifications apportées à un composant monolingue affectent tous les modules de transfert pour cette langue. Le texte final en langue cible est généré à l'aide d'une analyse morphologique.

- **Approche d'Interlingue** :

Interlingua est une combinaison de deux mots latins Inter et Lingua qui signifie respectivement intermédiaire et langue, cette approche vise à mettre en place une homogénéité linguistique**.** C'est basé a transformée la langue source en une langue intermédiaire (représentation) qui est indépendante de toutes les langues impliquées dans la traduction. Le résultat pour la langue cible est ensuite dérivé par cette représentation auxiliaire. Par conséquent, l'analyse et la synthèse, sont nécessaires dans ce type de système. Ce système est connue par sa pertinente dans la traduction automatique multilingue[37][42].

4.2. Traduction automatique basé sur des corpus :

En raison du haut niveau de précision atteint lors de la traduction, l'approche basée sur le corpus cette méthode a dominé les autres approches et devenue l'un des domaines les plus explorés de la traduction automatique.

La figure 2.2 suivante représente le schéma de la traduction automatique basé sur des corpus :

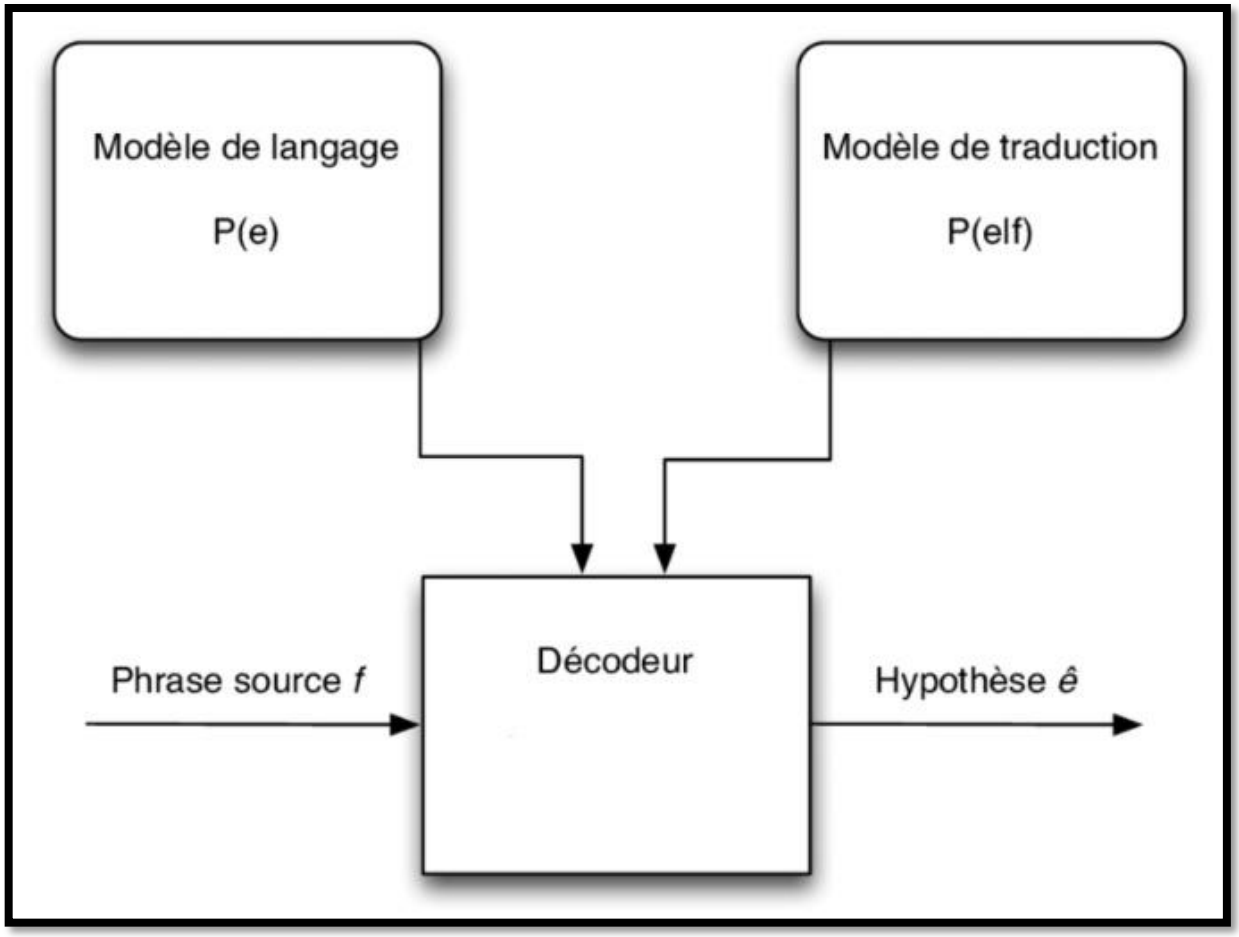

Figure 2. 2 Schéma de la traduction automatique basé sur des corpus[43].

Certaines des approches basées sur le corpus sont expliquées ci-dessous :

- **Traduction automatique statistique (SMT) :** Cette approche est basée sur un modèle statistique. L'avantage du système SMT est qu'aucune connaissance linguistique n'est requise pour les construire. La difficulté dans SMT système est de créer un corpus parallèle massif. Nous avons deux modèles dans SMT, l'un basé sur des mots et l'autre basé sur des phrases. Et dans cette approche on trouve trois (03) modèles :
  - ✓ Modèle de traduction statistique basé sur les mots
  - ✓ Modèle statistique basé sur des phrases
  - ✓ Modèle statistique basé sur la syntaxe

- **Traduction automatique basée sur des exemples (EBMT) :** La traduction basée sur des exemples est basée la comparaison et la recherche d'exemples analogues. le système (EBMT) reçoit un ensemble de phrases dans la langue source (à partir de

laquelle on traduit) et les traductions correspondantes de chaque phrase dans la langue cible avec une cartographie point à point. Ces exemples sont utilisés pour traduire un type similaire de phrases de la langue source vers la langue cible. Et si une phrase précédemment traduite se reproduit, la même traduction est susceptible d'être à nouveau correcte.

- **Traduction automatique basée sur le contexte (CBMT) :** La traduction automatique contextuelle est produite comme une technique basée sur des corpus qui n'oblige ni principes ni corpus parallèles. Au lieu de cela, un cadre basé sur la connexion CBMT est nécessaire avec ses paramètres :
    - ✓ Un très grand corpus de textes cibles monolingues.
    - ✓ Un dictionnair bilingue complet.
    - ✓ le plus petit corpus de texte source monolingue doit diriger son algorithme

5. Comparaison entre RBMT et SMT

On a décidé de faire la comparaison entre RBMT et SMT par ce qu'ils sont les approche les plus utilisées dans la traduction automatique :

La première différence principale c'est le type de données utilisé dans l'implémentation de chacun des deux systèmes, le RBMT utilise des règles précisé et bien définie par les chercheurs, le SMT dépend dans son implémentation sur des corpus parallèles (paires de phrases traduites) des deux langues, cela qui fais que RMBT traduise les phrases données mot par mot alors ce système ne pourra jamais fournis une traduction correct si il trouve un mot qu'il n'existe pas auparavant dans le dictionnaire de la traduction par contre le SMT traduise la même phrase correctement à l'aide de son corpus intermédiaire[44], ces données ont l'impact direct sur la précision des système, ce qui fait que le SMT et plus performant et eu plus de précision que le RMBT, et par conséquence, la traduction automatique statique utilise généralement des corpus massive qui nécessite un matériel avec des capacités énormes et ça coute très chère par contre, la traduction automatique basée sur des règles et adapté dans des petits projets qui utilise des corpus simple facile a implémenter à l'aide des machine ordinaire. Le prochain tableau va contenir plus de comparaison selon d'autres critères :[44]

Traduction automatique basée sur de règles	Traduction automatique statique
cohérente et qualité prévisible.	Qualité de traduction imprévisible.
Connaît la grammaire règles.	Ne connaît pas la grammaire.
Cohérence entre les versions.	Incohérence entre les versions.
Manque de fluidité.	Bonne fluidité.
Difficile de gérer les exceptions aux règles.	Bon pour attraper les exceptions aux règles.
Coûts de développement et de personnalisation élevés.	Développement rapide et rentable, coûts, à condition que le corpus requis existe.

Tableau 2. 1 défirence entre RBMT et SMT

6. Évaluation d'un système TAP:

Le fait d'évaluer le système de Traduction automatique est important pas seulement pour ses utilisateurs potentiels et acheteurs, aussi aux chercheurs et aux développeurs, et pour évaluer la qualité des systèmes de traduction automatique, plusieurs approches sont utilisées.

6.1. L'évaluation humaine

Lors d'une évaluation humaine de la traduction automatique, plusieurs participants évaluent chaque traduction en fonction de critères précis. Les critères de qualité peuvent être multiples et inclure, par exemple, des critères de correction grammaticale et du sens du texte. hâter [45] présente un des critères humains habituellement utilisés, qui est une mesure manuelle avec laquelle les humains ne classent pas directement les traductions, mais plutôt génèrent une nouvelle traduction de référence qui est plus proche de la sortie du système en conservant la fluidité du sens de la référence. Cette référence permet de calculer les erreurs produites par le système, en le comparant avec la traduction automatique. Ces critères de qualité constituent la vraie mesure de la qualité du système, mais requièrent une coûteuse intervention humaine. Par ailleurs, toute évaluation subjective souffre des problèmes de non-reproductibilité et de variabilité inter-annotateur. C'est pourquoi plusieurs mesures

automatiques ont été développées au fil des années. Leur objectif est d'être corrélé avec les scores que produirait une évaluation humaine, tout en étant reproductible, beaucoup moins coûteuse et rapide pour pouvoir optimiser et comparer les systèmes.

6.2. Évaluation automatique

Pour évaluer la qualité des systèmes de traduction automatique, plusieurs approches sont utilisées :

6.2.1 BLEU (BiLingual Evaluation Understudy) :

La métrique BLEU, proposée par Papineni en 2001 était la première mesure automatique acceptée comme une référence pour l'évaluation de traductions. Le principe de cette méthode doit calculer le degré de similarité entre le candidat (la machine) la traduction et une ou plusieurs traductions de référence basées sur la précision de n-gramme particulière. Le score de BLEU est défini par la formule suivante [46]:

$$\text{BLEU} = \text{BP} \times e^{\left(\sum_{n=1}^{N} w_i \log pn\right)}$$

Où :

"pn" : le nombre de n-grammes de traduction automatique est présent aussi dans une ou plusieurs traduction de référence, divisée par le nombre de n-grammes totaux de traduction automatique.

"wi" : poids positifs.

"BP" : la Peine de Brièveté, qui pénalise des traductions pour être "trop court'. La peine de brièveté est calculée sur le corpus entier et a été choisie pour être une décomposition exponentielle dans "r/c", où "c" est la longueur de la traduction de candidat et "r" est la longueur efficace de la traduction de référence.

$$BP = \begin{cases} 1 & \text{Si } c > r \\ e^{1-\frac{r}{\varepsilon}} & \text{Si } c \leq r \end{cases}$$

6.2.2 WER (Word Error Rate) :

La métrique WER, Proposée par Popovic et Ney en 2007. À l'origine utilisée dans la Reconnaissance de la parole Automatique, elle sent a évaluer la performance en terme de taux

d'erreurs au niveau des mots .le taux(WER)est calculé par une distance appelée de levenshtein , qui comptabilise le nombre minimal d'opération qu'il faut effectuer pour passer de la traduction produite a la traduction, en considérant trois opérations qui sont l'insertion, la suppression et la substitution qui reçoivent trois le m poids de référence mesurés. Pour cela, l'idée est de calculer le nombre minimal de révise (l'insertion, l'effacement ou la substitution du mot) pour être exécuté sur la traduction d'hypothèse pour le rendre identique à la traduction de référence. Le nombre d'édits à être exécuté, noté "dL (en ce qui concerne, hyp)" est alors divisé par la grandeur de la traduction de référence, a dénoté "Nref" comme montré dans la formule suivante [47] :

$$\mathrm{WER} = \frac{1}{N_{raf}} \mathrm{x} \mathrm{d_L}(\mathrm{ref}, \mathrm{hyp})$$

Où :

dL (en ce qui concerne, hyp) : est la distance Levenshtein entre la traduction de référence "en ce qui concerne" et l'hypothèse traduction "hyp". Un manque du WER est le fait qu'il ne permet pas de réordonner des mots, alors que l'ordre de mot de l'hypothèse peut se distinguer de l'ordre de mot de la référence bien que ce soit la traduction correcte.

6.2.3 PER (Position-independent word Error Rate) :

La PER métrique, proposé par Tillman en 1997. Comparez les mots de traduction automatique avec ceux de la référence sans tenir compte de leur ordre dans la phrase. Le PER score est défini par la formule suivante [48]:

$$\mathrm{PER} = \frac{1}{N_{\mathrm{ref}}} \times \mathrm{d_{per}}(\mathrm{ref}, \mathrm{hyp})$$

Où :

$\mathrm{d_{per}}$: calcule la différence entre les occurrences de mots dans la traduction automatique et la traduction de référence.

Un manque du PER est le fait que l'ordre de mot peut être important dans certains cas.

6.2.4 TER (Translation Error Rate) :

La métrique TER, proposé par Snover en 2006. Est défini comme le nombre minimal de révise devait changer une hypothèse pour qu'il corresponde exactement à une des références. Le possible révise dans TER incluent l'insertion, l'effacement et la substitution de mots simples et du fait de réviser qui déplace des ordres de mots contigus. Normalisé par la longueur moyenne des références. Puisque nous sommes concernés avec le nombre minimal de révise devait modifier l'hypothèse, nous mesurons seulement le nombre de révise à la référence la plus proche. Le score de TER est défini par la formule suivante [49]:

$$\mathrm{TER} = \frac{\mathrm{Nb(op)}}{\mathrm{AvregN}_{\mathrm{Ref}}}$$

Où :

Nb (op) : est le nombre minimal de révise.

$\mathrm{AvregN}_{\mathrm{Ref}}$: la grandeur moyenne dans les références de mots.

7. Les logiciels de la traduction automatique

Dans ce tableau on découvre les logiciels de la traduction automatique les bien réputées:

Nom	Description	Nombre de langue généré
Google Translate	Un des services de traduction en ligne les plus populaires est offert par Google, il contient plusieurs caractéristiques que vous aimeriez incluent l'économie, l'écoutant, le partageant, ou la duplication du texte traduit. Mais il ne peut pas générer une grande quantité de texte.	Plus de 100 langues.
Bing Translator	Un autre grand nom dans les traducteurs est Bing, qui utilise le Traducteur de Microsoft. Vous pouvez choisir votre langue de contribution ou avoir le site le découvrent automatiquement comme vous tapez. Si vous faites permettre votre microphone, vous pouvez parler le texte vous voulez être traduits, qui est convenable. Après que vous recevez la traduction, vous avez des options pour l'entendre à haute voix d'une voix mâle ou femelle, le partager, ou fouiller Bing avec cela	Plus de 60 langues.
	Sur Translatedict, vous pouvez choisir l'utilisation auto découvrent pour votre propre dialecte. Entrez juste dans	

Translatedict	votre mot, expression, ou une grande quantité de texte, choisissez la langue de traduction et frappez le bouton de Translate. Vous verrez la traduction écrite et pouvez claquer le bouton solide pour l'entendre à haute voix. Translatedict fournit aussi des régions uniquement au traducteur de voix et à la caractéristique de texte-à-discours. Plus, vous pouvez demander l'aide avec les traductions professionnelles et recevoir une citation en remplissant la forme en ligne.	Plus de 50 langues
Translate.com	Un bon traducteur qui utilise le service de Microsoft. Vous pouvez utiliser votre voix ou clavier pour entrer dans le texte, lire ensuite ou écouter la traduction. Si vous croyez que la traduction devrait être reconsidérée, vous pouvez obtenir une traduction humaine avec les 100 premiers mots libres. Claquez juste l'icône de contact et signez le registre ou créez un compte.	Plus de 30 langues
DeepL Translator	Le Traducteur DeepL est un instrument vraiment frais avec ses définitions et options d'achèvement de sentence automatiques. Quand vous recevez la traduction, cliquez juste deux fois sur un mot pour plus de détails. Quand vous choisissez ce mot dans la traduction, vous verrez une boîte de dropdown avec plus d'options. Vous pouvez aussi jeter un coup d'œil à la définition de mot qui surgit au fond de la page en même temps. Plus, vous verrez des exemples du mot étant utilisé tant dans les langues de production que dans la contribution.	26 langues
Babylon Online Translator	Pendant que Babylone offre vraiment le logiciel que vous pouvez télécharger pour les traductions, vous pouvez aussi régler son option en ligne. Avec une option d'échange simple, le site peut ne pas avoir les cloches et les sifflets comme d'autres, mais est annoncé être tout à fait exact. Si votre situation d'affaires pourrait profiter d'un traducteur professionnel, Babylone offre ce service aussi. Claquez juste le bouton d'Human Translation sur la page de traducteur en ligne et vous serez dirigés vers cette section du site pour les détails	plus de 75 langues
PROMT Online Translator	PROMT le Traducteur En ligne n'offre pas autant de langues les autres traducteurs, mais il a vraiment d'autres caractéristiques agréables. Utilisez la détection de langue automatique et choisissez même un thème pour la traduction. Vous pouvez alors copier, coller, vérifier l'orthographe, ou accéder à un dictionnaire. Il y a aussi un clavier virtuel ainsi si vous utilisez le site sur une tablette, par exemple, en éclatant dans vos mots ou sentences est simple. PROMT offre aussi le logiciel de traduction que vous pouvez acheter et télécharger.	20 langues

Tableau 2. 2 Les logiciels de la traduction automatique [50]

8. Conclusion :

Dans ce chapitre on a bien détaillé les côtés majeurs dans le domaine de la traduction automatique tel que les approches existantes, l'usage de chaque système, et les différentes mesures d'évaluation des systèmes de TAP,

On peut déterminer que la traduction automatique et l'un des domaines de recherche les plus importants dans le traitement automatique du langage naturel (TALN), ce qui nous a mener à relever le défi et a créer une application web pour la TAP de l'anglais vers l'Arabe. Dans les chapitres suivants on va présenter en détaille toute les phases nécessaires pour construire cette application, et on va commencer par la conception qui est le titre du chapitre suivant.

Chapitre III:

Conception et Modélisation

1. Introduction

dans ce chapitre nous allons présenté l'architecture générale et la méthode utilisée pour la conception de notre application a cet effet ,nous avons utilisé les diagrammes du langage UML (Unified Modeling Langage) et plus précisément : le diagramme de cas d'utilisation, le diagramme de séquence et le diagramme des paquets .

concernent la méthode de conception,on a choisi le processus unifié (Unified Process :UP).

2. Le procéssus unifié (UP)

Le processus unifié est un processus de développement logiciel itératif, centré sur l'architecture, piloté par des cas d'utilisation et orienté vers la diminution des risques. C'est un patron de processus pouvant être adapté à une large classe de systèmes logiciels, à différents domaines d'application, à différents types d'entreprises, à différents niveaux de compétences.

3. Le Langage de modélisation unifié UML

Le langage de modélisation unifié (UML), est un langage de modélisation graphique conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet. Il en existe quatorze diagramme UML :

- Diagrammes de structure ou diagrammes statiques :
  - ✓ Diagramme de classes.
  - ✓ Diagramme d'objets.
  - ✓ Diagramme de composants.
  - ✓ Diagramme de déploiement.
  - ✓ Diagramme des paquetages.
  - ✓ Diagramme de structure composite.
  - ✓ Diagramme de profils.
- Diagrammes de comportement :
  - ✓ Diagramme des cas d'utilisation.
  - ✓ Diagramme états-transitions.
  - ✓ Diagramme d'activité.
- Diagrammes d'interaction ou diagrammes dynamiques :

- ✓ Diagramme de séquence.
- ✓ Diagramme de communication.
- ✓ Diagramme global d'interaction.
- ✓ Diagramme de temps.

4. L’approche proposée

4.1. Diagramme des cas d'utilisation (use case)

Le diagramme de cas d'utilisation (use case) et un diagramme UML utilisés pour une représentation du fonctionnement et le développement de l’interface du système. et pour le développement,

les cas d'utilisation sont plus appropriés. En effet, un cas d'utilisation (use cases) représente une unité discrète d'interaction entre un acteur (humain ou machine) et un système. Ainsi, dans un diagramme de cas d'utilisation, les utilisateurs sont appelés acteurs et ils apparaissent clairement dans les cas d'utilisation.

4.1.1 Identification des acteurs et leurs rôles

L’acteur C’est une entité externe qui interagisse avec le système.

Acteur	Rôle
Utilisateur	Recorder un fichier wav Sélectionner le ficher wav Télécharger le résultat .wav Ecouter le résultat

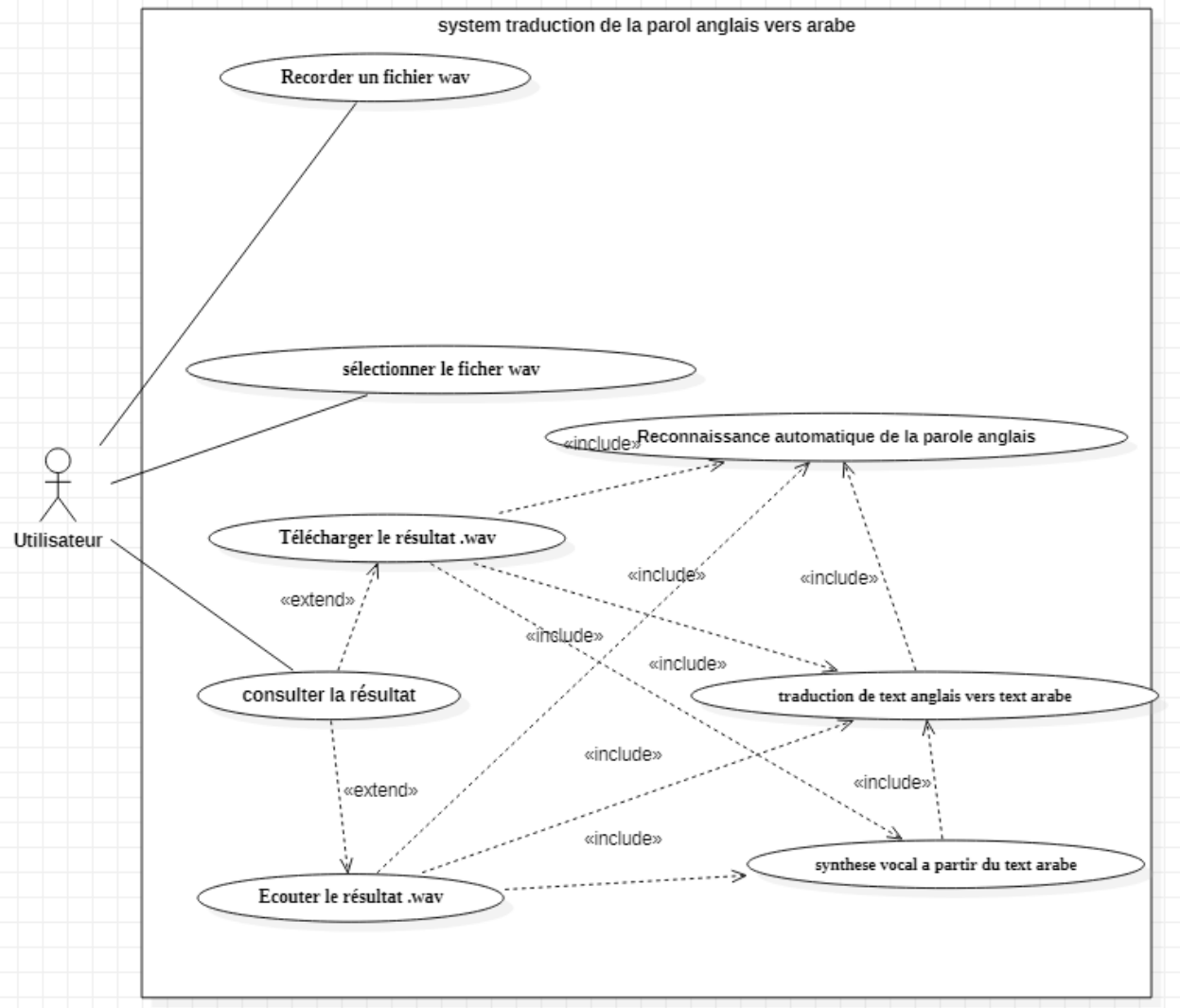

Figure 3. 1 Diagramme de cas d'utilisation

4.2. Diagramme de séquence consulter la résultat

Les diagrammes de séquences permettent de décrire comment les éléments du système interagissent entre eux et avec les acteurs. Les objets au cœur d'un système interagissent en échangeant des messages. Les acteurs interagissent avec le système au moyen d'IHM (interface hommes machine). La figure suivant illustre notre diagramme de séquence globale

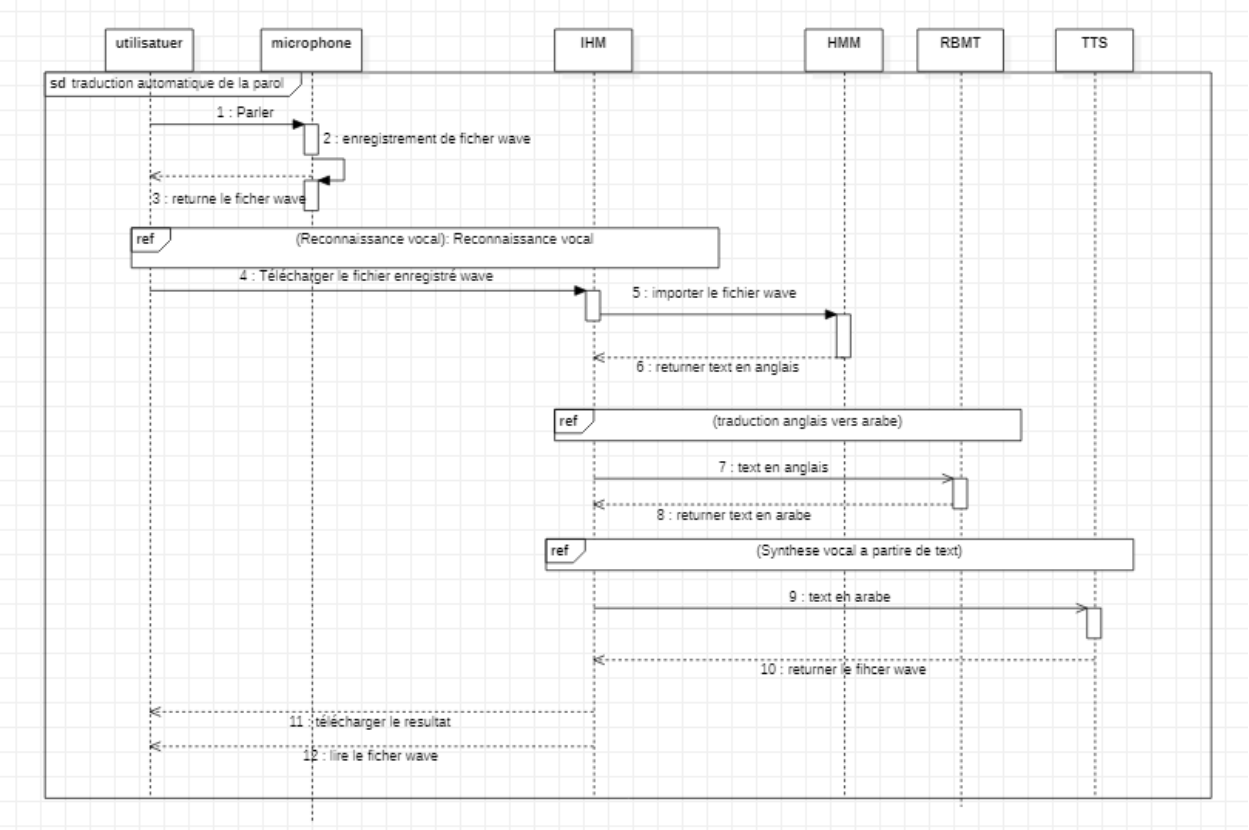

Figure 3. 2 Diagramme de séquence consulter la résultat

4.2.1 Diagramme de séquence de Reconnaissance de la parole en anglais

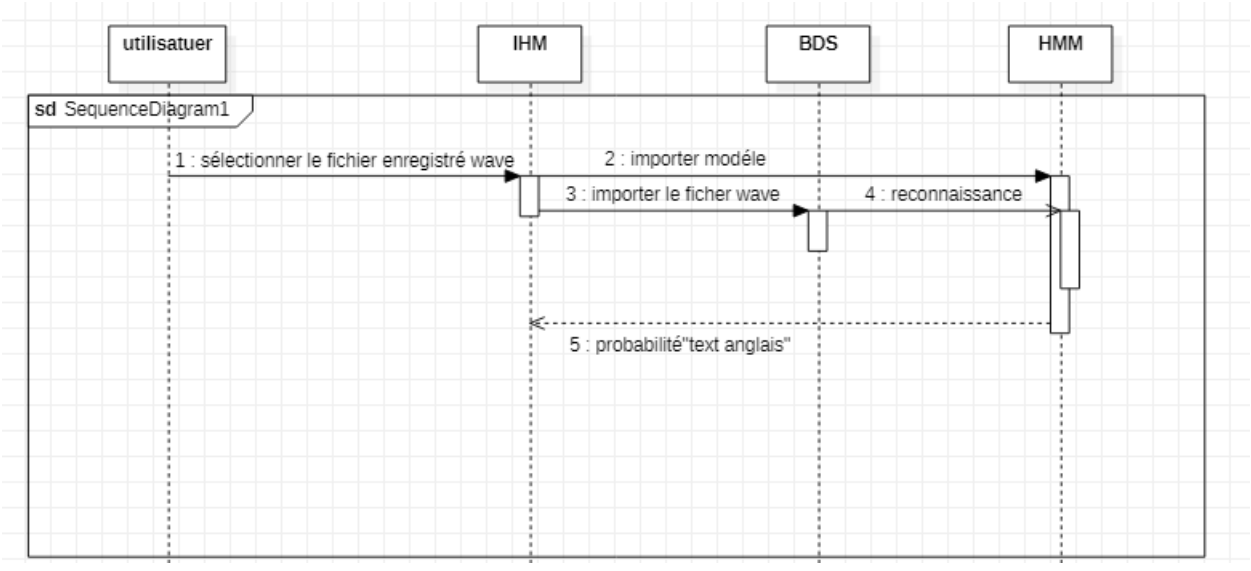

Figure 3. 3 Diagramme de séquence (Reconnaissance de la parole en anglais)

4.2.2 Diagramme de séquence de Traduction de texte anglais vers texte arabe

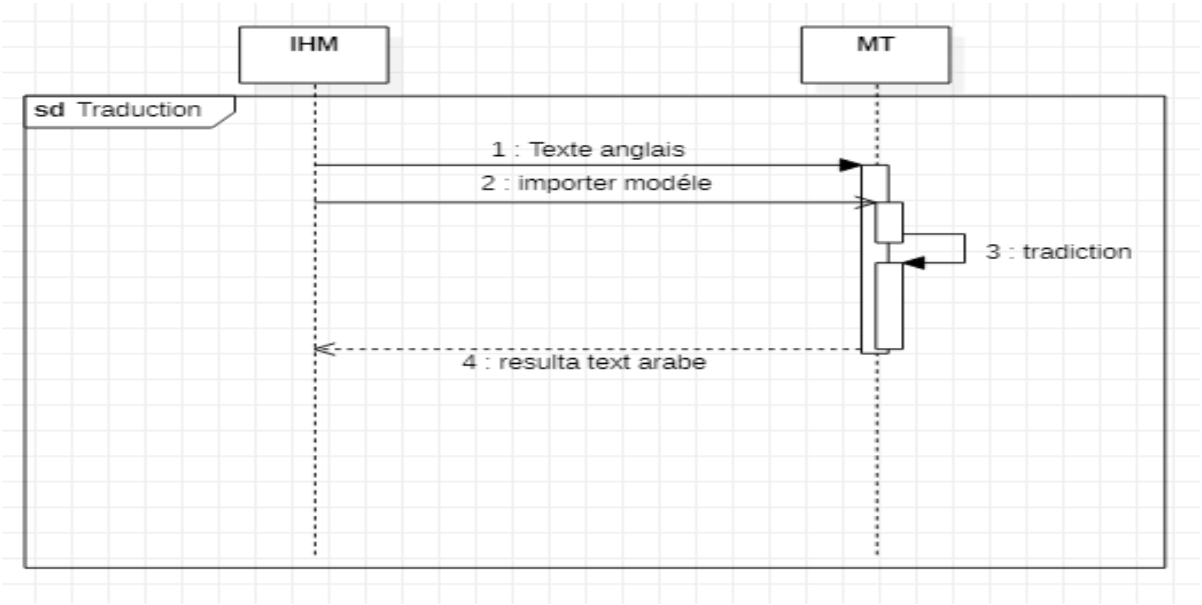

Figure 3. 4 Diagramme de séquence (Traduction de texte anglais vers texte arabe)

Diagramme de séquence de Synthèse vocal a partir du texte arabe

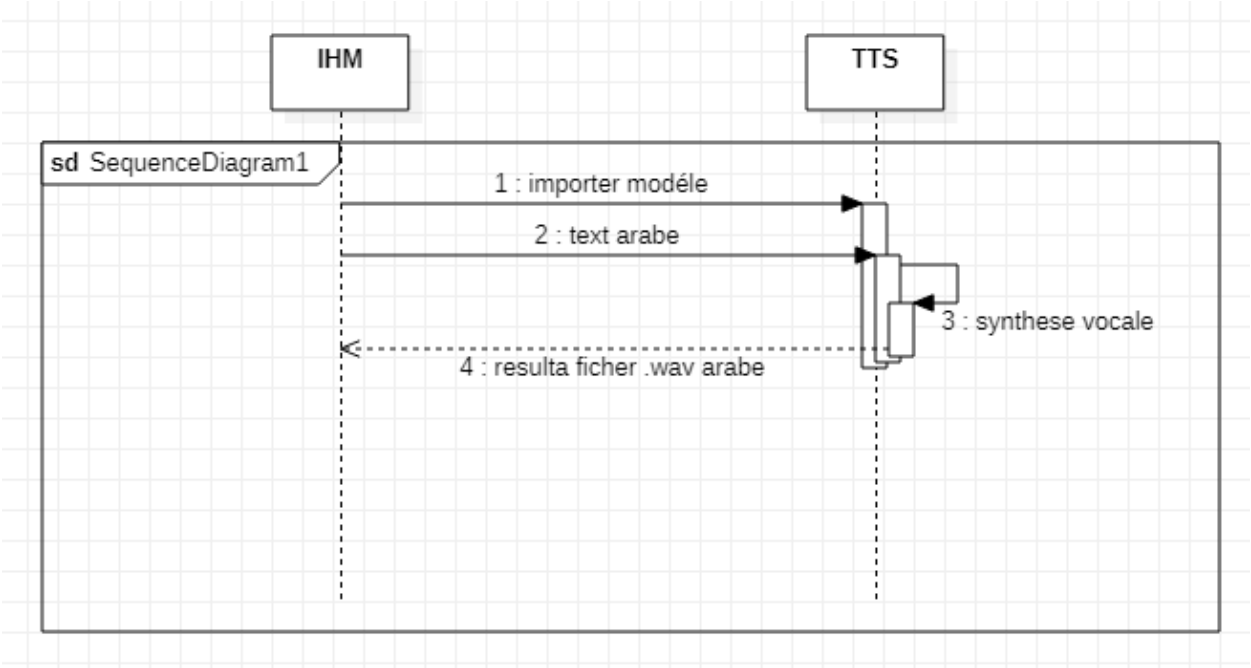

Figure 3. 5 Diagramme de séquence (Synthèse vocal a partir du texte arabe)

4.3. Diagramme de paquetage (package)

Les diagrammes de paquetages sont la représentation graphique des relations existant entre les paquetages composant un système, la figure3.6 illustre bien notre diagramme de paquetage

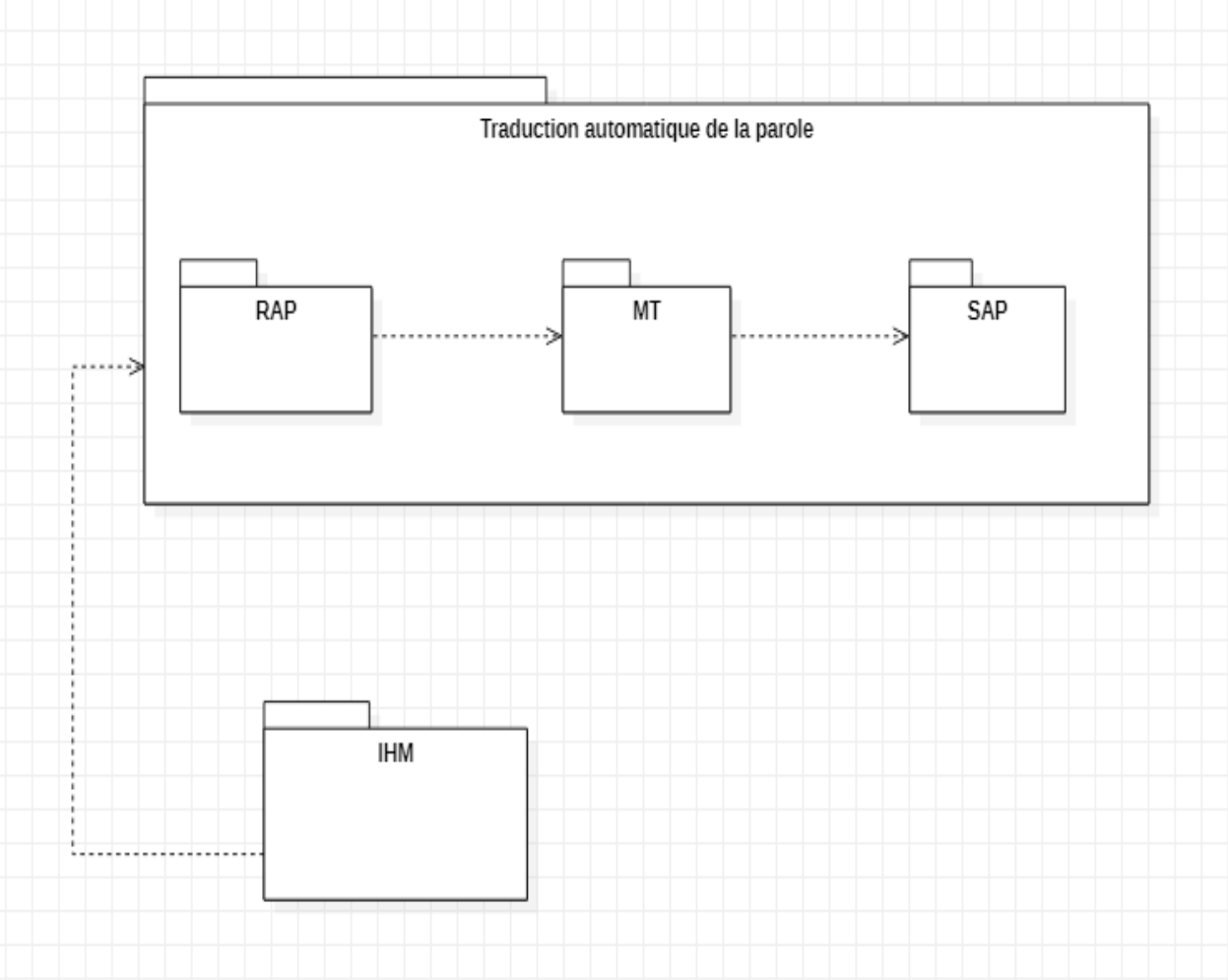

Figure 3. 6 Diagramme de paquetage

RAP : Système de reconnaissance automatique de la parole.

MT : (machine translation) Système de traduction automatique.

SAP : Système de synthèse vocal à partir de texte.

IHM : Interface homme machine.

5. Conclusion

Dans ce chapitre on a présenté le schéma général et la méthode suivie pour mener à bien réaliser ce projet, on a bien décrire les trois (03) diagrammes utilisés en démontrent les acteurs et leurs rôles et les différentes entités de notre système, ainsi qu'on a bien détaillé l'enchainement dans le temps d'exécution de notre application.

Dans le chapitre suivant on va spécifier les détails techniques de notre application.

Chapitre IV:

Réalisation et expérimentation

1. Introduction

Dans ce chapitre on va présenter les étapes nécessaires pour construire notre application « translator », la création du système de reconnaissance de la parole anglaise,la traduction text-to-text de l'anglais vers l'arabe et la synthèse vocal arabe. On va présenter aussi l'interface graphique et expliquer comment l'utilisateur peut interagir avec notre systéme.

2. L'environnement de développement

Le choix de l'environnement de programmation convenable est très important pour le développement des projets. Cela se fait suivant plusieurs facteurs : la facilité d'utilisation, la disponibilité de plusieurs fonctionnalités et plusieurs bibliothèques, la communication avec d'autres environnements… etc.Dans notre cas a utilisé :

2.1. Environnement matériel

PC de bureau :

- CPU: AMD Ryzen™ 3 3200G @3.6GHz.
- RAM: XPG 8 GB 3000MHz DDR4.
- GPU: Vega 8 Graphics.
- Disk: SSD 120 GB/HDD 350 GB.
- Système d'exploitation : Windows 10 professionnel 64 bits.

Un ordinateur portable de type HP :

- CPU: Intel core i3-3217U 1.8GHz.
- RAM : 4GB.
- Disk : HDD 500GB.
- Système d'exploitation : Windows 10 professionnel 64 bits.

2.2. Environnement logiciel

• Visual studio code

Pour écrire nos code python et page web, nous avons utilisé ce IDE36 qui est très simple et léger qui ne consomme pas trop de mémoire. La communauté développe plusieurs outils comme « Kite37 » qui est un système de prédiction intelligent pour les codes en python ce

qui aide avec la productivité, un terminal puissant et une très bonne ergonomie. On a utilisé la version 1.60.0 figure 4.1

Figure 4. 1 visual studio code logo

- **Excel**

Excel est un logiciel de la suite bureautique Office de Microsoft et permet la création de tableaux, de calculs automatisés, de plannings, de graphiques et de bases de données. On appelle ce genre de logiciel un "tableur". On a utilisé Excel 2010 figure 4.2 pour créer les fichiers .fileids, .transcription, .dic, .phone …

Figure 4. 2 Excel logo

- **Notepad ++**

Notepad++ est un éditeur de texte libre générique, fondé sur la composante Scintilla, fonctionnant sous Windows, codé en C++, qui intègre la coloration syntaxique de code source pour les langages et fichiers C, C++ Scheme, Properties, Diff, Smalltalk, PostScript et VHDL .. ainsi que pour tout autre langage informatique, car ce logiciel propose la possibilité de créer ses propres colorations syntaxiques pour un langage quelconque.[1] On a utilisé

[1] Notepad++ ,notepad-plus-plus.org/

notepad++ v8.1.3 figure 4. 3 pour réglé les fichiers .fileids, .transcription, .dic, .phone et le parametres de fichier de configuration pour l'apprentissage de reconnaissance....

Figure 4. 3 Notepad++ logo

- **zebNet Duplicate Line Remover**

Le Déménageur de Ligne de Double de zebNet vous permet de facilement enlever des lignes en duplicata d'un dossier de texte.[2] On a enlevé tous les doublant qui existe dans notre corpus.

- **Cmder**

Cmder est un paquet de logiciel créé de la frustration pure sur l'absence d'émulateurs de console agréables sur Windows, on a l'utilisé pour l'installation des bibliothèques et le lancement des apprentissages.

Figure 4. 4 cmder logo

- **Google Colab (Colaboratory)**

Est un service offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.

[2] https://www.zebnet.co.uk/freeware/tools-and-utilities/duplicate-line-remover

Figure 4. 5 googel colab logo

- **Praat**

Praat est un logiciel libre scientifique gratuit conçu pour la manipulation, le traitement et la synthèse de sons vocaux. Il a été conçu à l'institut de sciences phonétiques de l'université d'Amsterdam par Paul Boersma et David Weenink. Il peut s'exécuter sur un grand nombre de plates-formes. Praat est écrit en C++[3].

Figure 4. 6 Praat logo

2.3. Langage de programmation et bibliothèque

Pour réaliser ce projet, nous avons utilisé plusieurs Langage de programmation et bibliothèque

- **Python**

Python est une programmation interprétée voir figure(4.7) , de haut niveau et est très polyvalent. Ce langage est dynamiquement typé et récupéré. Il prend en charge plusieurs paradigmes de programmation, y compris séquentielle, orientés objet et programmation fonctionnelle. Python est un langage qui présente une très grande bibliothèque, variée et robuste pour plusieurs types de développement. Pour le domaine du TAL, python représente un choix indiscutable avec tout ce qui offre comme librairies et facilité d'utilisation, la

[3] https://www.fon.hum.uva.nl/praat/

communauté lui accorde beaucoup de temps, c'est donc pour ça que nous avons choisis de l'utilisé comme outil principale du développement de notre backend. La version utilisée est python 3.9, mais nos codes ont été conçus de façon à fonctionner correctement avec les versions plus anciennes jusqu'au 2.7.

Figure 4. 7 Pyhton logo

- **HTML**

Signifie « HyperText Markup Language » qu'on peut traduire par « langage de balises pour l'hypertexte ». Il est utilisé afin de créer et de représenter le contenu d'une page web et sa structure. HTML fonctionne grâce à des « balises » qui sont insérées au sein d'un texte normal. Chacune de ces balises indique la signification de telle ou telle portion de texte dans le site. HTML permet d'inclure des images et d'autres contenus dans les pages web. Grâce à HTML, chacun peut créer des sites web aussi bien statiques que dynamiques voir Figure (4.9).

Figure 4. 8 HTML logo

- **CSS**

Le CSS (Cascading Style Sheets) est un langage informatique utilisé sur l'internet pour mettre en forme les fichiers HTML ou XML. Ainsi, les feuilles de style, aussi appelé les fichiers CSS, comprennent du code qui permet de gérer le design d'une page en HTML voir Figure (4.9).

Figure 4. 9 CSS logo

- **Django**

Django est un Framework python open-source dédié au développement web. Il est orienté pour les développeurs ayant comme besoin de produire un projet solide rapidement et sans surprise. Django offre une liaison parfaire avec python. Il offre aussi des modèles déjà prêts à utiliser directement et/ou étendre pour satisfaire nos besoins.[4]

- **Pyttsx3**

Pyttsx3 (2.90) est une bibliothèque de conversion de texte-à-discours dans le Python. À la différence des bibliothèques alternatives, cette biblioteque travaille hors ligne elle est compatible tant avec le Python 2 qu'avec 3.[5] pour l'installer on utilise la commande « pip install pyttsx3 ».

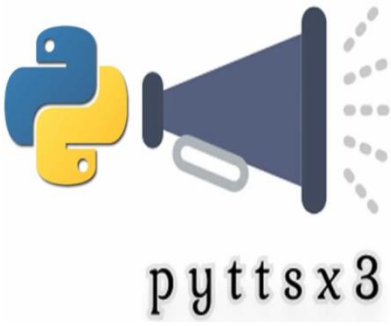

Figure 4. 10 pyttsx3 logo

[4] https://www.djangoproject.com/
[5] pypi.org/project/pyttsx3/

- **Pocketsphinx**

Pocketsphinx est une partie du Sphinx d'Université Carnegie Mellon c'est la Trousse à outils c'est une Source Ouverte Pour la Reconnaissance de la parole.

Ce paquet fournit une interface de python à l'Université Carnegie Mellon Sphinxbase et les bibliothèques Pocketsphinx créées avec SWIG et Setuptools.[6]

3. Corpus

Dans notre projet on a utilisé trois (03) corpus

3.1. Le corpus de reconnaissance

on a choisis le corpus ''Fluent Speech Commands'' qui contient 30,043 enregistrements. Il est enregistré sous forme de fichiers .wav mono canal de 16 khz , contenant chacun un seul énoncé utilisé pour contrôler les appareils ménagers intelligents ou assistant pour maisons intelligentes .[7]

Notre dataset contient 101 locuteurs.voir les figures

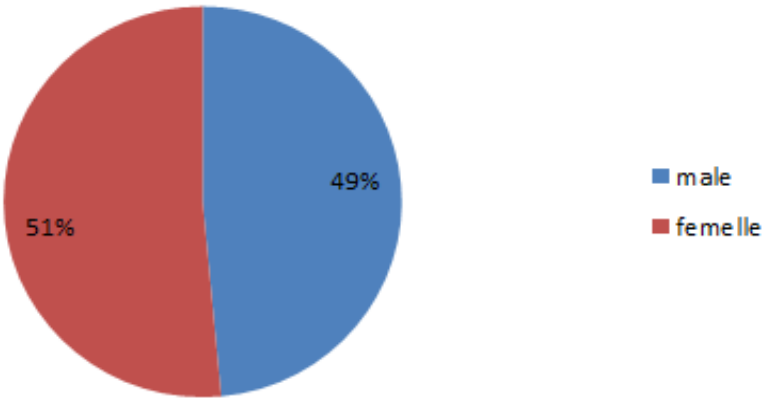

Figure 4. 11 Le partage des pourcentage des locuteurs.

Qui sont divisé en trois (03) tranches d'age voir le figure (4.12)

[6] pypi.org/project/pocketsphinx/
[7] https://fluent.ai/fr/fluent-speech-commands-a-dataset-for-spoken-language-understanding-research/

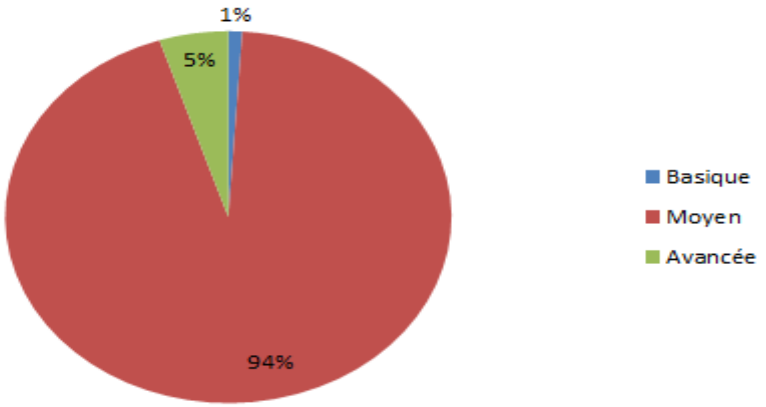

Figure 4. 12 Tranches d'âge des locuteurs

Et chaque locuteur a un niveau de fluidité auto-déclaré différent voir le figure (4.13)

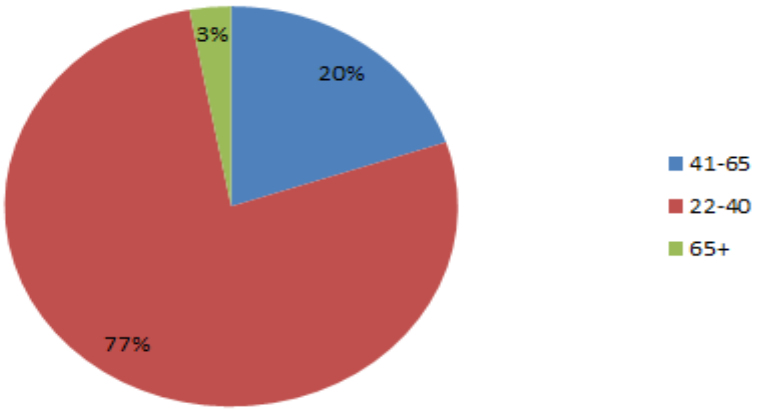

Figure 4. 13 niveau de fluidité auto-déclaré

Ce tableau montre la division des paroles entre apprentissage et test

	Nombre de parole
Apprentissage	**26250**
Teste	**3793**

Tableau 4. 1 La division des paroles

3.2. Corpus de traduction

Une fois le prétraitement de notre dataset de reconnaissance est achevé (élimination des mots répétitifs), on a pu construire notre corpus par la traduction manuelle des résultats obtenus. Ce corpus contient 101 mots, voir la figure (4.14)

1	CHANGE	1	تغيير
2	LANGUAGE	2	لغة
3	RESUME	3	استئناف
4	TURN	4	تشغيل
5	THE	5	ال
6	LIGHTS	6	أضواء
7	ON	7	فتح
8	SWITCH	8	تحول
9	OFF	9	إيقاف
10	VOLUME	10	الصوت

Figure 4. 14 Exemple du corpus de traduction

3.3. Corpus de synthése vocal

On a utilisé les voix de synthèse vocale Microsoft qui sont des synthétiseurs vocaux fournis pour être utilisés avec des applications qui utilisent l'API Microsoft Speech (SAPI) ou la plate-forme Microsoft Speech Server. Les voix client sont livrées avec les systèmes d'exploitation Windows.[8]

4. Architecture fonctionnelle

Dans le cadre de notre travail, nous développons une application web qui offrira une traduction instantané de la parole de l'anglais vers l'arabe voir le figure .

[8] Microsoft text-to-speech voices,www.webbie.org.uk/texttospeech.

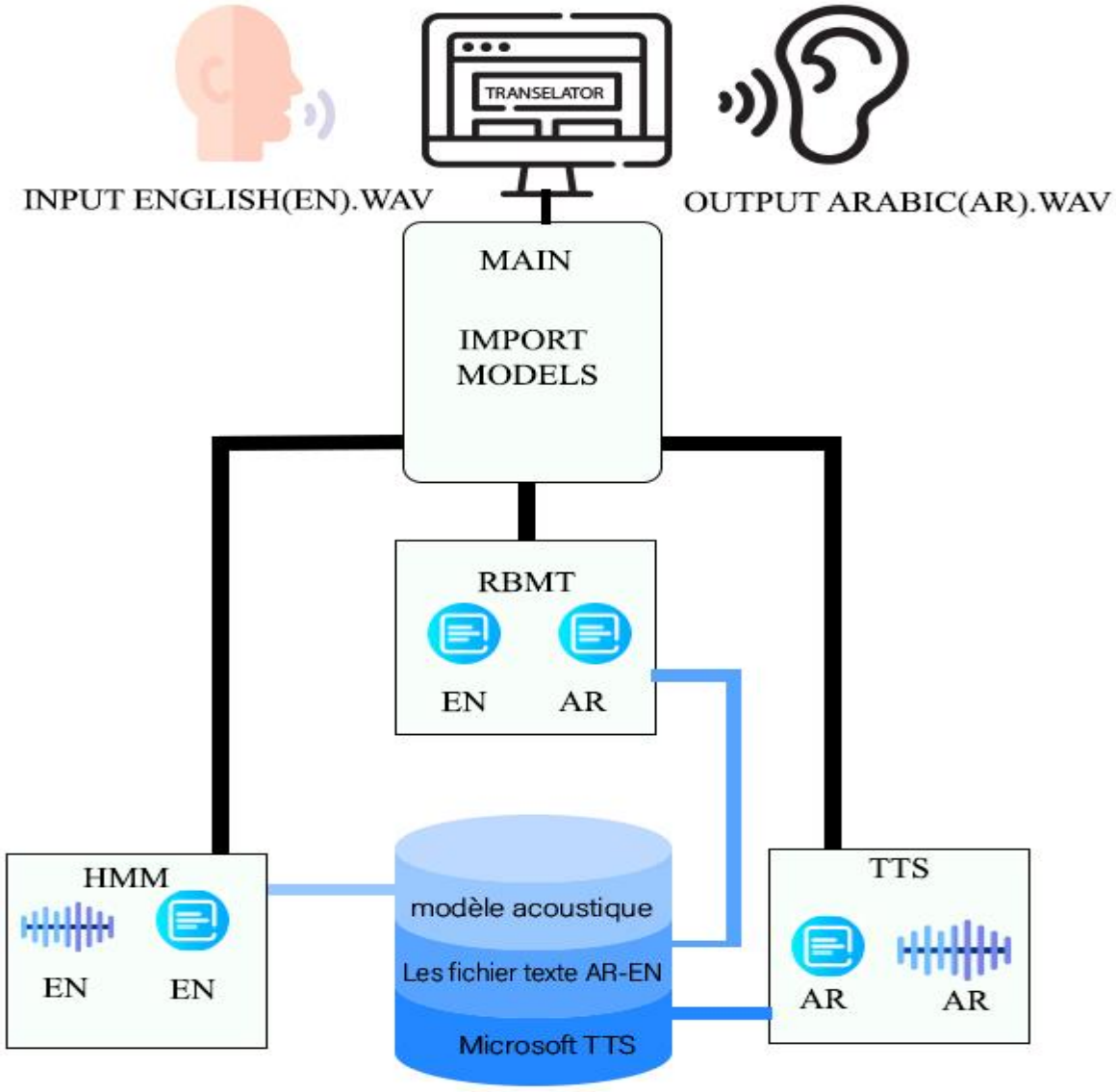

Figure 4. 15 Schéma de fonctionnement général de TRANSLATO

5. Développement

Pour la première phase de notre projet qui consiste à entrainer notre propre modèle acoustique on a utilisé la bibliothèque reconnaissance vocale appelé CMUSPHINX .

Ensuite dans la 2eme phase de traduction on a utilisé une traduction basée sur des règles (RBMT) qui est un système de traduction automatique basé sur des informations linguistiques sur les langues source et cible essentiellement extraites de dictionnaires et de grammaires.

Enfin notre 3eme phase représente la synthèse vocal dans cette dernière phase on a choisis s'utilisé la bibliothèque pyttsx3 fournis par python.

Et pour une utilisation optimale de notre système on a développé une interface graphique a l'aide de HTML, CSS et JAVASCRIPT.

5.1. Système de reconnaissance

L'apprentissage de notre modèle acoustique suivtre L'architecture suivant :

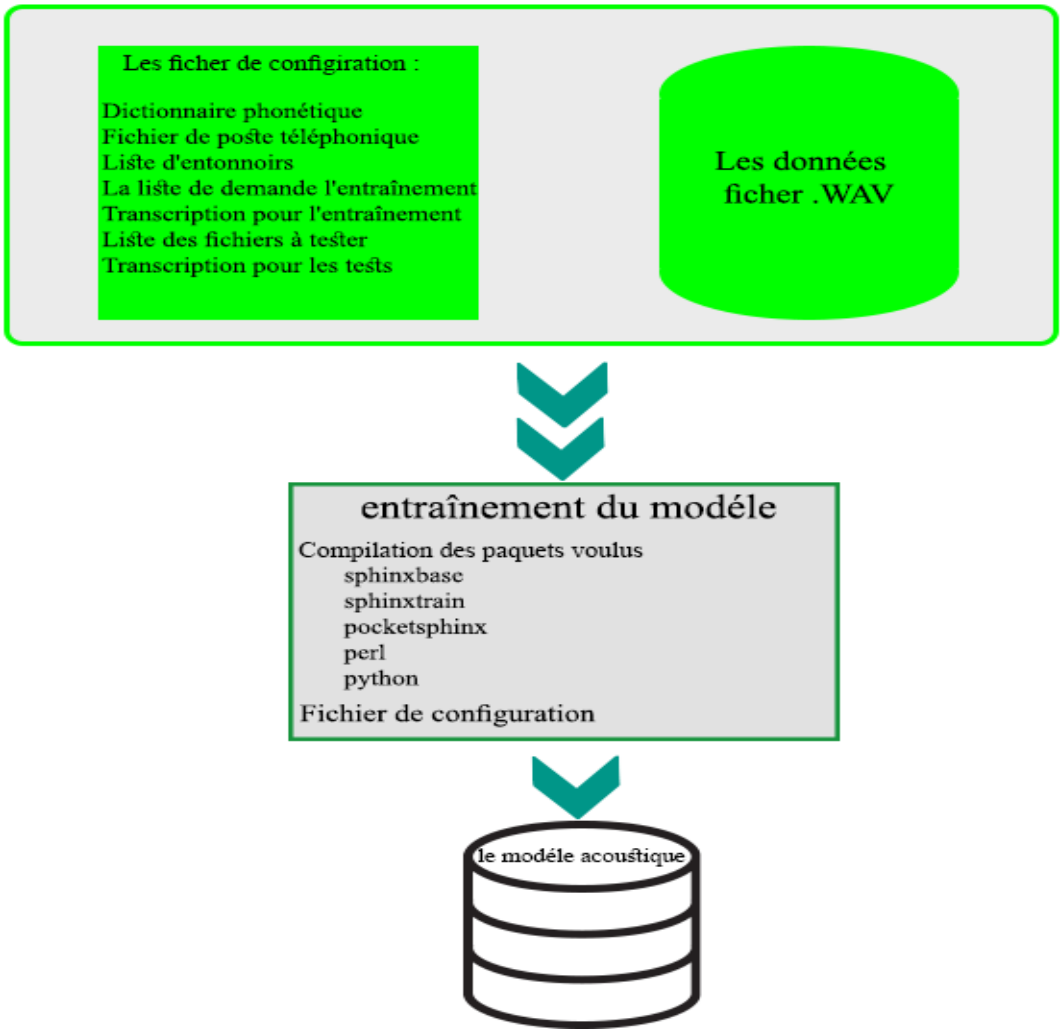

Figure 4. 16 Architecture d'apprentissage du modèle acoustique

5.1.1 Préparation des données

La structure des fichiers de la base de données illustrée dans la figure 4.17 suivante :

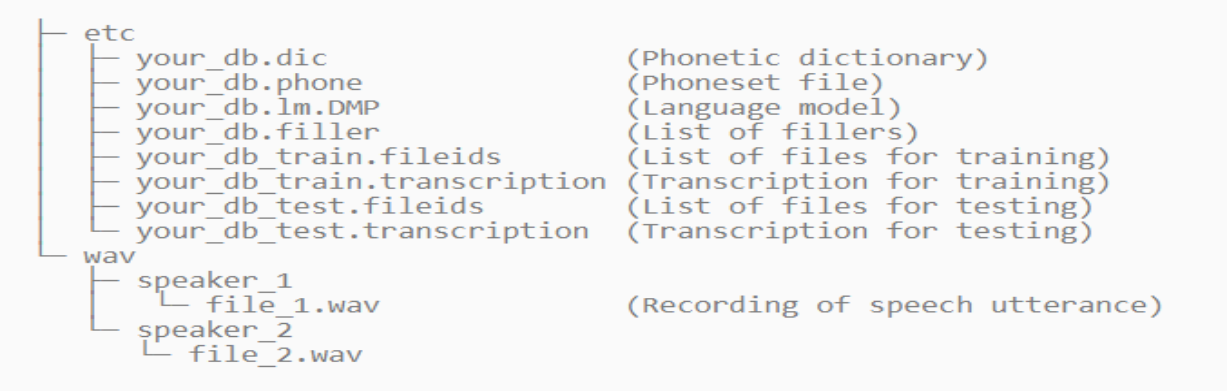

Figure 4. 17 La structure des fichier de la base de données[9]

- ***.fileids :** Les fichiers namedataset_train.fileids et namedataset _test.fileids sont des fichiers texte qui répertorient les noms des enregistrements un par un et qui contient le chemin dans un système de fichiers relatif au répertoire wav.
- ***.transcription:** Les fichiers namedataset_train.transcription et namedataset_test.transcription sont des fichiers texte répertoriant la transcription de chaque fichier audio.

Il est important que chaque ligne commence par <s> et se termine par </s> suivi d'un identifiant entre parenthèses. Notez également que les parenthèses ne contiennent que le fichier, sans le répertoire wav. Il est essentiel d'avoir une correspondance exacte entre le fichier *.fileids et le fichier *.transcription. Le nombre de lignes dans les deux doit être identique.

- **.dict :** devrait avoir une ligne par mot avec le mot après la transcription phonétique.

- **.wav :** les enregistrements audio doivent contenir un son d'entraînement qui doit correspondre à l'audio que nous souhaitons reconnaître à la fin. La durée optimale des enregistrements audio est comprise entre 5 et 30 secondes. La quantité de silence au début et à la fin de l'énoncé ne doit pas dépasser 200 ms. Les wav doivent être sous forme .WAV avec une fréquence d'échantillonnage spécifique – 16 kHz, 16 bits, mono.
- **.phone :** devrait avoir un phone par ligne, Le nombre de phones devrait correspondre aux phones utilisés dans le dictionnaire plus le phone de SIL spécial pour le silence.

[9] https://cmusphinx.github.io/

- **.lm.DMP :** devrait être dans le format d'ARPA ou dans le format de DMP.
- **.filler :** contient des phones d'entonnoir (pas - couvert par le modèle de langue non-linguistique comme l'haleine, “hmm” ou du rire). Il peut contenir juste fait taire.

5.1.2 Compilation des packages nécessaires

Les packages suivants sont requis pour l’apprentissage, ils doivent être téléchargé :

- Sphinx base
- Sphinx Train
- Pocket sphinx

Les langages suivants sont également requis, ils doivent être installés :

- Perl
- Python

5.1.2.1 Mise en place des scripts d’apprentissage

Pour démarrer l’apprentissage, on doit accéder au dossier de la base de données et exécutez les commandes suivantes à l’aide de cmder :

python ../sphinxtrain/scripts/sphinxtrain -t fluent setup

Cela copiera tous les fichiers de configuration requis dans le sous-dossier etc/ de votre dossier de base de données et préparera la base de données pour l’apprentissage.

Au cours du processus d’apprentissage, d'autres dossiers de données seront créés, et le répertoire de base de données va contenir etc, feat, logdir, model_parameters, model_architecture, result et wav.

Après cette configuration de base, nous devons éditer les fichiers de configuration dans le dossier etc/. Nous n'avons besoin d'en changer que quelques-unes. Tout d'abord, trouvez le fichier

etc/sphinx_train.cfg.

Après, on doit vérifier que les paramètres suivants(cfg : wavfile et dictionary et transcriptifile…) sont similaires à ça :

```
$CFG_WAVFILE_EXTENSION = 'wav';
$CFG_WAVFILE_TYPE = 'mswav'; # one of nist, mswav, raw


# Variables used in main training of models
$CFG_DICTIONARY     = "$CFG_LIST_DIR/$CFG_DB_NAME.dic";
$CFG_RAWPHONEFILE   = "$CFG_LIST_DIR/$CFG_DB_NAME.phone";
$CFG_FILLERDICT     = "$CFG_LIST_DIR/$CFG_DB_NAME.filler";
$CFG_LISTOFFILES    = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.fileids";
$CFG_TRANSCRIPTFILE = "$CFG_LIST_DIR/${CFG_DB_NAME}_train.transcription";
$CFG_FEATPARAMS     = "$CFG_LIST_DIR/feat.params";
```

Figure 4. 18 les paramètres similaires

5.1.2.2 L'apprentissage

Dans notre projet on a utilisé une partie (fluentv3) de la base de donnée (fluent) qui contient 10000 enregistrements d'entrainement et 2000 enregistrements de teste.

On doit suivre les prochaines commendes :

Pour accéder au répertoire fluentv3 :

- cd fluentv3

Pour commencer l'apprentissage (entrainement) :

- python ../sphinxtrain/scripts/sphinxtrain run

Cette base de données a pris 4 heures d'entrainement.

L'étape la plus importante est la première qui vérifie que tout est correctement configuré et que vos données d'entrée sont cohérentes voir Figure.

```
C:\sphinx\fluintv3
λ python ../sphinxtrain/scripts/sphinxtrain run
Sphinxtrain path: C:/sphinx/sphinxtrain
Sphinxtrain binaries path: C:/sphinx/sphinxtrain/bin/Release/Win32
Running the training
MODULE: 000 Computing feature from audio files
Extracting features from  segments starting at  (part 1 of 1)
Extracting features from  segments starting at  (part 1 of 1)
Feature extraction is done
MODULE: 00 verify training files
    Phase 1: Checking to see if the dict and filler dict agrees with the phonelist file.
        Found 126 words using 38 phones
    Phase 2: Checking to make sure there are not duplicate entries in the dictionary
    Phase 3: Check general format for the fileids file; utterance length (must be positive); files exist
    Phase 4: Checking number of lines in the transcript file should match lines in fileids file
    Phase 5: Determine amount of training data, see if n_tied_states seems reasonable.
        Estimated Total Hours Training: 11.2335833333333
        Rule of thumb suggests 3000, however there is no correct answer
    Phase 6: Checking that all the words in the transcript are in the dictionary
        Words in dictionary: 123
        Words in filler dictionary: 3
```

Figure 4. 19 La vérification des fichiers de configuration

La sortie typique pendant le décodage ressemblera à voir Figure :

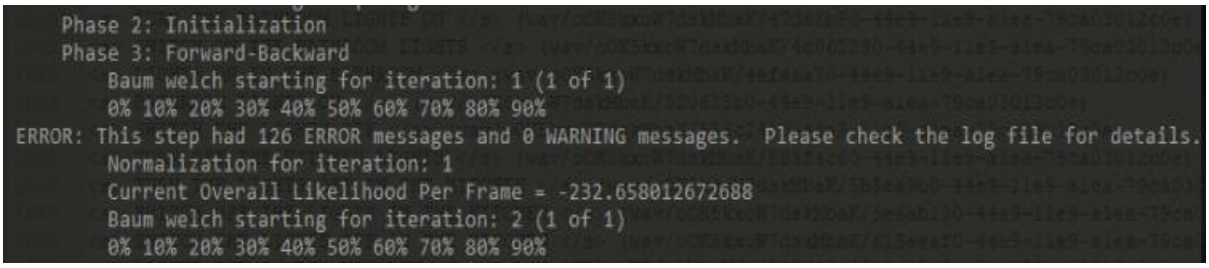

Figure 4. 20 La sortie typique pendant le décodage

Ces scripts traitent toutes les étapes nécessaires pour entraîner le modèle. Une fois qu'ils ont terminé, l'apprentissage est terminé.

Lorsque le travail de reconnaissance est terminé, le script calcule le taux d'erreur de mot de reconnaissance (WER) et le taux d'erreur de phrase (SER). Et pour notre modèle acoustique on a obtenu les résultats suivant voir Figure :

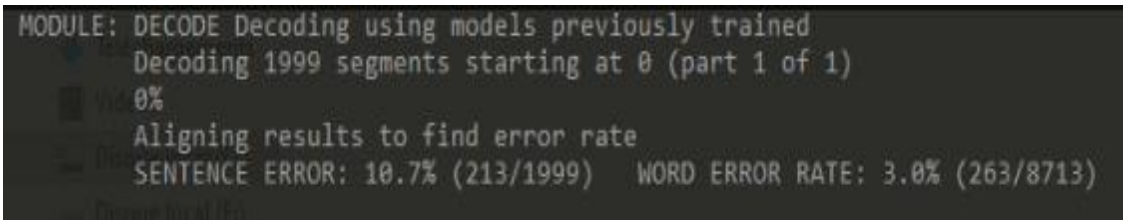

Figure 4. 21 Résultat

SER : 10.7% et WER : 3.0%

Pour utiliser le modèle acoustique entrainé il faut importer la bibliothèque pocketsphinx

Et utiliser les paramètres suivants voir Figure :

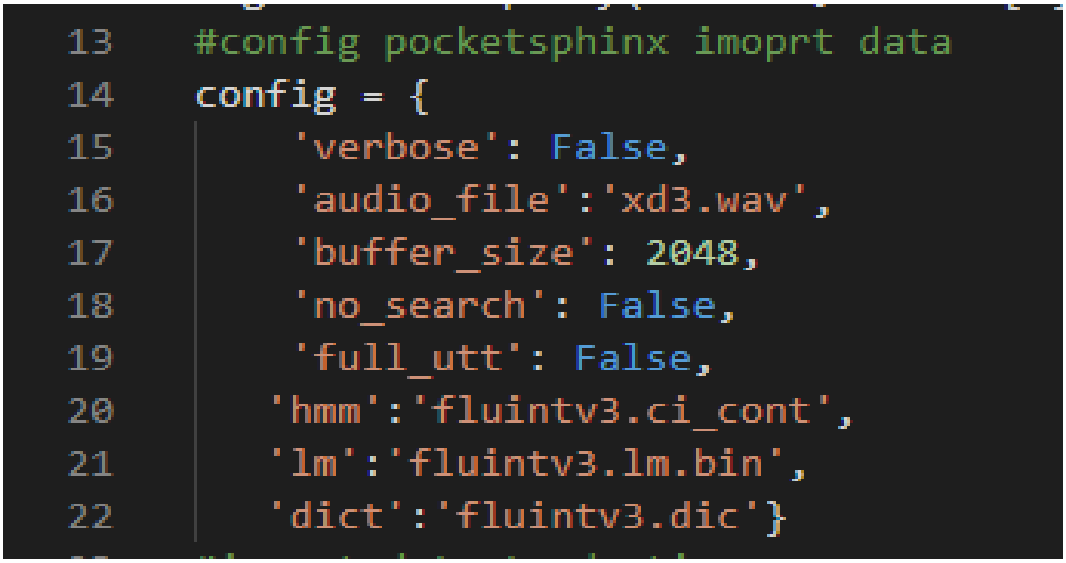

Figure 4. 22 Config pocketsphinx

5.2. Système de traduction automatique (RBMT)

Pour la traduction on a utilisé la traduction automatique basée sur des règles (RBMT) qui est un système basé sur des informations linguistiques sur les langues source et cible essentiellement extraites de dictionnaires et de grammaires, ce system RBMT (figure 4.23) se base sur une langue cible ,une langue source, dictionnaire bilingue grammaire, la syntaxe ,et la Sémantique

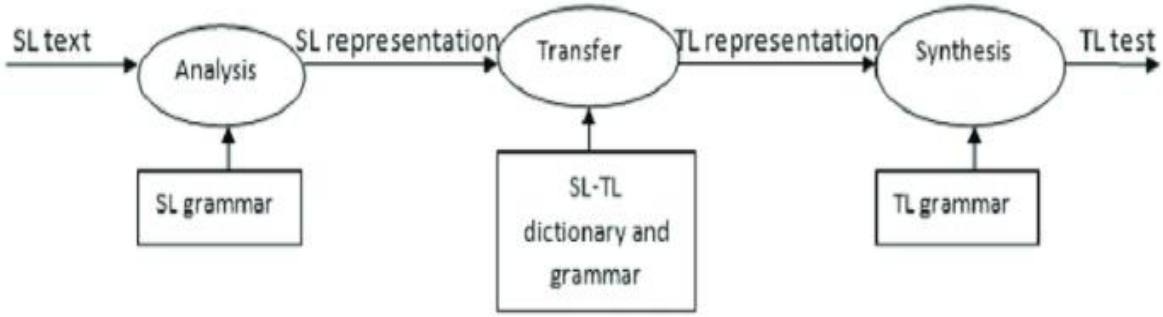

Figure 4. 23 schéma général de la traduction automatique basée sur des règles [10]

SL : langue source.

TL : langue cible

5.2.1 L'analyse syntaxique de la langue source

Cette analyse nécessite nettoyer le texte de la langue source tout d'abord en supprimant les ' : ', les ' ( ) ' et les '//'.

Ensuit on ajout un espace entre chaque mot voir Figure :

```
def addSpaces(str):
    str = re.sub(r"([\w/'+$\s-]+|[^\w/'+$\s-]+)\s*", r"\1 ", str)
    return str;
```

Figure 4. 24 l'ajout des espaces

Après on enlève la ponctuation voir Figure:

[10] English to Kurdish Rule-based Machine Translation System

```
def remove_punctuation(str):
 replace_punctuation = string.maketrans(string.punctuation, ' '*len(string.punctuation))
 text = str.translate(replace_punctuation)
 return text
```

Figure 4. 25 Suppression de la ponctuation

5.2.2 L'analyse sémantique de la langue source

Cette phase nécessite à modifier la forme du pluriel et la 3eme personne de singulier pour tous les mots voir Figure :

```
for i in range(len(ara)):
    engStr, araStr = clean(eng[i]), clean(ara[i])
    eng2ara[engStr] = araStr
    eng2ara[engStr+"s"] = araStr # plural forms of names and third person
                # a more accurate fix needs to be added to the data files
    ara2Eng[araStr] = engStr
```

Figure 4. 26 L'analyse sémantique

Après l'analyse on obtient une liste représentative de la langue source, avec l'utilisation de corpus de traduction, le dictionnaire et la grammaire des deux langues, la traduction faite mot par mot voir Figure.

```python
def translate(sentence, transdict):
    sent = addSpaces(sentence)

    words = [clean(word) for word in sent.split(" ")]

    i = 0
    while i < len(words)-2:
        if words[i] in adjs:
            current = words[i]
            temp = words[i+1]
            words[i] = temp
            words[i+1] = current
            i = i+2
        else:
            i = i+1
    #print(words)

    trans = [transdict[w] if w in transdict.keys() else w for w in words]

    print(' '.join(trans))
    return (' '.join(trans))
```

Figure 4. 27 Traduction

Et le résultat sera enregistrer et écrit pour qu'on puis l'utilise en 3ème étape.

5.3. Synthèse vocal

Dans cette étape on a utilisé la bibliothèque python « pyttsx3 » qui a comme rôle de convertir le texte en la parole et qui fonctionne même hors ligne.

5.3.1 L'installation de la bibliothèque :

L'installation de pyttsx3 et faite à travers le terminal cmder par la commande suivante :

```
pip install pyttsx3
```

5.3.2 L'utilisation de pyttsx3

Tout d'abord il faut importer la bibliothèque de cette manière :

```
import pyttsx3
```

Ensuite, on sélectionne le locuteur qui parle en arabe (id=1)

```
engine = pyttsx3.init()
voices = engine.getProperty('voices')
engine.setProperty('voice', voices[1].id)
```

Figure 4. 28 la sélection du locuteur

Après on utilise :

engine.say( )

engine.runAndWait( )

pour l'implémenter

5.4. Description de l'interface graphique Translator :

Cette partie contient des captures d'écran des différentes pages et interfaces de notre plateforme.

5.4.1 Accueil :

Dès qu'un utilisateur accède à l'application, il sera directement face à une présentation du site, qui par la suite l'invitera a commencé le processus de traduction en cliquant sur « UploadToTranslate », comme on peut le consulter dans la figure suivante :

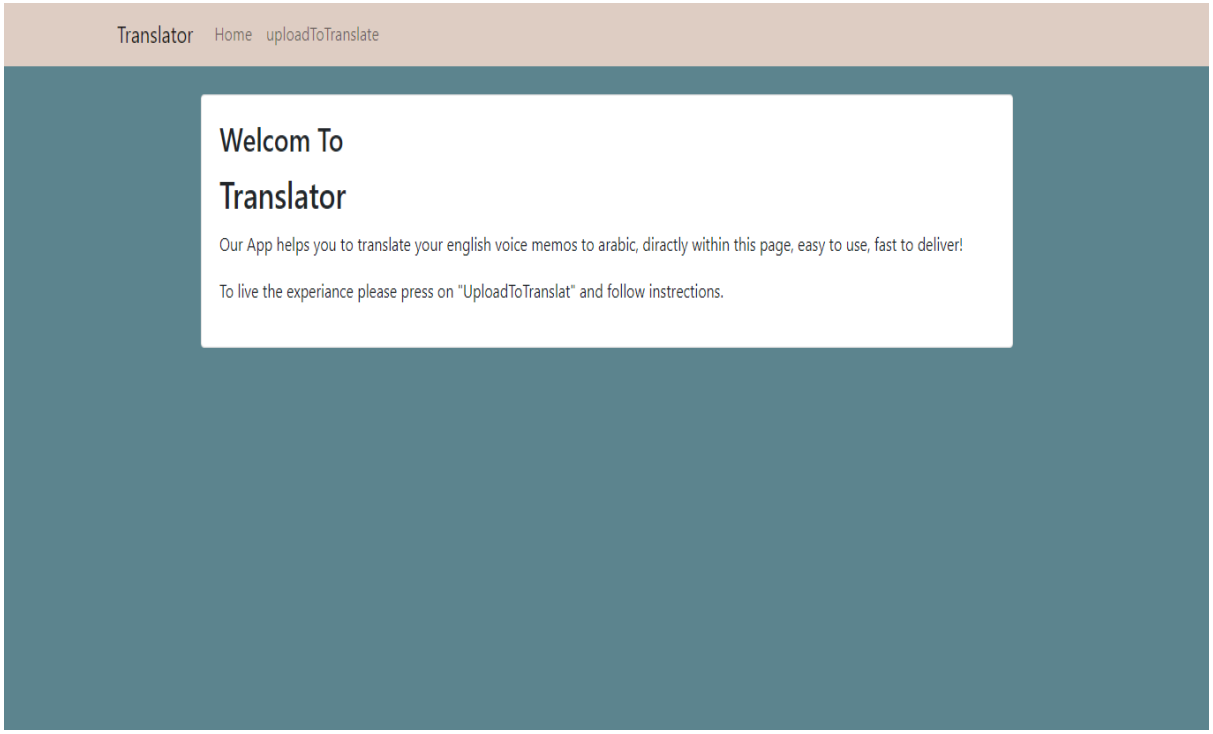

Figure 4. 29 Page d'accueil de l'application.

5.4.2 Traduction :

En cliquant sur « uptotranslate » l'utilisateur va consulter cette page qu'on peut la consulter dans la figure suivante :

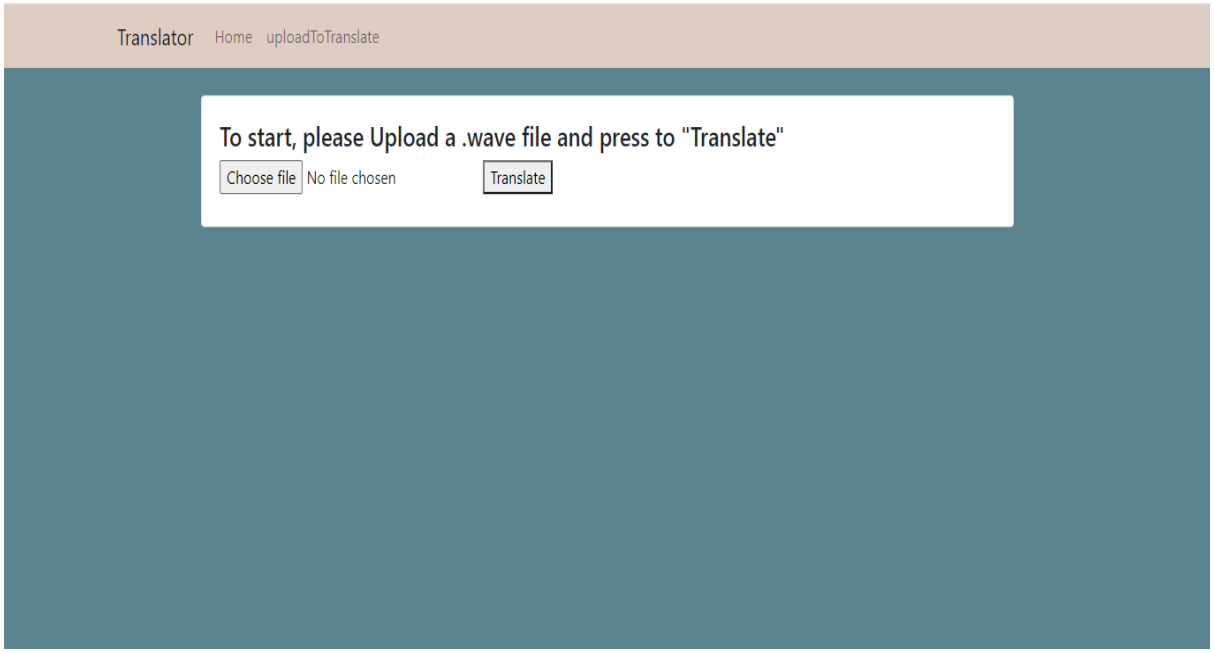

Figure 4. 30 page uploadtotranslate

Après, il doit importer son fichier .wav qu'il souhaite le traduire en arabe, en cliquant sur le bouton « Choose file ». Voir figure suivante :

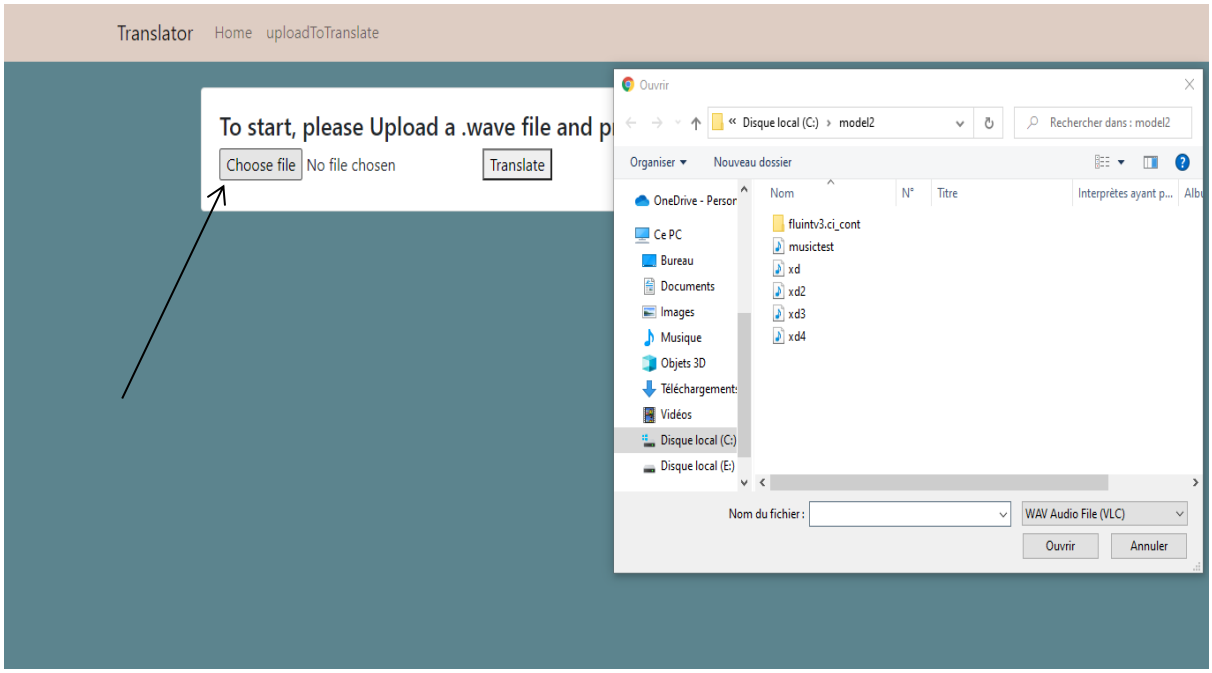

Figure 4. 31 upload file

Ensuit il doit cliquer sur le bouton « Translate » pour effectuer le processus de traduction : voir figure suivante :

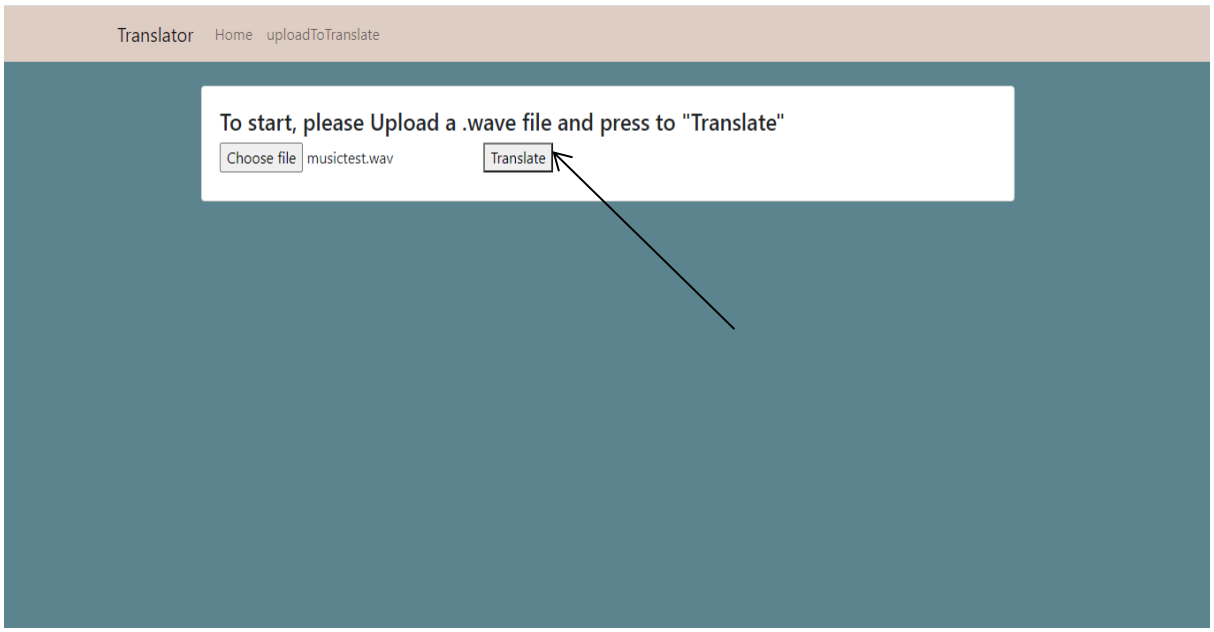

Figure 4. 32 translate

Le résultat va être affiché en dessous, et l'utilisateur peut écouter et télécharger le résultat en cliquant sur son lien (Figure 4.33) :

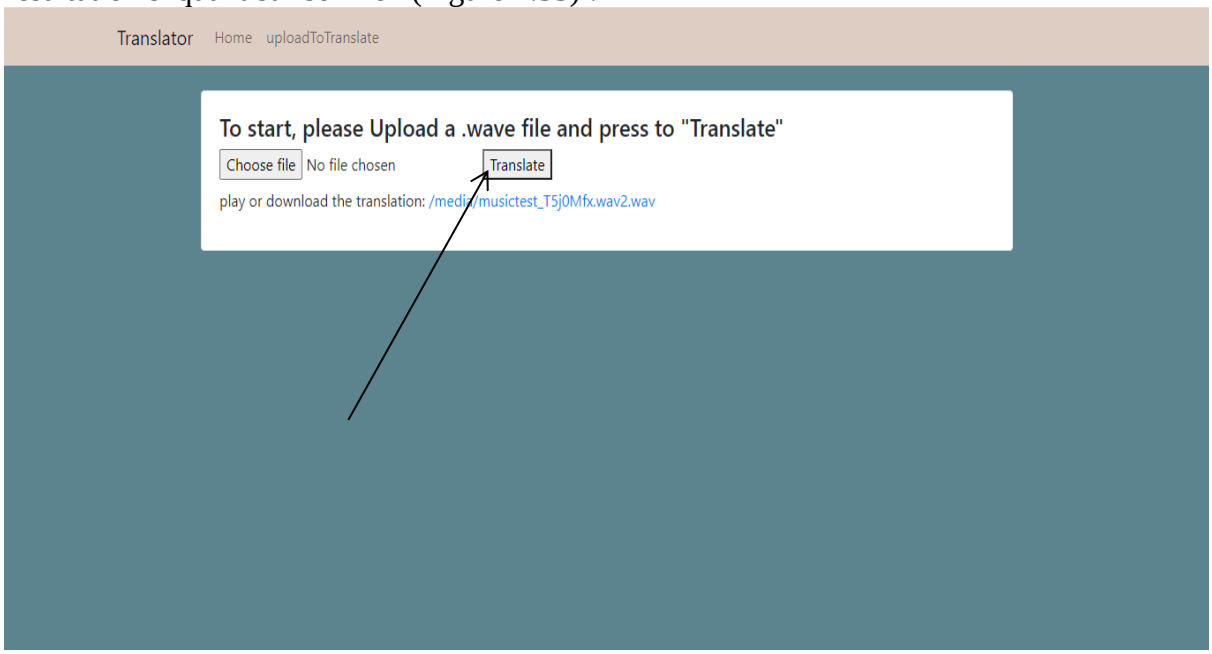

Figure 4. 33 résultat obtenue

Le lien lui dirigé vers une autre page (Figure 4.34), pour consulter le résultat tout on cliquant sur 1, et pour plus d'option s'il clique sur 2

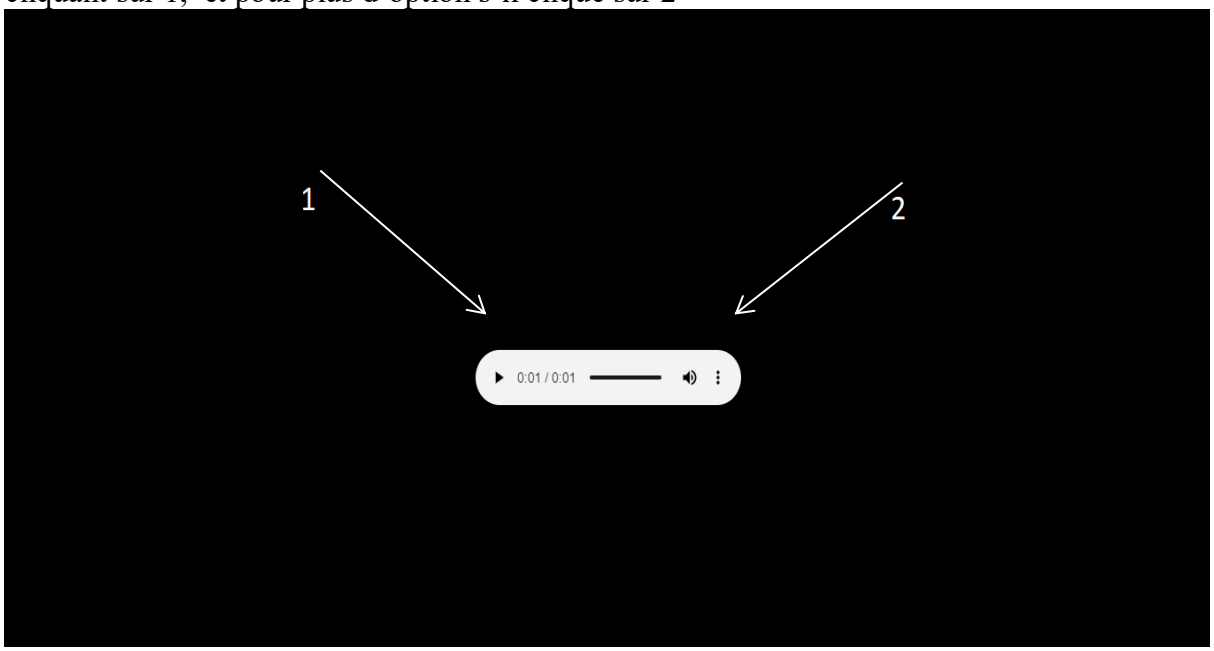

Figure 4. 34 lecture de résultat

En cliquant sur 2 l'utilisateur peut télécharger le résultat ou bien contrôler la vitesse de lecture :

Figure 4. 35 Option de téléchargement

Le résultat sera télécharger comme on peut le voir dans la figure suivant :

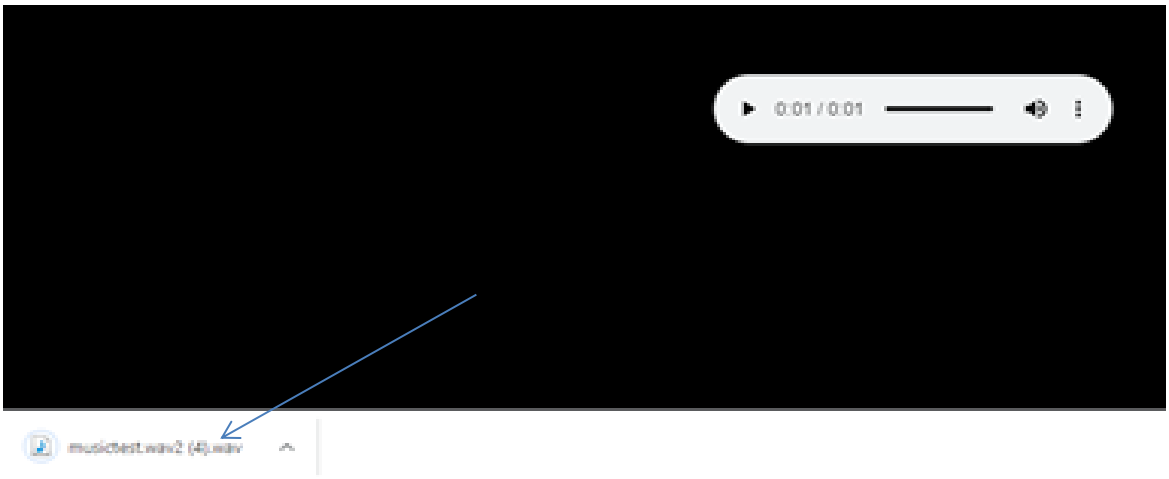

Figure 4. 36 le wav télécharger

5.5. Tests et comparaison

Après avoir implémenté l'application dans tous ses aspects présentés dans la conception, il est nécessaire de testé ces performances.

5.5.1 Pour la reconnaissance automatique :

Dans un premier temps, on a essayé de faire l'apprentissage avec le corpus globale de fluent mais comme elle contient trop d'enregistrements et notre équipement est faible, l'apprentissage a échoué, alors on a décidé de crée 03 corpus (fluentv1, fluentv2, fluentv3) en coupant le corpus globale, faire l'apprentissage avec chaque corpus va nous permette à faire l'évaluation des performances de notre système. Voici la description de chaque corpus :

	Nombre d'enregistrement	Nombre enregistrement d'entrainement	Nombre enregistrement de teste
Fluentv1	3500	2500	1000
Fluentv2	6911	5002	1909
Fluentv3	12000	10000	2000

Tableau 4. 2 La description des 03 corpus

Après l'apprentissage on a obtenu les résultats suivant :

	Fluentv1	Fluentv2	Fluentv3
WER	21.8%	21.1%	2.9%
SER	42.5%	42.8%	10.4%

Tableau 4. 3 Résultat d'apprentissage de chaque corpus

En analysant les résultats, nous concluons que plus le nombre d'enregistrement est élevé, le système sera plus performant.

Pour plus d'amélioration au niveau des performances du système de reconnaissance automatique de la parole, on a décide de faire des testes en modifiant deux (02) paramètres qui se trouvent dans le fichier sphinx_traine.cfg, qui sont :

- MFCC : Mel-Frequency Cepstral Coefficients sont des coefficients cepstraux calculés par une transformée en cosinus discrète appliquée au spectre de puissance d'un signal. Les bandes de fréquence de ce spectre sont espacées logarithmiquement selon l'échelle de Mel.
- GMM : Modèles de mélange gaussien

Les modifications sont faites dans la ligne 31 pour MFCC et la ligne 144 pour GMM.

Cela peut être consulté dans les figures 43 et 44.

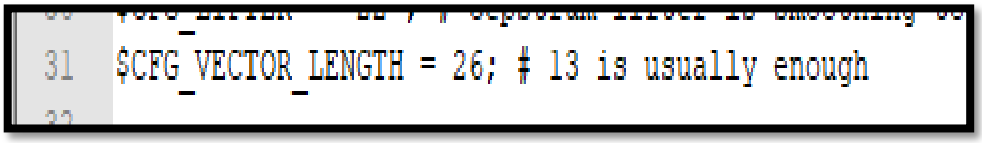

Figure 4. 37 L'emplacement des modifications MFCC

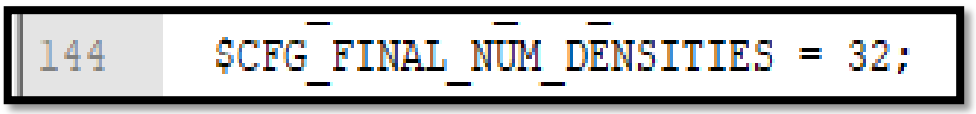

Figure 4. 38 L'emplacement des modifications GMM

Pour chaque corpus les testes étaient fait 09 fois. Les résultats obtenus seront détaillés dans les tableaux suivants :

Pour fluentv1 :

WER (%)			
GMM(144) / MFCC (31)	08	16	32
13	**21.89**	**19.4**	**21.2**
26	**27.4**	**27.6**	**30.6**
39	**38.7**	**erreur**	**erreur**

Tableau 4. 4 WER fluentv1

SER (%)			
GMM(144) / MFCC (31)	08	16	32
13	**42.5**	**39.1**	**41.9**
26	**47.9**	**49.6**	**54.5**
39	**60.4**	**erreur**	**erreur**

Tableau 4. 5 SER fluentv1

Pour fluentv2

WER (%)			
GMM(144) / MFCC (31)	08	16	32
13	**21.4**	**25.1**	**34.5**
26	**38.4**	**50.2**	**53.9**
39	**39**	**98.2**	**erreur**

Tableau 4. 6 WER fluentv2

SER (%)			
GMM(144) / MFCC (31)	08	16	32
13	**42.8**	**49.5**	**65.5**
26	**62.8**	**77**	**79.3**
39	**100**	**100**	**erreur**

Tableau 4. 7 SER fluentv2

Pour fluentv3 :

WER (%)

GMM(144) / MFCC (31)	08	16	32
13	**2.9**	**6.5**	**27.8**
26	**3.0**	**3.0**	**3.4**
39	**6.0**	**94.4**	**erreur**

Tableau 4. 8 WER fluentv3

SER (%)			
GMM(144) / MFCC (31)	08	16	32
13	**10.4**	**13.6**	**26.8**
26	**10.7**	**10.6**	**12.0**
39	**18.7**	**98.8**	**erreur**

Tableau 4. 9 SER fluentv3

L'erreur est produite puisque notre équipement est faible.

En analysant les résultats, on a remarqué que les modifications et les testes qui en étaient fait n'apporte aucune amélioration a notre système et nous concluons que les paramètres par défaut qui sont « MFCC=13 » et « GMM =08 » étaient les bons pour notre système.

Afin d'implémenter le système de reconnaissance automatique de la parole le plus performant on a décidé de faire l'apprentissage avec le corpus « fluentv3 » avec les paramètres MFCC et GMM par défaut.

5.5.2 Pour de la traduction automatique

Pour la traduction automatique nous avons décidé d'implémenter deux (02) modèles de traduction différents sur notre corpus de traduction.

Premièrement on a implémenté le système de traduction basé sur des règles (RBMT) et les scores obtenu sont montré dans la figure suivante :

```
++++++++++++++++Number Of Sentences Of++++++++++++++++++++

Source Text:  220
Target Text 220
Test Text 28
Ref Text 28
Pred Text 28

 ++++++++++++++++++Corpus Bleu++++++++++++++++++++++++++++++

CorpBleu:  BLEU = 65.3, 91/5/0/0/0 (BP=1.0, ratio=1.0, hyp_len=33, ref_len=33)
```

Figure 4. 39 Score RBMT

Ensuite on a implémenté le deuxième système, la traduction automatique classique (SMT), et cette fois ci on a obtenu le score qui va apparaitre dans la figure suivante :

```
++++++++++++++++Number Of Sentences Of++++++++++++++++++++

Test Text 28
Ref Text 28
Pred Text 28

 ++++++++++++++++++Corpus Bleu++++++++++++++++++++++++++++++

CorpBleu:  BLEU = 28.1, 103/5/3/0/0 (BP=1.0, ratio=1.03, hyp_len=34, ref_len=33)
```

Figure 4. 40 Score SMT

En analysant les scores, on a décidé d'utiliser et implémenté le RBMT pour notre projet puisque il a été le plus performant avec un BLEU de 65.3%.

Si on a pu obtenir ce score là c'est par ce que le RBMT soutient bien les petit corpus, mais si le corpus était plus grand ou bien si on avait un corpus massive, il sera mieux d'utilisé le SMT, bien sûr à l'aide d'un équipement puissant.

6. Conclusion

Dans ce chapitre on a présenté l'interface de notre plateforme « Translator », les bibliothèques et les corpus utilisés dans notre projet. Pour finir on a présenté et analysé quelques testes sur notre système pour bien l'évaluer.

Conclusion générale

Conclusion générale

Dans le cadre de ce travail et au cœur de ce mémoire, nous avons traité le problème de la Traduction Automatique de la Parole (TAP) offline de l'anglais vers l'Arabe. Aux début nous avons fait un tour d'horizon sur la TAP, ensuite on a présenté les détails de notre plateforme tel qu'on a détaillé le système de reconnaissance basé sur les HMM avec le modèle de traduction automatique basé sur des règles et on a implémenté le système de la synthèse vocal à l'aide de l'utilisation de la bibliothèque pytts3 de python, concernant l'apprentissage on a utilisé la base de données fluentv3 qui nous a pris sept (07) heures d'entrainement qui a donné des résultats très acceptable par rapport à notre équipement utilisé, notre contribution clé est l'utilisation de la langue Arabe et l'implémenter d'un système offline avec un résultat de 10.4 Sentence Errer Rate et 2.9 Word errer rate

Perceptives

Afin d'améliorer notre travaille, nous envisageons quelques perspectives pour des futures recherches qui sont :

- Réaliser le processus inverse c'est-à-dire la traduction automatique de la parole de l'Arabe vers l'Anglais.
- L'intégration de dialecte Arabe pour répondre aux besoins de notre société.
- L'utilisation d'un corpus beaucoup plus grand.
- L'utilisation d'autres modèles de reconnaissance automatique qui sont basé sur les réseaux de neurones par exemple.
- L'implémentation de l'approche hybride dans la traduction automatique.

Référence

Références

[1] Constance-Louise Gauriau , Benoît Prieur, “Introduction au TALN (Traitement Automatique du Langage Naturel) avec spaCy,Constance-Louise Gauriau , Benoît Prieur,2021-01,page75-79”.

[2] Elizabeth D. Liddy, *Natural Language Processing Natural Language Processing,Elizabeth D. Liddy,p2*.

[3] *Natural language processing in Action,Hobson Lane Cole Howard Hannes Max Hapke,p4*.

[4] *Les principaux paramètres du signal vocal,Droua-Hamdani G.*

[5] *Reconnaissance automatique de la parole : tout commence par la voix Le 06/02/2019 (Mis à jour le 07/02/2019) Pierre Ponlevé*.

[6] *intelligenceartificielle.com*.

[7] *Exemple de chaîne de Markov avec le Modèle du Sac en Papier Auteur : Emmanuel Manproc Prochasson Le 29/06/2006*.

[8] *(Forney, 1973) J. Forney, G.D., 1973. The viterbi algorithm. Proceedings of the IEEE 61(3), 268–278. (Baum et al., 1966) L. Baum, T. Petrie, G. Soules, et N. Weiss, 1966. Statistical*.

[9] *inference for probabilistic functions of finite state markov chains. Annals of Mathematical Statistics 37, 1554–1563.*

[10] *(Rabiner, 1989) L. Rabiner, 1989. A tutorial on hidden markov models and selected applications inspeech recognition. Proceedings of the IEEE 77(2), 257–286.*

[11] *(Rabiner et Juang, 1993) L. Rabiner et B.-H. Juang, 1993. Fundamentals of speech recognition. Upper Saddle River, NJ, USA : Prentice-Hall, Inc.*

[12] *(Russell, 1993) M. Russell, 1993. A segmental hmm for speech pattern modelling. Dans les actes de IEEE International Conference on Acoustics, Speech and Language Processing, Volume II, Minneapolis, MN, USA, 499–502.*

[13] *(Gong et Haton, 1994) Y. Gong et J.-P. Haton, 1994. Stochastic trajectory modeling for speech recognition. Dans les actes de IEEE International Conference on Acoustics, Speech and Language Processing, Volume I, Adelaide, SA, Australia, 57–60.*

[14] *(Mari et al., 1996) J.-F. Mari, D. Fohr, et J.-C. Junqua, 1996. A second-order hmm for high performance word and phoneme-based continuous speech recognition. Dans*

les actes de IEEE International Conference on Acoustics, Speech and Language Processing, Washington, DC, USA, 435–438.

[15] "L. Personnaz et I. Rivals, 2003. Réseaux de neurones formels pour la mod'elisation, la commande et la classification. Lavoisier.,"

[16] *R. Gemello, D. Albesano, et F. Mana, 1997. Continuous speech recognition with neural networks and stationary-transitional acoustic units. Dans les actes de IEEE International Conference on Neural Networks, Houston, USA, 2107–2111.*

[17] *Réseau de neurones formels de type Perceptron Multicouche,HRcommons ,14 août 2009.*

[18] *Antoine Cornuéjols, Laurent Miclet, Yves Kodratoff, Apprentissage Artificiel : Concepts et algorithmes, Eyrolles, 2002 (ISBN 2-212-11020-0) [détail des éditions]*.

[19] L. Barrault, "Diagnostic pour la combinaison de systèmes de reconnaissance automatique de la parole.," p. 184, 2009.

[20] *Diagnostic pour la combinaison de systèmes de reconnaissance automatique de la parole,Thése de doctora par Loïc BARRAULT,p26-31*.

[21] *https://digitiz.fr/blog/logiciels-reconnaissance-dictee-vocale/*.

[22] "Text-to-Speech Synthesis || Introduction Taylor, Paul;2009,p1".

[23] *article de Christian Coudert et Jean-Pierre Carpanini du Centre d'Évaluation et de Recherche sur les Technologies pour Aveugles et Malvoyants de l'Association* Valentin *Haüy*.

[24] *van Bezooijen et van Heuven, 1997, p. 481*.

[25] *Diagramma di flusso di sintesi vocale Fonte Disegno proprio Data 3 marzo 2009 Autore Utente:Grigio60 Licenza d'uso (riusare il file) Pubblico dominio*.

[26] *(FR) P. H. Van Santen, Richard William Sproat, Joseph P. Olive, et Julia Hirschberg, Les progrès dans la synthèse vocale. Springer: 1997.*

[27] *https://boowiki.info/art/synthese-vocale/synthese-vocale*.

[28] *Sur la base de PSOLA*.

[29] *T. Dutoit, V. Pagel, N. Pierret, F. Bataiile, O. van der Vrecken. Le projet MBROLA: Vers un ensemble de synthétiseurs vocaux de haute qualité d'utilisation à des fins non commerciales. ICSLP Proceedings, 1996.*

[30] *J. V. Ralston, et al., "Comprehension of synthetic speech produced by rule," Speech Research Laboratory, Indiana University, Bloomington, IN47405, Research on Speech Perception Progress Report 15, 1989.*

[31] *M. Goldstein, "Classification of methods used for assessment of text-to-speech systems according to the demands placed on the listener," Speech Communication, vol. 16, pp. 225-244, 1995.*

[32] *H. C. Nusbaum, et al., "Subjective evaluation of synthetic speech: Measuring preference, naturalness, and acceptability," Speech Research Laboratory, Indiana University, Bloomington, IN47405, Research on Speech Perception Progress Report 10, 1984.*

[33] *H. Nusbaum, et al., "Measuring the naturalness of synthetic speech," International Journal of Speech Technology, vol. 1, pp. 7-19, 1995.*

[34] *J. Terken and G. Lemeer, "Effects of segmental quality and intonation on quality judgments for texts and utterances," Journal of Phonetics, vol. 16, pp. 453-457, 1988.*

[35] *C. R. Paris, et al., "Linguistic cues and memory for synthetic and natural speech," Human Factors, vol. 42, pp. 421-431, 2000.*

[36] *https://fr.myservername.com/top-11-best-text-speech-software.*

[37] Kaji H, An efficient execution method for rule-based machine translation (1988). Available at: http://www.aclweb.org/anthology/C/C88/C88-2167.pdf (Accessed on 25th April 2010).

[38] LA TRADUCTION AUTOMATIQUE,Frédérique LAB.LLETIN DE L'EPI,page 170.

[39] Ali N, Machine translation: a contrastive linguistic perspective. Available at: http://www.unesco.org/comnat/france/ali.htm (Accessed on 25th April 2010).

[40] Flexible finite-state lexical selection for rule-based machine translation.Francis Tyers University of Alicante Felipe Sánchez-Martínez University of Alicante.

[41] Krauwer S, The Eurotra Project (2008). Available at: http://www-sk.let.uu.nl/stt/eurotra.html (Accessed on 25th April 2010).

[42] DeNeefe S, Knight K, Wang W and Marcu, D, What can syntax-based MT learn from phrase-based MT? Available at: http://www.isi.edu/natural-language/mt/ats-vs-ghkm.pdf (Accessed on 25th April 2010).

[43] Evolutive translation models,Fr´ed´eric Blain,HAL Id: tel-01142926.

[44] "International Journal of Computer Applications (0975 – 8887) Volume 125 – No.7, September 2015."

[45] Ulrich Germann. Greedy Decoding for Statistical Machine Translation in Almost Linear Time. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1, NAACL '03, pages 1{8, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics.

[46] K. Papineni, S. Roukos, T. Ward, and W. Zhu, "Bleu: a Method for Automatic Evaluation of Machine Translation", Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, pp. 311-318, (2002).

[47] M. Popovic and H. Ney." Word error rates: Decomposition over POS classes and applications for error analysis". In Proceedings of ACL Workshop on Machine Translation.

[48] C. Tillman, , S. Vogel, H. Ney, H. Sawaf, and A. Zubiaga. :"Accelerated DP-based search for statistical translation". In Proceedings of the 5th European Conference on Speech Communication and Technology, pp- 2667.2670. Rhodes, Greece. (1997).

[49] M. Snover, B. Dorr, , R. Schwartz, L. Micciulla, J. Makhoul.: "A Study of Translation Edit Rate with Targeted Human Annotation". In Proceedings of AMTA, Boston, (2006).

[50] "Google Translate Community.Accessed: 2016-12-05.".