University Paris-Est Creteil, Creteil, France

Master 1 OIVM Optique Image Vision MultiMedia

Reconnaissance vocale en temps réel

Auteur:
Iskander Djouad et Nassim Battache

Superviseur :
Amir Nakib

January 10, 2022

Abstract

La reconnaissance vocale consiste en l'analyse de la voix humaine, afin de la transformer en texte. Tout passe par la voix, qui est identifiée puis captée en fréquences sonores (Speech-to-text). Vient ensuite l'analyse de ces fichiers sonores, par les technologies du deep learning liées à l'intelligence artificielle [1] .

Dans cette approche nous allons spécifier les différentes étapes nécessaire a fin de mettre en place un algorithme capable de faire La reconnaissance vocale a un temps presque réel en mode online a l'aide de l'API Google tout en codent en python on utilisent PyAudio, SpeechRecognition et Pyttsx3 qui sont des bibliothèques python .

Contents

Introduction **1**

1 Installation des bibiotehques : **2**
1.0.1 Pour Windows : . . . 2
1.0.2 Pour Linux/MacOs : . . . 2
1.0.3 Pourquoi ces librairies ? . . . 2

2 Speech to Text (STT) **3**
2.1 élément déclencheur . . . 3
2.2 Identifier et transcrire la voix en texte . . . 3
2.3 Google speech API . . . 3

3 Text to Speech (TTS) **5**

4 Méthode et résultat : **6**
4.1 Tools . . . 6
4.2 Results . . . 7

Conclusion **8**

perspective **9**

Introduction

La reconnaissance vocal ou comme souvent appeler la reconnaissance automatique de la parole est une technologie biométrique qui a pour but de transcrire un son pourvu d'un microphone ou d'un fichier audio et de le transcrire en texte ont passent par différentes étapes de traitement .

En effet c'est un domaine très complexe , W. Minker et S. Bennacef explique dans "Parole et dialogue homme-machine" qu'il existe une différence importante entre le langage formel, qui est compris et utilisé par les machines, et le langage naturel, que les humains utilisent. Le langage formel est structuré par des règles syntaxiques strictes et sans ambigüité. À l'inverse, dans le langage naturel, des mots ou des phrases peuvent avoir plusieurs sens selon l'intonation de l'énonciateur ou le contexte par exemple [2] .

Ce projet a pour but de crée un algorithme qui traite le fonctionnement de la reconnaissance vocale , on partons de la thèse qui dit que la reconnaissance vocale correspond à un cycle d'utilisation complet de la voix nous allons détailler chronologiquement les fonction utiliser , depuis le moment où l'individu prend la parole (éliment déclencher) jusqu'à la condition d'arrêt .

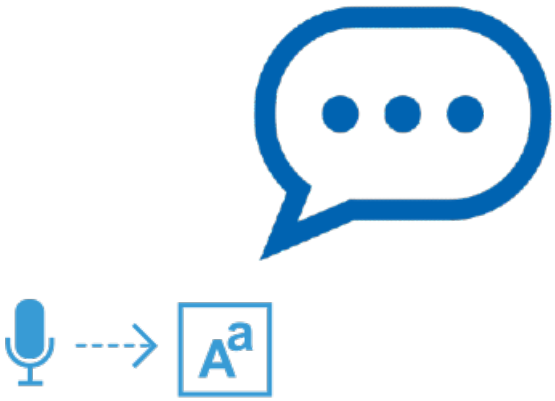

Figure 1: reconnaissance vocale shéma [4]

Chapter 1

Installation des bibiotehques :

afin de commencer la programmation il est nécessaire d'installer les librairies suivantes :

- PyAudio
- SpeechRecognition
- Pyttsx3

Ces deux modules ne sont pas intégrés à python donc il est nécessaire de les intégrer garce aux deux commandes ci-dessous selon le système d'exploitation :

1.0.1 Pour Windows :

```
pip install pyaudio
pip install speechRecognition
```

```
pip installer pyttsx3
```

Figure 1.1: Instruction pour instaler les modules

Remarque : parfois l'installation de pyaudio s'avers assez compliquer pour une version de python supérieur a 3.6. par conséquent l'installation classique ne fonctionne pas en effet il faut donc passer par le terminal de Windows « cmd » et exécuter les deux commandes suivantes :

```
pip install pipwin
pipwin install pyaudio
```

Figure 1.2: CMD

1.0.2 Pour Linux/MacOs :

```
sudo apt-get install python-pyaudio python3-pyaudio
pip install SpeechRecognition
```

Figure 1.3: CMD

1.0.3 Pourquoi ces librairies ?

Pyaudio est la librairie qui permet à python de communiquer avec les ports audios de la machine. Cette librairie permet de jouer et d'enregistrer de l'audio très facilement.

speechRecognition est la librairie qui aide à la reconnaissance vocale. La reconnaissance vocale est le processus de conversion audio en texte.

Pyttsx3 est utilisé pour la conversion du texte en parole.

Chapter 2

Speech to Text (STT)

Le Speech to Text est une fonction qui consiste a identifier la voix et la distangué d'un bruit lamda. L' algorithme permet d'améliorer au mieux le rendu sonore, en supprimant les bruits extérieurs. Ils segmentent également le texte en fractions pour séparer les mots entre eux. Le texte est ensuite analysé par la technologie du deep learning [3]

2.1 élément déclencheur

La première étape dans notre approche est d'initie un élément déclencheur, cette élément peut bien être un fichier audio ou bien même le microphone de l'ordinateur, il consiste a réveiller le système .

Afin de commencer le traitement , dans notre cas l'élément déclencheur est le microphone de l'ordinateur il est considéré comme source de la prévenance de la parole .

2.2 Identifier et transcrire la voix en texte

Une fois la reconnaissance vocale se déclenche , il est nécessaire d'exploiter la voix. Pour cela, il est primordial de l'enregistrer et de la numériser .

On utilise la bibliothèque python speechRecognition Durant cette étape, la voix est captée en fréquences par la méthode Recognize sonores pouvant être exploitées par la suite . ‘

Dés que la fréquences sonore franchie le seuil il est considère comme de la parole et non pas un bruite , on parle alors de filtrage du son de la source .

2.3 Google speech API

afin de réaliser une reconnaissance vocale on doit choisir un services vocale qui fournisse des bibliothèques dans le langage de programmation python , voici quelque unes : Google Speech-to-Text, Discours Microsoft Azure ,IBM Watson Speech to Test ,Amazon Transcribe ,Nuancer , CMU Sphinx ,Mozilla DeepSpeech ,Kaldi ,Wav2letter Facebook . . .

Dans notre cas on utilise Google Speech API .

Avant d'utiliser Google speech API il est nécessaire de vérifier notre connexion a l'internet pour cela on utilise la bibliothèque urllib a fin d'ouvrir le navigateur google en cas de connexion a internet .

En suite a l'aide de la méthode RecognizeGoogle() en mettons le signal sonore enregester comme entrée ainsi que la langue sible dans notre cas on n'a choisi le français . ‘

Cette méthode permet d'analyser la phrase et d'extraire le plus de données linguistiques. Elle commence par associer des tags aux mots de la phrase, c'est ce qu'on appelle la tokenisation [6]. Ce sont en réalité des “étiquettes” que l'on appose sur chaque mot afin de les caractériser .

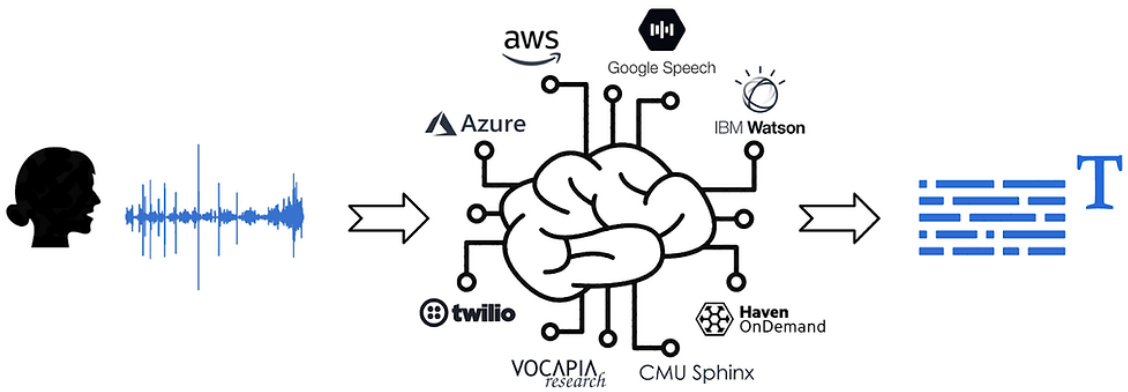

Figure 2.1: STT

Chapter 3

Text to Speech (TTS)

Cette étape est primordiale dans le cas du développement d'un assistent Virtual , d'une parte elle nous permet de vérifier le bon fonctionnement de STT on réalisent un jeu d'essai , ou bien l'associer l'un des serveur vocaux pour on suite par exemple demander l'heure ou bien une recherche sur Wikipédia ou YouTube ..

Pour cela il est préférable d'utiliser la bibliothèque python pyttsx3 pour traduire un texte donnée ou une commande on vocaux

pyttsx3.say() permet de transcription un texte a un vocale et pyttsx3.stop() permet d'arrêté la transcription tandis que runWait() annonce un fin de cycle de transcription

Les principales fonctionnalités de cette bibliothèque sont :

- Le choix entre les voix installées sur le système est possible
- Ajuster le volume
- Choisir la vitesse des paroles
- Possibilités de l'enregistrement de l'audio dans un fichier
- La connexion internet n'est pas requise

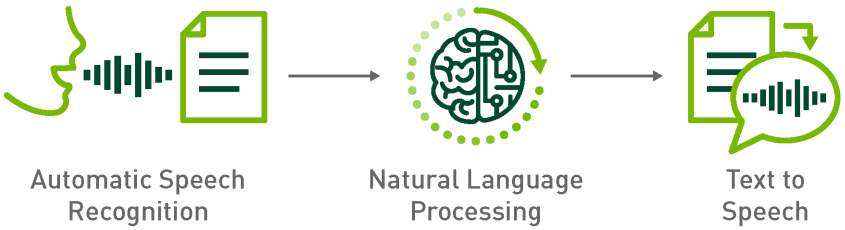

Figure 3.1: TTS

Chapter 4

Méthode et résultat :

Dans cette section on va presenter les differentes methodes utilisées et les resultats obtenus.

4.1 Tools

Avant de commencer l'algorithme il est préférable de vérifier notre connexion a l'internet car on va par la suite utiliser Google speech api qui est un module qui requière la connexion a internet a fin d'envoyer le cycle sonore aux serveur vocale de google et de recevoir sa transposition .

En suite on initialise les modules speechRecognition et Pyttsx3 et a l'aide du module on initialise le microphone de l'ordinateur comme source . On initialise aussi une variable audio qui permet d'enregistrer la fréquence sonore .

On utilise sur cette dernière la méthode RecognizGoogle() pour envoyer l'enregistrement aux servers google pour pouvoir le transcrire en texte . Si la transcription et égale a une phrase définie alors le serveur vocal réponds comme définie , sinon il réécrie le texte de transaction sur le terminal .

```python
import speech_recognition as sr
import pyttsx3
import re
import pdb
import urllib
from urllib.request import  urlopen
import datetime
pret = True
def ConnectToInternet ():
    try :
        urlopen('https://www.google.com',timeout=1)
        return True
    except urllib.error.URLError :

        return False

def SpeechToTeext ( ):
 global pret
 engine = pyttsx3.init()
 rec = sr.Recognizer()
 mic = sr.Microphone()

 with mic as source:

            rec.adjust_for_ambient_noise(source)
            print ("SVP commancer a parler apres une seconde ")


            while True :
```

```python
while True :

        audio=rec.listen(source)
        if ConnectToInternet () :

         try :

            if rec.recognize_google(audio, language="fr-FR")=="Bonjou
              print("bonjour comment aller vous   " )
              engine.say("bonjour comment aller vous" )
              engine.runAndWait()
            elif rec.recognize_google(audio, language="fr-FR")=="qui
              Bot = "je suis votre assistant vertueil cree par les é
              print(Bot )
              engine.say(Bot)
              engine.runAndWait()
            elif rec.recognize_google(audio, language="fr-FR")=="au r
              Bot = "a bien tot merci "
              print(Bot )
              engine.say(Bot)
              engine.runAndWait()
              engine.stop()
            else :
                you = rec.recognize_google(audio, language="fr-FR")
                print(you )
                engine.say(you)
                engine.runAndWait()

         except :
                 rec.recognize_google(audio, language="fr-FR")=="" or
                 print ("j'ai pas bien comprie ")
                 engine.say("j'ai pas bien comprie ")
                 engine.runAndWait()
        else:
         pass
```

Figure 4.1: Algorithme de reconnaissance vocale en python

La figure 4 illustre la fonction principale du code mais avant de l'exécuter il faudrait vérifier la connexion a l'internet

4.2 Results

Après avoir installer tous les modules décrits dans le chapitre 1 , et vérifier notre connexion internet , on a exécuter le code illustre dans la figure 4 , et on a obtenu le résultat suivent :

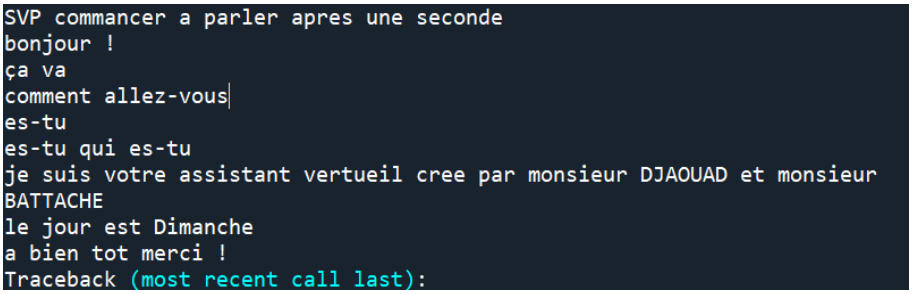

Figure 4.2: Spyder karnel

La figure 4 nous illustre le résultat de l'exécution de notre algorithme , on remarque que notre algorithme met un peut de temps avant de commencer à transcrire , ce temps est estimé a une seconde prêt ,car comme on sait python est un langage de programmation orienté objet de haut niveau , les programmes écrits avec ce langages on un temps d'exécution car ils prennent un temps un peu plus important que le C++ qui est lui aussi un langage de programmation orienté objet mais plus proche du langage machine que python .

après une seconde de l'exécution de notre programme on peut prendre la parole est on voit que notre semi-assistant se met à nous répondre en un temps presque réel , car on effet , pour pouvoir envoyer l'audio aux servers de google , est réceptionner le résultat la machine prend du temps selon le débit d'internet et aussi la capacité de la machine elle-même .

Conclusion

Ce projet explique et démontre les diffirentes étapes de la construction d'un algorithme de reconnaissance vocale en temps réel avec python comme langage de programmation et Google speech to text comme server vocal

Nous avons pue constater qu'il est extremement difficile de faire de la reconnaicence vocal en temps réel on utillison python comme langage de programmation et un server vocal externe (google stt) .

Le fonctionnement de la reconnaissance vocale se base sur la complémentarité entre plusieurs technologies issues du même domaine .

Perspective

Pour nos perspective on peut ajouter des fonctionnalité a notre assistent vocale pour qu'il puisse réaliser des tache plus rudes (rechercher sur google , Wikipédia , YouTube ..)

L'associer avec d'autre technologie biométrique tel comme la reconnaissance facial ou bien de la vision par ordinateur (création d'avatar) a fin de crée un assistent Virtuel .

Développe notre propre logiciel de synthèse vocal a fin de faire du temps réel ,

Bibliography

[1] Reconnaissance automatique de la parole. https://fr.wikipedia.org/wiki/Reconnaissance$_a utomatique_d e_l a_p arole$.

[2] Reconnaissance vocale : Fonctionnement et composition. https://vivoka.com/fr/reconnaissance-vocale-fonctionnement-composition/.

[3] Solutions de synthèse vocale à usage personnel et professionnel. https://geekflare.com/fr/speech-to-text-solutions/.

[4] Anthony Zhang. reconnaissance de la parole. https://github.com/Uberi/speech$_r ecognition/blob/c$89856088ad81d81d38be314e