Procesoare de semnal

Recunoașterea vorbirii folosind Python

Student:

Florin Constantin Asavei

Grupa: 2241

Profesor indrumător:

Prof. Dr. Ing. Eugen Lupu

Cuprins:

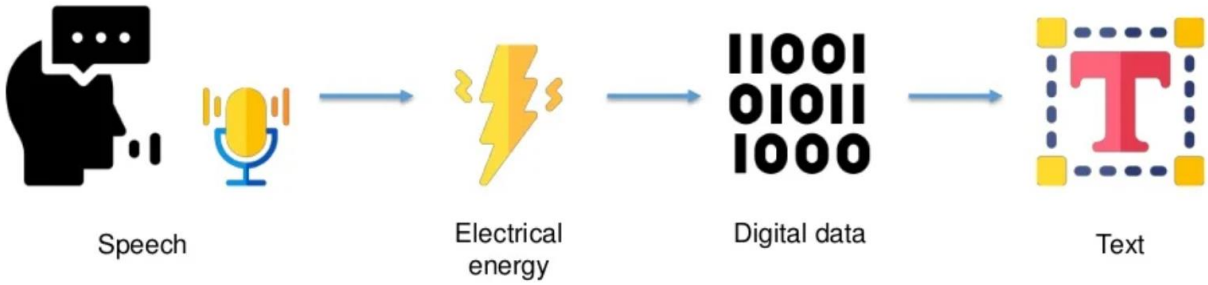
1.Introducere

Recunoașterea vorbirii ne ajută să „vorbim” computerelor și să le dăm instrucțiuni. Recunoașterea vorbirii este folosit la asistenții virtuali precum Siri de la Apple, Alexa de la Amazon, chiar și la telefoanele noastre care ne permite să căutăm informații rapide pe web.

Recunoașterea vorbirii încorporează câmpul calculatorului și ajută să identifice cuvintele vorbite și convertirea lor în texte. Asta permite computerelor să înțeleagă limbajul uman.

2.Specificația problemei

Vorbirea este convertită prima data de la sunetul fizic la energia electrică folosind un microfon și apoi către datele digitale folosind un analog la convertorul digital. Datele digitale pot fi convertite în text utilizând algoritmi precum Rețele Neuronale (Neural Networks) sau Modele Markov ascunse (Hidden Markov Models).



3.Date

Vom instala următoarele librării cu ajutorul comenzii cmd.exe (pip install).

SpeechRecognition – folosită pentru a recunoaște vorbirea de la un microfon live sau dintr-un fișier audio pentru a afișa în text.

PyAudio – este un set de legături Python pentru PortAudio, o bibliotecă C++ multiplatformă care se interferează cu driverele audio. Folosită pentru fișierele audio.

listen – folosita pentru a asculta vocea dintr-un microfon live

language – folosim când setăm limba română cu ’ro-RO’

4.Criterii de evaluare

Eu consider că programul evaluat nu este perfect, în mare parte depinde foarte mult de voce. Ca programul să funcționeze cum trebuie, detectează vocea în funcție de tonalitate, accent, pronunție clară. Dacă nu indeplinesc aceste criterii, programul „înțelege" altfel cuvintele și traduce diferit. Un alt factor important depinde și de distanța dintre sursa audio și microfon, cu cât e mai aproape se înțelege mai bine cuvintele. Cu cât e la distanță mai mare, programul trebuie să elimine zgomotele de fundal și distinge mai greu cuvintele pronunțate.

5.Abordare

Am incercat prin 2 abordări diferite la proiectul „Speech Recognition using Python", una folosind microfonul de la laptop/altă sursă pentru recunoașterea vorbirii și alta folosind un program care transformă cuvintele dintr-un fișier audio în text folosind „recognize_google".

Prima variantă, setăm microfonul și rulăm programul. Programul v-a afișa un mesaj „Liniște, calibrare zgomot fundal..." pentru a elimina zgomotele din fundal și pentru a se concentra la vocea noastră, după un timp se afișează mesajul „Calibrat, poți vorbi...", sunetele din fundal au fost calibrate și putem vorbi cu ușurință la microfon. Secvența „recognize_google" ajută să identifice ce am spus noi (în limba română), pentru ca mai departe să afișeze pe ecran în text. În cazul în care nu recunoaște ce am spus noi sau nu se înțelege cuvintele dă eroare.

A doua variantă, citim fișierul audio, folosim „recognize_google" și rulăm programul. Programul analizează cuvintele, frazele spuse din fișierul audio și afișează pe ecran cuvintele analizate. Pentru o bună identificare a cuvintelor setăm limba în română folosind „language='ro-RO' " pentru a afisa cuvintele înțelese în română.

6.Rezultate și analiză

La prima variantă folosind microfonul, putem vorbi în orice limbă pentru a afișa pe ecran text, de menționat că programul nu este conceput să înțeleagă cum trebuie vorbirea. Vocile diferă în funcție de tonalitate, accent, pronunție și tot asa. O altă problemă pentru care programul nu înțelege cuvintele este distanța dintre microfon și sursa audio (distanța mare), este recomandat să fie la distanța mică atunci când se dorește să recunoască vorbirea.

A doua variantă folosind fișierul audio, setăm limba în română folosind „ language=’ro-RO’ ”. Cuvintele din fișierul audio nu seamănă cu cele afișate pe ecran, de exemplu, lipsesc semnele de punctuație. Pentru acest caz folosim WER (word error rate) rata de eroare a cuvintelor, care este o măsură comună a performanței unui sistem de recunoaștere a vorbirii sau de traducere automată. WER poate fi calculată ca WER = $\frac{S+D+I}{N}$, unde S- nr de substituții, D- nr de ștergeri, I- nr de inserții, C- nr de cuvinte corecte și N- nr de cuvinte de referință. În caz contrar, programul dă eroare pentru că nu a reușit să recunoască vocea din fișierul audio.

7.Concluzii

Am învățat mult făcând acest proiect, m-a ajutat să înțeleg cum funcționează acest program și cum ajunge vorbirea folosind un microfon live (sau fișier audio) și trece printr-un proces, ca la final să ajungă la rezultatul dorit. Am demonstrat că se poate vorbi și recunoaște vorbirea pentru a fi convertită în text.

8.Referințe

- https://pypi.org/project/SpeechRecognition/ (am instalat librăria pentru Speech recognition folosind pip install SpeechRecognition în Command Prompt)

- https://www.slideshare.net/Simplilearn/speech-recognition-using-python-how-speech-recognition-works-in-python-simplilearn/Simplilearn/speech-recognition-using-python-how-speech-recognition-works-in-python-simplilearn (Am preluat informații/imagini pentru a pune în documentație)

- https://www.youtube.com/watch?v=PWVH3Vx3dCI (Am urmarit pașii din tutorial pentru a dezvolta proiectul Speech Recognition Using Python folosind microfonul live)

- https://data-flair.training/blogs/python-speech-recognition-ai/ (cod pentru recunoașterea textului dintr-un fișier audio)

- https://stackabuse.com/introduction-to-speech-recognition-with-python/ (cum să convertesc dintr-un fișier audio în text)

- https://en.wikipedia.org/wiki/Word_error_rate (am preluat informații despre WER)