**Voice text aligning algorithms**

**Grupa A3**

**Modulul foneme -> text**

Grosu Teodora-Elena

Rebegea Bogdan-Florin

Sima Paul Vladut

Ticu Madalina

Zaharia Alexandru Danut

1. **Descriere:**

System capable of aligning a voice recording with the corresponding text. The system must identify areas in the audio or text that do not correspond to the parallel resource or areas that contain content that does not correspond to the language known to the system (nonspecific phoneme, pronunciation errors, noises).

Extensions: The system allows indexing and compacting aligned resources by identifying "perfect" identities between voice-text aligned pairs. Eventually, the system also lets you generate a textbased on a voice signal or recognize a voice signal.

The method must be independent of the speaker.

1. **Modulul foneme -> text**

**Saptmana 1:**

Research legat de tema proiectului:

Definitia termenului “fonem”:

*fonem sn [At: DA ms / Pl: ~e / E: fr phonème] (Lin) 1 Cea mai mică unitate sonoră a limbii, care are funcțiunea de a diferenția cuvintele între ele, precum și formele gramaticale ale aceluiași cuvânt. 2 (Înv) Sunet.*

Fonologia limbii române stabilește inventarul de unități segmentale (vocale și consoane) și unități suprasegmentale (accent, intonație) cu valoare funcțională, caracteristic limbii române. Totodată, ca în fonologia oricărei alte limbi, sunt determinate seriile de sunete echivalente (alofone) prin care se realizează fiecare fonem în parte.

Prin analiza valorii funcționale a diferitelor aspecte fonetice se ajunge la concluzia că limba română standard folosește următoarele unități segmentale:

• șapte vocale,

• patru semivocale, care în unele analize sunt echivalate cu vocalele corespunzătoare,

• o vocală asilabică și devocalizată /ʲ/, care uneori este echivalată cu vocala /i/,

• douăzeci de consoane

și următoarele unități suprasegmentale:

• accent,

• intonație,

fiecare dintre acestea putându-se realiza fonetic în mai multe moduri echivalente fonologic.

Lipsesc așadar din inventarul fonematic al limbii române numeroase vocale și consoane, clicuri, croneme, tonuri etc. Astfel de aspecte fonetice pot apărea totuși de exemplu în interjecții, pronunții idiolectice, regionale sau arhaice, ori în împrumuturi neadaptate.

Unitățile segmentale sunt sunetele simple (vocale și consoane) din care se alcătuiesc silabele și apoi cuvintele. Din punct de vedere funcțional, existența a două categorii distincte de sunete --- vocale și consoane --- și diferența dintre ele se poate demonstra astfel:

1. Se constată că seria [a, e, i, ...] și seria [b, k, d, ...] sunt incompatibile, în sensul că dacă în cadrul fiecărei serii în parte prin înlocuirea unui sunet cu altul din aceeași serie se obține fie un cuvânt cu un alt sens fie un cuvânt perceput ca posibil, dar fără sens (de exemplu par-por sau par-car), înlocuirea unui sunet dintr-o serie cu un sunet din cealaltă serie nu este posibilă fără schimbarea structurii cuvântului (de exemplu aer-cer, unde numărul de silabe se modifică).

2. Nu este posibilă eliminarea dintr-un cuvânt a unui sunet din seria [a, e, i, ...] (înlocuirea lui cu zero) fără ca structura cuvântului să se modifice. În schimb sunetele din seria [b, k, d, ...] pot lipsi din cuvânt.

Astfel se face distincție între vocale, ca sunete care pot alcătui singure o silabă și care pot purta accentul, și consoane, care nu au această calitate. Există desigur și alte definiții pentru vocale și consoane, care au în vedere modul de articulare (deci aspectul fonetic) sau proprietățile acustice ale acestora, dar în subiectul de față interesează cu precădere relațiile pe care le contractează diversele sunete. Trebuie remarcat de asemenea că o astfel de analiză funcțională mai are avantajul de a releva rolul de vocală sau consoană al unui sunet în particular pentru limba română; în alte limbi aceleași sunete pot juca alte roluri, de exemplu în limba engleză sunetele [l] și [m] pot funcționa în anumite condiții ca vocale.

**Webgrafie:**

Proiect de cercetare-dezvoltare in Tehnologia Vorbirii: pg 23:

<http://speed.pub.ro/speed3/wp-content/uploads/2014/02/Indrumar-de-proiect-PCDTVv11.pdf?fbclid=IwAR2hfdOEoNpPKAdPfJ3InmWdfRwR_pEyy7EPnIoMaNFpsYoJlKp3eV4vriA>

Phoneme-to-Text Transcription System with an Infinite Vocabulary: <http://www.ar.media.kyoto-u.ac.jp/publications/Co>

**Saptamana 2:**

Speech Recognition Engine CMU PocketSphinx:

<https://github.com/cmusphinx/pocketsphinx>

Folosind functia de generare text din audio din libraria PocketSphinx, am obtinut cuvinte mult mai aproapiate de textul real decat incercand sa generam foneme din audio, si apoi transformand fonemele in text.

Textul obtinut:

*to the end of world war two world was split into two are*

*east and west*

*this marked the beginning of the air are cold the cold war*

Textul original:

*After the end of the world war two, the world was split into two: east and west.*

*This marked the beginning of the era called the cold war*

Dupa cum se observa, cuvintele pot fi mult mai usor aproximate, algorimul de generare audio-text fiind mult mai efficient decat audio-foneme

Pentru a mari eficienta speech-recognition si pentru a obtine un rezultat si mai apropiat de adevar, putem recurge la urmatorii algoritmul descris in lucrarea de la adresa:

https://arxiv.org/ftp/arxiv/papers/1105/1105.6162.pdf

Algoritmul Viterbi (https://en.wikipedia.org/wiki/Viterbi\_algorithm) este un algoritm de programare dinamica ce ar putea sa ne ajute sa detectam secventa corecta de cuvinte si a aproximam si mai mult cuvintele initiale(in cazul in care obtinem o secventa de cuvinte legate intre ele, intrucat in vorbirea libera, pauzele dintre cuvinte

sunt aproape insesizabile)

**Keywords**: Word boundary detection, sequence memory, text segmentation, temporal pattern

recognition, word splitting, Viterbi algorithm.

In acelasi timp, ne putem folosi de POST(part-of-speech-tagging) pentru a reusi sa identificam ce parte de vorbire

corespunde fiecarui cuvant dintr-un text, in functie de context, pentru a obtine o propozitie care are sens(in loc

de o insiruire de cuvinte despartite de spatii)

**Contributie individuala:**

Teodora Grosu si Madalina Ticu: dezvoltarea unui algoritm care sa parseze dictionarul de cuvinte si care compara fonemele din input cu cele din dictionar pentru a recunoaste cuvintele.

Detalii de implementare:

def PhenomeToText(path):

cuvinteRezultate = list()

dict1 = parse\_dictionary("dictionary.txt")

file = open(path)

for line in file:

#print(line)

cuvinte = line.split("SIL")

for i in cuvinte:

i = i.strip(" ")

#pentru cuvintele rezultate din audio->foneme:

for x in dict1:

if i == x:

cuvinteRezultate.append(dict1[x])

#incerc sa recunosc cuvintele:

k = ""

for j in i:

k = k + j

#print(k)

for y in dict1:

if k == y:

cuvinteRezultate.append(dict1[y])

k = "" #caut urmatorul cuvant

break

return cuvinteRezultate

if \_\_name\_\_ == '\_\_main\_\_':

print(PhenomeToText("fonemeRezultate.txt"))

Functia returneaza o lista de cuvinte gasite, care vor fi apoi scrise intr-un document text, pentru a putea fi folosit de celelalte module :

<https://github.com/GorganGabi/UAIC-FII-AI-sound-text-alignment/blob/master/Phenome_to_text/textul_final.py>

O alta abordare :

Zaharia Alex, Sima Vlad si Rebegea Bogdan: dezvoltarea unui dictionar propriu de foneme si a unui script care recunoaste cuvintele pe baza fonemelor

Detalii de implementare:

def PhenomeToText():

dict1={…} #dictionarul

with open ('test.txt', encoding='utf8') as f:

for line in f:

phenome = ' '.join([dict1.get(word, word) for word in line.split()])

print(phenome) #return phenome

**Observatii:** Algoritmii prezinta o acuratete scazuta, nu putem recunoaste fonemele, intrucat acestea nu sunt complete

**Sarcini pentru saptamana 3:**

* Cautarea unei modalitati de a recunoaste fonemele in limba romana
* Optimizarea algoritmului de recunoastere a fonemelor
* Gasirea unui dictionar de foneme in limba romana