

UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI
FACULTATEA DE ELECTRONICĂ, TELECOMUNICAȚII ȘI TEHNOLOGIA INFORMAȚIEI

SISTEM DE RECUNOAȘTERE AUTOMATĂ A VORBIRII ÎN LIMBA
ROMÂNĂ BAZAT PE REȚELE NEURONALE PROFUNDE

LUCRARE DE DISERTAȚIE

prezentată ca cerință parțială pentru obținerea titlului de
Master în domeniul *Inginerie Electronică și Telecomunicații*
programul de studii de masterat *BIOSINF*

Conducător științific

Conf. Dr. Ing. Horia CUCU

Absolvent

Ing. Alexandru Lucian GEORGESCU

București
2018

TEMA LUCRĂRII DE DISERTAȚIE

a masterandului **GEORGESCU E. Alexandru-Lucian , 421-BIOSINF**

1. Titlul temei: Sistem de recunoaștere automată a vorbirii în limba română bazat pe rețele neuronale profunde

2. Descrierea contribuției originale a masterandului (în afara părții de documentare):

- Extinderea corpusului de vorbire în limba română. Crearea unui sistem de colectare și adnotare automată a vorbirii existente în publicații online.
- Antrenarea modelului acustic cu diferite topologii de rețele neuronale profunde, în diferite configurații.
- Crearea modelului de limbă bazat pe două abordări principale: probabilistic (n-gram) și cu ajutorul rețelelor neuronale recurente (RNNLM).
- Evaluarea sistemelor de recunoaștere automată a vorbirii create. Se vor efectua experimente pe baza cărora vor fi furnizate rezultate ce indică performanțele obținute de către modelele acustice și de către modelele de limbă. Se vor realiza măsurători referitoare la resursele de calcul necesare funcționării întregului sistem.

3. Resurse folosite la dezvoltarea proiectului:

bază de date de vorbire în limba română, bază de date de text în limba română, dicționare și resurse fonetice, utilitarul Kaldi, limbajul de programare Java

4. Proiectul se bazează pe cunoștințe dobândite în principal la următoarele 3-4 discipline:

ICVSI; TVRV; TACAI

5. Proprietatea intelectuală asupra proiectului aparține: U.P.B.

6. Data înregistrării temei: 2017-11-28 22:54:01

Conducător(i) lucrare,
Conf. dr. ing. Horia CUCU

semnătura:

Responsabil program master,
Prof. dr. ing. Dragoș BURILEANU

semnătura:

Student,

semnătura:

Decan,

Prof. dr. ing. Cristian NEGRESCU

semnătura:

Cod Validare: **3a6564928d**

Declarație de onestitate academică

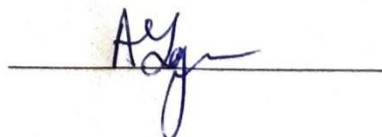
Prin prezenta declar că lucrarea cu titlul "*Sistem de recunoaștere automată a vorbirii în limba română bazat pe rețele neuronale profunde*", prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității "Politehnica" din București ca cerință parțială pentru obținerea titlului de *Master* în domeniul *Inginerie Electronică și Telecomunicații*, programul de studii *BIOSINF*, este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

Declar că toate sursele utilizate, inclusiv cele de pe Internet, sunt indicate în lucrare, ca referințe bibliografice. Fragmente de text din alte surse, reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și fac referință la sursă. Reformularea în cuvinte proprii a textelor scrise de către alți autori face referință la sursă. Înțeleg că plagiatul constituie infracțiune și se sancționează conform legilor în vigoare.

Declar că toate rezultatele simulărilor, experimentelor și măsurărilor pe care le prezint ca fiind făcute de mine, precum și metodele prin care au fost obținute, sunt reale și provin din respectivele simulări, experimente și măsurători. Înțeleg că falsificarea datelor și rezultatelor constituie fraudă și se sancționează conform regulamentelor în vigoare.

București, 22.06.2018

Absolvent *Alexandru Lucian GEORGESCU*

A handwritten signature in blue ink, appearing to read 'Alex', is written over a horizontal line.

CUPRINS

Cuprins.....	7
Listă de Figuri.....	9
Listă de Tabele.....	11
Listă de Acronime.....	13
INTRODUCERE	15
Motivația lucrării	15
Obiectivele lucrării.....	16
Structura lucrării	17
CAPITOLUL 1 Rețele neuronale profunde	15
1.1 Aspecte generale	15
1.2 Rețele complet conectate.....	18
1.3 Rețele convoluționale (CNN).....	18
1.4 Rețele neuronale recurente (RNN).....	19
CAPITOLUL 2 Metode și modele pentru recunoașterea automată a vorbirii.....	21
2.1 Modelare acustică.....	23
2.1.1 Parametrizarea semnalului vocal	23
2.1.2 Modelul Markov ascuns (HMM).....	24
2.1.3 Mixturi de densități gaussiene (GMM).....	26
2.1.4 Rețele neuronale cu întârziere în timp (Time delay neural network -TDNN).....	27
2.2 Modelare lingvistică.....	28
2.2.1 Modele statistice de tip n-gram.....	28
2.2.2 Rețele neuronale recurente pentru modelarea lingvistică (RNN-LM).....	29
2.3 Resurse fonetice, acustice, lingvistice pentru RAV	30
2.4 Evaluarea RAV	32
CAPITOLUL 3 Metode de antrenare nesupervizată a modelelor acustice pentru RAV	34
3.1 Problema resurselor pentru limbi slab dotate	36
3.2 Metode de obținere a complementarității RAV	37
3.3 Îmbinarea și aplicarea metodelor de complementaritate.....	39
3.4 Metode de selecție.....	40
3.5 Îmbinarea și aplicarea metodelor de selecție.....	42
3.6 Performanțe ale metodelor de adnotare automată și antrenare nesupervizată	44

3.7	Concluzii	45
CAPITOLUL 4	Experimente de recunoaștere automată a vorbirii în limba română.....	46
4.1	Stadiul anterior și elemente inovative	46
4.1.1	Extinderea corpusului de vorbire.....	47
4.1.2	Extragerea trăsăturilor acustice.....	47
4.1.3	Îmbunătățiri la nivelul modelului acustic de tip HMM-GMM.....	47
4.1.4	Îmbunătățiri la nivelul modelului lingvistic și al vocabularului	48
4.1.5	Modelarea acustică folosind rețele neuronale.....	48
4.1.6	Modelarea lingvistică folosind rețele neuronale	48
4.1.7	Utilitarul Kaldi.....	48
4.2	Pregătirea experimentelor	49
4.2.1	Modele acustice	49
4.2.2	Modele lingvistice	50
4.2.3	Corpusul de vorbire pentru antrenare și evaluare	50
4.3	Rezultate experimentale de recunoaștere automată a vorbirii	51
4.3.1	Modele acustice de tip HMM-GMM și HMM-DNN antrenate cu Kaldi	51
4.3.2	Modele de limbă complexe de tip n-gram	53
4.3.3	Reevaluarea laticelor folosind modele de limba n-gram	54
4.3.4	Reevaluarea laticelor folosind modele de limba RNN	56
4.3.5	Sumarizarea rezultatelor finale și concluzii.....	57
4.4	Aplicația demonstrativă de RVC-VE.....	58
4.5	Experimente de antrenare nesupervizată a modelelor acustice.....	59
4.5.1	Metoda de adnotare automată	59
4.5.2	Pregătirea experimentelor	61
4.5.3	Rezultate experimentale.....	62
4.5.4	Concluzii.....	63
	Concluzii finale.....	64
	Contribuții personale	66
	Bibliografie.....	68

LISTĂ DE FIGURI

Figura 1.1 Arhitectura generală a unei rețele neuronale	16
Figura 1.2 Funcția de activare treaptă.....	17
Figura 1.3 Funcția de activare sigmoidală	17
Figura 1.4 Funcția de activare ReLU	18
Figura 1.5 Perceptronul multistrat	18
Figura 1.6 Rețea neuronală convoluțională	19
Figura 1.7 Rețea neuronală recurentă	20
Figura 2.1 Arhitectura unui sistem RAV	22
Figura 2.2 Extragerea parametrilor MFCC	23
Figura 2.3 Arhitectura HMM.....	25
Figura 2.4 Funcția densitate de probabilitate normală.....	26
Figura 2.5 Arhitectura unei rețele TDNN.....	27
Figura 2.6 Structura unui RNN pentru modelare lingvistică	29
Figura 3.1 Adnotarea automată a corpusurilor audio.....	35
Figura 4.1 Interfața aplicației demonstrative de RAV	58
Figura 4.2 Arhitectura aplicației RAV	59
Figura 4.3 Metodă de adnotare automată a vorbirii	60

LISTĂ DE TABELE

Tabelul 2.1 Fonemele limbii române	31
Tabelul 3.1 Resurse de vorbire în limba română	37
Tabelul 4.1 Corpusul de vorbire	51
Tabelul 4.2 Rezultate pentru modelele acustice bazate pe HMM-GMM	51
Tabelul 4.3 Rezultate pentru modele acustice bazate pe DNN.....	52
Tabelul 4.4 Ratele OOV în raport cu creșterea vocabularului.....	53
Tabelul 4.5 Evaluarea modelelor de limba în contextul RAV	54
Tabelul 4.6 WER[%] folosind tehnica de reevaluare a laticelor cu modele n-gram	55
Tabelul 4.7 Consumul de memorie și factorul de timp real.....	56
Tabelul 4.8 WER[%] folosind tehnica de reevaluare a laticelor cu modele RNN	56
Tabelul 4.9 Compararea sistemelor RVC-VE	58
Tabelul 4.10 Corpusurile de vorbire folosite în procesul de adnotare automată	61
Tabelul 4.11 Sistemele RAV folosite la procesul de adnotare automată.....	62
Tabelul 4.12 Evaluarea metodei de adnotare automată	62
Tabelul 4.13 Rezultatele adnotării automate asupra corpusului nou achiziționat	62
Tabelul 4.14 Performanțele noilor sisteme RAV după reantrenare	63

LISTĂ DE ACRONIME

RAV - recunoașterea automată a vorbirii
RVC-VE - recunoașterea automată a vorbirii continue cu vocabular extins
DNN - rețele neuronale profunde
ReLU - rectifier linear unit
MLP - perceptron multistrat
CNN - rețele convoluționale
RNN - rețele neurale recurente
GMM - mixturi de densități gaussiene
MFCC - coeficienți mel-cepstrali
FFT - transformata Fourier rapidă
DCT - transformata cosinus discretă
iVector - vector identitate
JFA - Joint Factor Analysis
LDA - analiza discriminatorie liniară
WCCN - Within-Class Covariance Normalization
HMM - modelul Markov ascuns
EM - Expectation Maximization
TDNN - Rețele neuronale cu întârziere în timp
fMMLR - feature-Space Maximum Likelihood Linear Regression
ChER - rata de eroare la nivel de caracter
WER - rata de eroare la nivel de cuvânt
SER - rata de eroare la nivel de propoziție
PPL - perplexitatea
OOV - rata de apariție a cuvintelor necunoscute
PLP - coeficienți perceptuali de predicție liniară
VTLN - normarea lungimii tractului vocal
HLDA - Heteroscedastic Linear Discriminant Analysis
PCA - analiza componentei principale
CMVN - normarea mediei și varianței cepstrului
SGMM - subspații de mixturi gaussiene
UBM - model universal de vorbire
MMI - Informația Mutuală Maximă
FBMMI - Space Boosted Maximum Mutual Information feature
FBANK - trăsături extrase cu bancuri de filtre Mel
ROVER - Recognizer Output Voting Error Reduction
WTN - Word Transition Network
A-stabil - stabilitate acustică
CRF - Conditional Random Fields
MLLT - metoda verosimilității maxime
SAT - adaptare la vorbitor
PNCC - coeficienți cepstrali cu energie normalată
DTW - Dynamic Time Warping

INTRODUCERE

MOTIVAȚIA LUCRĂRII

Vorbirea este una dintre cele mai importante forme de comunicare interumană și, având în vedere acest fapt, s-a dorit întotdeauna extinderea interacțiunii vocale, astfel încât să fie posibilă relaționarea în mod cât mai natural cu tehnologiile care ne înconjoară. Ultimii ani au adus o creștere semnificativă a sistemelor de recunoaștere a vorbirii (RAV), fiind înregistrat un deosebit succes al dispozitivelor acționate pe cale vocală. Astfel de sisteme sunt capabile să realizeze sarcini de detecție a cuvintelor cheie rostite, ce au rolul de a "trezi" dispozitivul, pentru a detecta momentul în care utilizatorul i se adresează. De asemenea, sistemele pot realiza apoi transcrierea integrală a comenzii, ce va fi ulterior interpretată. Utilizarea lor este întâlnită frecvent în aplicații de tip asistent personal virtual (Cortana [72], Amazon Alexa [80], Siri [74]) sau sisteme de tip casă inteligentă (Amazon Echo [77], Apple HomeKit [69]). Acest tip de interacțiune este mai ușor și mai rapid, având avantajul că nu necesită cunoștințe suplimentare de operare și control al dispozitivelor, vorbirea fiind un proces natural și automat.

Totodată, interesul pentru sistemele de recunoaștere automată a vorbirii continue cu vocabular extins (RVC-VE) a crescut, acestea fiind foarte utile în sarcinile de dictare, ce permit de exemplu transcrierea unui discurs. Provocările în acest caz sunt reprezentate în special de natură spontană a vorbirii și variabilitatea vorbitorilor.

Totuși, sistemele de RAV prezintă unele aspecte sensibile. Câteva dintre acestea sunt: acuratețea sistemului, timpul în care acesta procesează vorbirea, resursele consumate, etc. Fiecare aspect poate fi mai mult sau mai puțin relevant, în funcție de scopul final al aplicației. O acuratețe scăzută va conduce la transcrieri greșite ale semnalului vocal, iar acest fapt va genera interpretări greșite. O latență mare a sistemului îl va face foarte greu de utilizat într-un scenariu de interacțiune în timp real, de tip dialog. Problema resurselor consumate (energie, putere de

calcul) poate să devină serioasă dacă dispozitivul pe care va rula aplicația este un terminal mobil, alimentat de către o baterie sau dacă există limitări din punct de vedere hardware.

Această lucrare aduce în prim-plan progresele făcute în urma aprofundării noțiunilor de RAV, în contextul activității mele în cadrul laboratorului de cercetare Speed (Speech and Dialogue) [76] din cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației. Încheierea studiilor de licență s-a finalizat cu un proiect în care am abordat crearea unor sisteme RAV simple, cu vocabular redus. Acestea erau destinate recunoașterii de cifre conectate în limba română, respectiv a câtorva tipuri predefinite de fraze în limba engleză.

Lucrarea de față își propune abordarea unei sarcini mai complexe, și anume crearea unui sistem RVC-VE pentru limba română, folosind tehnologii bazate pe rețele neuronale profunde (*deep neural networks* - DNN). În momentul actual, acestea sunt considerate starea artei. Acest fapt a pornit de la contextul în care s-a dorit îmbunătățirea unui sistem RVC-VE anterior, creat în cadrul grupului Speed în anul 2014. Deși sistemul este funcțional, prin comparație cu sistemele existente pentru alte limbi, acuratețea acestuia dovedește faptul că nu este suficient de bun, fiind în continuare posibilă îmbunătățirea sa. Astfel, s-a încercat introducerea unor elemente suplimentare, care să ofere această îmbunătățire. Folosirea unor corpusuri de vorbire și text mai mari reprezintă principalele resurse care pot conduce la crearea unor modele acustice și lingvistice mai performante. Uneori, cum este cazul și limbii române, aceste resurse nu sunt foarte bogate, iar procurarea lor este dificilă. Totodată, modelele pot deveni mai performante dacă la crearea lor se folosesc algoritmi mai eficienți. Așadar, toate aceste idei enunțate reprezintă motivația ce stă la baza temei abordate în prezenta lucrare.

OBIECTIVELE LUCRĂRII

Lucrarea de față are ca scop îmbunătățirea unui sistem de RVC-VE pentru limba română. Elementul de noutate principal îl reprezintă folosirea rețelelor neuronale pentru antrenarea modelelor componente ale sistemului: acustic și lingvistic. Pentru eficiență, infrastructura de calcul va fi îmbunătățită.

Deoarece crearea unor astfel de sisteme de recunoaștere a vorbirii depinde în mod esențial de antrenarea pe baza unui corpus de vorbire cât mai mare, se va implementa o aplicație ce colectează date audio de pe Internet. Apoi se vor studia metodele de adnotare automată a acestor date și se vor crea noi corpusuri de vorbire.

Pentru realizarea acestor obiective, vor fi efectuate următoarele sarcini:

- extinderea infrastructurii de calcul în vederea procesării distribuite folosind procesoare grafice;
- crearea unor modele acustice mai puternice, folosind tehnici ce conduc la optimizarea modelelor statistice și crearea unor modele noi, pe bază de rețele neuronale;
- crearea unor modele de limbă mai puternice, prin extinderea vocabularului și a complexității, dar și prin antrenarea unor modele noi, pe bază de rețele neuronale;
- evaluarea sistemului de RAV anterior și a sistemului curent în diferite configurații;
- analiza resurselor de calcul necesare pentru sarcină de RAV;
- implementarea unei aplicații de transcriere în timp real;
- crearea unui sistem de colectare și adnotare automată a corpusurilor audio;
- reantrenarea sistemelor pe baza corpusurilor nou achiziționate.

STRUCTURA LUCRĂRII

Capitolul 1 oferă o viziune de ansamblu asupra rețelelor neuronale, oferind câteva detalii despre modul în care acestea funcționează, principii de bază și domeniile lor de aplicabilitate. Sunt prezentate câteva arhitecturi de DNN, împreună cu particularitățile lor.

Capitolul 2 realizează o introducere a noțiunilor fundamentale referitoare la sistemele de RAV. Este prezentată arhitectura generală a unui astfel de sistem, împreună cu modulele componente. Este descris modul în care semnalul vocal este prelucrat și sunt explicate metodele și algoritmii ce stau la baza creării sistemului, precum și resursele necesare.

Capitolul 3 reprezintă un studiu al literaturii de specialitate asupra metodelor de antrenare nesupervizată a modelelor acustice. Din cauza lipsei bazelor de date de vorbire, acestea fiind o resursă esențială în sarcină de RAV, sunt investigate principalele metode prin care acestea pot fi create în mod automat. Se realizează o comparație a metodelor.

Capitolul 4 descrie pe larg noutățile aduse în încercarea de a crea un nou sistem de RAV, mai performant. Se insistă asupra modului în care experimentele au fost desfășurate, asupra rezultatelor obținute și a comparației între ele. În final, se oferă detalii despre configurarea unei aplicații software ce realizează transcrierea vorbirii în limba română în timp real.

A doua parte a capitolului prezintă experimentele de antrenare nesupervizată a modelelor acustice. Este descris modul în care se achiziționează automat un corpus de vorbire de pe Internet, adnotat ulterior cu o metodă descrisă și evaluată în cadrul capitolului. Sunt prezentate rezultate în urma reantrenării modelului acustic folosind corpusul nou obținut.

CAPITOLUL 1

REȚELE NEURONALE PROFUNDE

1.1 ASPECTE GENERALE

În ultimii ani, cele mai performante sisteme de inteligență artificială au avut la bază algoritmi din sfera învățării profunde (*deep learning*). Tehnica reprezintă o abordare a învățării automate (*machine learning*), folosind rețelele neuronale. Acestea nu sunt un concept tocmai nou, noțiunea datând încă din preajma anilor 1940 [57]. În 1958 apare prima încercare de clasificare bazată pe învățare automată, folosită la recunoașterea de imagini [55]. Rețelele neuronale au avut și momente de stagnare de-a lungul timpului, dar în prezent ele se bucură din nou de un foarte mare succes. Diverse domenii folosesc în mod curent rețele neuronale, cele mai întâlnite aplicații sunt în domeniul recunoașterii de imagini, recunoașterea formelor, predicția datelor financiare, clasificare, asocierea datelor, simulări, etc.

De asemenea, în ultimii ani, rețelele neuronale profunde au început să fie tot mai des folosite și în recunoașterea automată a vorbirii și au surclasat sistemele clasice, probabilistice. În 2012, experimente ce au constatat în folosirea unor diverse configurații de rețele neuronale pentru sarcina de RAV cu vocabular extins în limba engleză [63] au dus la obținerea unor îmbunătățiri relative de 33% față de abordarea probabilistică.

Modelul este de inspirație biologică și face referire la modul în care sistemul nervos uman primește, procesează și transmite informația: o rețea de celule (neuroni), interconectate în diferite moduri (sinapse). Așa cum învățarea naturală în cazul oamenilor presupune analiza unor exemple și aplicarea lor în situații similare, cu ajutorul rețelelor neuronale se dorește ca mașinile

de calcul să deprindă realizarea anumitor sarcini, plecând de la analiza unor exemple de antrenare. Acestea constau în date de intrare adnotate, pentru care se cunosc datele de ieșire. Astfel, se caută o funcție general valabilă care să stabilească corespondența dintre ele. Figura 1.1 prezintă arhitectura unei rețele neuronale. Rețeaua este organizată în straturi, unde fiecare strat conține un număr de neuroni. Stratul de intrare are rolul de a prelua datele de intrare. Numărul de neuroni din acest strat corespunde cu dimensionalitatea datelor. Fiecare neuron reprezintă o caracteristică, un atribut al datelor de intrare. Apoi, stratul de intrare este conectat la stratul ascuns. Acesta realizează procesarea propriu-zisă. În cazul rețelelor profunde, există mai multe straturi ascunse. Numărul neuronilor de pe aceste straturi nu este un parametru stabilit teoretic. De cele mai multe ori, el este determinat empiric, împreună cu numărul straturilor ascunse. Stratul de ieșire furnizează datele rezultate în urma procesării. Într-o sarcină de clasificare, dimensiunea acestui strat este egală cu numărul claselor distincte.

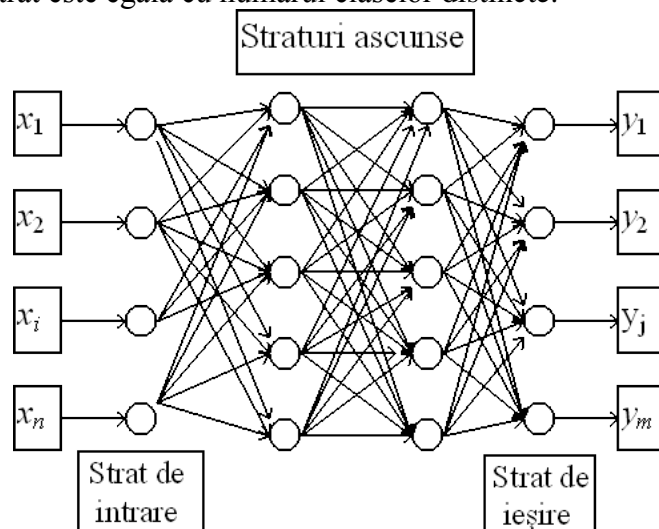


Figura 1.1 Arhitectura generală a unei rețele neuronale

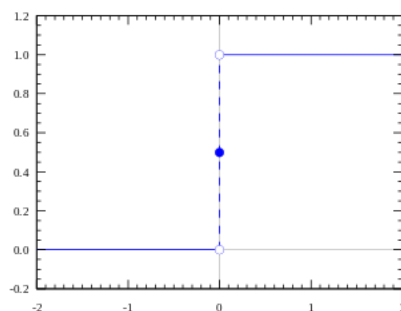
Sursa: [81]

Ieșirea fiecărui neuron de pe un strat intervine cu o anumită pondere în intrarea neuronilor de pe stratul următor. Astfel, neuronii sunt caracterizați de funcția de activare. Ea are rolul de a calcula suma ponderată a intrărilor și de a decide dacă neuronul se "activează", dacă va trimite mai departe rezultatul sumei.

$$Y = \sum_{i=1}^N (x_i * w_i) + b,$$

unde Y este ieșirea neuronului, formată din suma tuturor intrărilor x_i înmulțite cu ponderile lor, la care se adaugă factorul *bias*. Deoarece Y poate lua orice valori, funcția de activare stabilește dacă valoarea produsă de neuron se va transmite în straturile superioare. Una dintre cele mai simple funcții de activare este funcția treaptă, unde se realizează comparația cu un prag. Aceasta este utilă în clasificările binare, deoarece există numai două stări: neuronul se activează sau neuronul nu se activează.

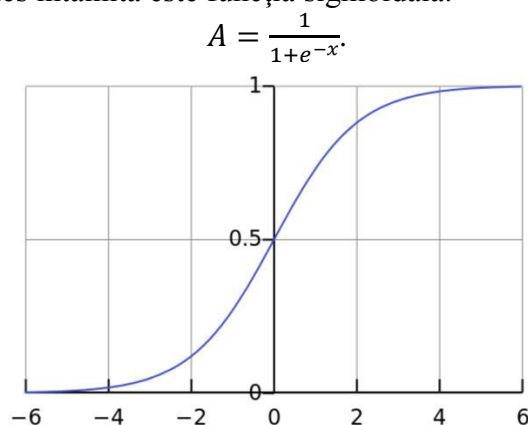
$$A = \begin{cases} 1, Y > \text{prag} \\ 0, Y \leq \text{prag} \end{cases}$$

**Figura 1.2 Funcția de activare treaptă**

Sursa: [78]

Există însă și situații când se dorește ca neuronii să fie mai mult sau mai puțin activați, existând valori intermediare ale ieșirilor, nu numai 0 sau 1. O astfel de funcție este funcția liniară, unde activarea este proporțională cu intrarea. Nu este însă o funcție de activare foarte utilă. Dată fiind natura liniară, indiferent de numărul de straturi, ieșirea ultimului strat este o funcție liniară ce depinde de intrarea primului strat. Așadar, această funcție nu poate fi folosită într-o rețea multistrat.

O altă funcție de activare des întâlnită este funcția sigmoidală:

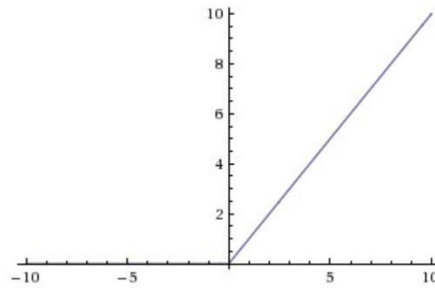
**Figura 1.3 Funcția de activare sigmoidală**

Sursa: [78]

Principalul avantaj al acesteia este faptul că funcția este neliniară și permite astfel utilizarea a mai mult de două straturi în rețea, iar activările nu sunt de tip binar. Funcția este utilă în sarcinile de clasificare, deoarece pentru valori de intrare în afara intervalului $[-2;2]$, tinde să realizeze distincții clare, îndreptându-se spre valorile de capăt, -1 sau 1. Faptul că pentru orice valoare de intrare, ieșirea se află într-o gamă bine stabilită, este un alt avantaj.

Rectifier Linear Unit (ReLU) este o altă funcție de activare des întâlnită. Este mai puțin solicitantă din punct de vedere computațional, deoarece invocă operații matematice simple, spre deosebire de funcția sigmoidală. ReLU este neliniară prin natură, fiind 0 pentru valori negative, iar pentru valori pozitive este egală cu funcția identitate.

$$A = \begin{cases} 0, & Y \leq 0 \\ x, & Y > 0 \end{cases}$$

**Figura 1.4 Funcția de activare ReLU**

Sursa: [78]

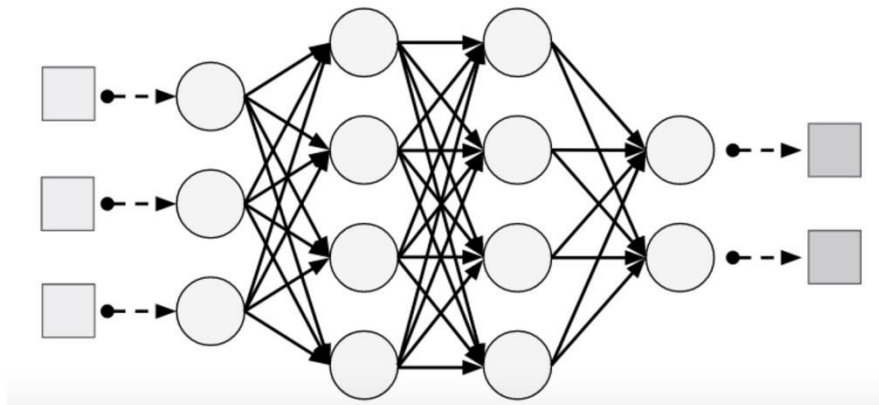
Acestea sunt doar câteva exemple din cele mai comune funcții de activare. Alegerea funcției de activare depinde de sarcină de clasificare sau predicție, de caracteristicile ce se doresc a fi approximate, de complexitatea finală a modelului.

Antrenarea rețelelor se face cu ajutorul algoritmului de propagare inversă (backpropagation) [59]. Este un algoritm iterativ, ce are ca scop modificarea ponderilor neuronilor, astfel încât să minimizeze funcția pierdere (*loss function*), până când se ajunge la convergență. Funcția pierdere reprezintă diferența dintre ieșirea actuală a rețelei și ieșirea dorită (referință). Este o metrică a erorii, ce indică precizia rețelei în cazul utilizării ieșirii actuale. Eroarea rețelei pentru n date de antrenare poate fi scrisă ca medie a funcțiilor de pierdere:

$$E = \frac{1}{2n} \sum_x \|y(x) - y'(x)\|^2.$$

1.2 REȚELE COMPLET CONECTATE

În rețelele complet conectate, fiecare neuron este conectat la fiecare neuron din stratul precedent, fiecare conexiune având o pondere proprie. Rețeaua constă în cel puțin 3 straturi de neuroni. Este o arhitectură foarte costisitoare din punct de vedere al consumului de memorie, din cauza numărului mare de ponderi, dar și al puterii de calcul necesare, din cauza multiplelor conexiuni. O astfel de rețea este perceptronul multistrat (*multilayer perceptron* - MLP):

**Figura 1.5 Perceptronul multistrat**

Sursa: [70]

1.3 REȚELE CONVOLUȚIONALE (CNN)

Rețelele convoluționale și-au dovedit cu succes aplicabilitatea în recunoașterea și clasificarea imaginilor. Specifice acestei rețele sunt straturile convoluționale și straturile de tip *pooling*. În prelucrarea de imagini, stratul de intrare este reprezentat de o matrice care conține valorile pixelilor și are dimensiunea egală cu cea a imaginii. Pe această matrice este aplicată o operație

de convoluție, cu o matrice de ponderi de lungime fixă (3x3, 4x4, etc.). În urma acestei operații, dimensiunea imaginii inițiale este redusă. Pentru a evita reducerea, uneori se recurge la bordarea imaginii inițiale cu valori de zero, procedura numită *padding*. Un alt parametru al rețelelor convoluționale este pasul cu care se deplasează fereastră de convoluție (*stride*). Straturile de tip *pooling* sunt utilizate atunci când imaginile au dimensiuni foarte mari și este necesară reducerea numărului de parametri antrenabili. Neuronii ce folosesc ca funcție de activare funcția *maxout*, vor avea valoarea egală cu maximumul intrărilor, în timp ce alți neuroni pot folosi o funcție de mediere a intrărilor. În CNN, straturile inferioare învață caracteristici primare ale imaginilor: forme, margini, orientări. Straturile superioare învață caracteristici complexe.

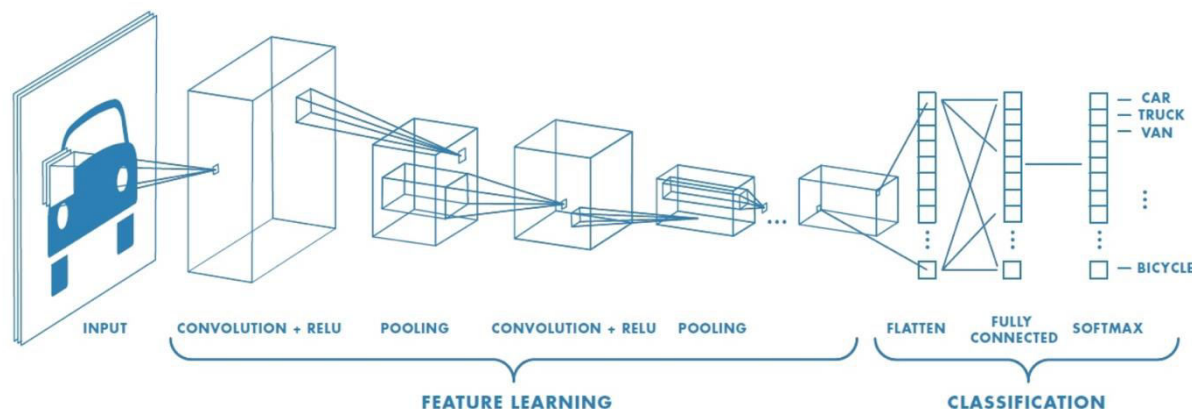


Figura 1.6 Rețea neuronală convoluțională

Sursa: [73]

Câteva arhitecturi foarte cunoscute de rețele neuronale bazate pe CNN sunt: LeNet [60], AlexNet [61], ZF Net [68], GoogLeNet [51], VGGNet [52].

1.4 REȚELE NEURONALE RECURENTE (RNN)

Rețelele neuronale recurente sunt foarte eficiente în sarcini ce modelează date cu caracter secvențial, ținând cont de dependențele temporale. Recunoașterea vorbirii sau a scrisului de mână, procesarea limbajului natural, predicția evenimentelor sunt câteva dintre domeniile unde pot fi aplicate.

Principala caracteristică a acestor rețele este date de conexiunile de tip ciclu între neuroni. Aceștia învață atât din intrarea curentă, cât și din informațiile învățate pe baza intrărilor din trecut. Din această cauză, RNN sunt considerate ca fiind rețele cu "memorie".

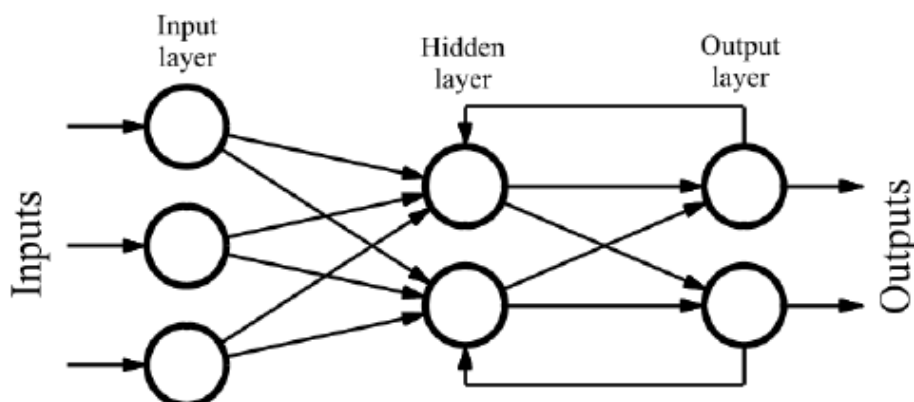


Figura 1.7 Rețea neuronală recurentă

Sursa: [71]

CAPITOLUL 2

METODE ȘI MODELE PENTRU RECUNOAȘTEREA AUTOMATĂ A VORBIRII

Recunoașterea automată a vorbirii (RAV) are ca obiectiv preluarea unui semnal audio ce conține vorbire și transformarea lui în textul aferent. O reprezentare matematică a acestei afirmații, presupune identificarea celei mai probabile secvențe de cuvinte W^* , corespunzătoare semnalului vocal X :

$$W^* = \operatorname{argmax} P(W|X).$$

Deoarece $P(W,X)$ este dificil de a fi calculată direct, se folosește regula lui Bayes pentru a transforma într-o formă echivalentă:

$$W^* = \operatorname{argmax} P(W|X) = \operatorname{argmax} \frac{P(X|W) * P(W)}{P(X)} = \operatorname{argmax} P(X|W) * P(W).$$

Deoarece $P(X)$ nu depinde de secvența de cuvinte W , întreaga problemă se rezumă la două aspecte:

- Calculul probabilității semnalului vocal X , știind secvența de cuvinte W care îi corespunde: $P(X|W)$. Această probabilitate se obține în urma antrenării modelului acustic.
- Calculul probabilității unei secvențe de cuvinte: $P(W)$. Această probabilitate se obține în urma antrenării modelului de limbă.

În general, crearea și utilizarea unui sistem RAV implică două mari etape.

Etapa de antrenare presupune crearea propriu-zisă a sistemului, a modulelor sale componente, în special a modelului acustic și a modelului lingvistic. În această fază este obligatoriu necesară existența atât a unui corpus de vorbire adnotat, pentru care există transcrierea textuală, cât și a unui corpus de text. Astfel, sistemul "învață" o corespondență între vorbire și text, dar și modul în care cuvintele pot alcătui o propoziție valabilă.

Etapa de decodare presupune folosirea informațiilor furnizate de aceste modele pentru a reuși aprecierea cât mai corectă a transcrierii corespunzătoare unui semnal vocal oferit la intrarea sistemului.

Figura 2.1 prezintă schema de principiu a unui sistem RAV, împreună cu cele două etape și modulele componente.

Modelul acustic este antrenat folosind o bază de date adnotată de vorbire, de dimensiuni cât mai mari și cu pronunții cât mai variate. Semnalul vocal este parametrizat și se încearcă stabilirea unei corespondențe între valorile vectorilor acustici și fonemul pe care îl reprezintă, obținându-se o interpretare statistică a acestuia. Fonemul este cea mai mică unitate sonoră a unei limbi, iar pentru limba română, acesta reprezintă de obicei manifestarea acustică a unei litere. La decodare, pentru un semnal vocal dat, modelul acustic are rolul de a estima care sunt cele mai probabile foneme ce se regăsesc în respectivul semnal.

Modelul de limbă este antrenat folosind o bază de date de text. Rolul lui este de a găsi o reprezentare a modului în care cuvintele se succed într-o frază, astfel încât aceasta să fie validă. Principiul de bază constă în observarea frecvenței de apariție a unor secvențe de cuvinte de lungime variabilă, ca mai apoi, la decodare, să se estimeze cât de probabil este grupul de cuvinte ce reprezintă transcrierea ipotetică.

Modelul fonetic este o componentă cu rol de legătură între modelul acustic și modelul lingvistic. Acesta constă într-un dicționar fonetic, unde pentru fiecare cuvânt există corespondență cu fonemele din care este alcătuit.

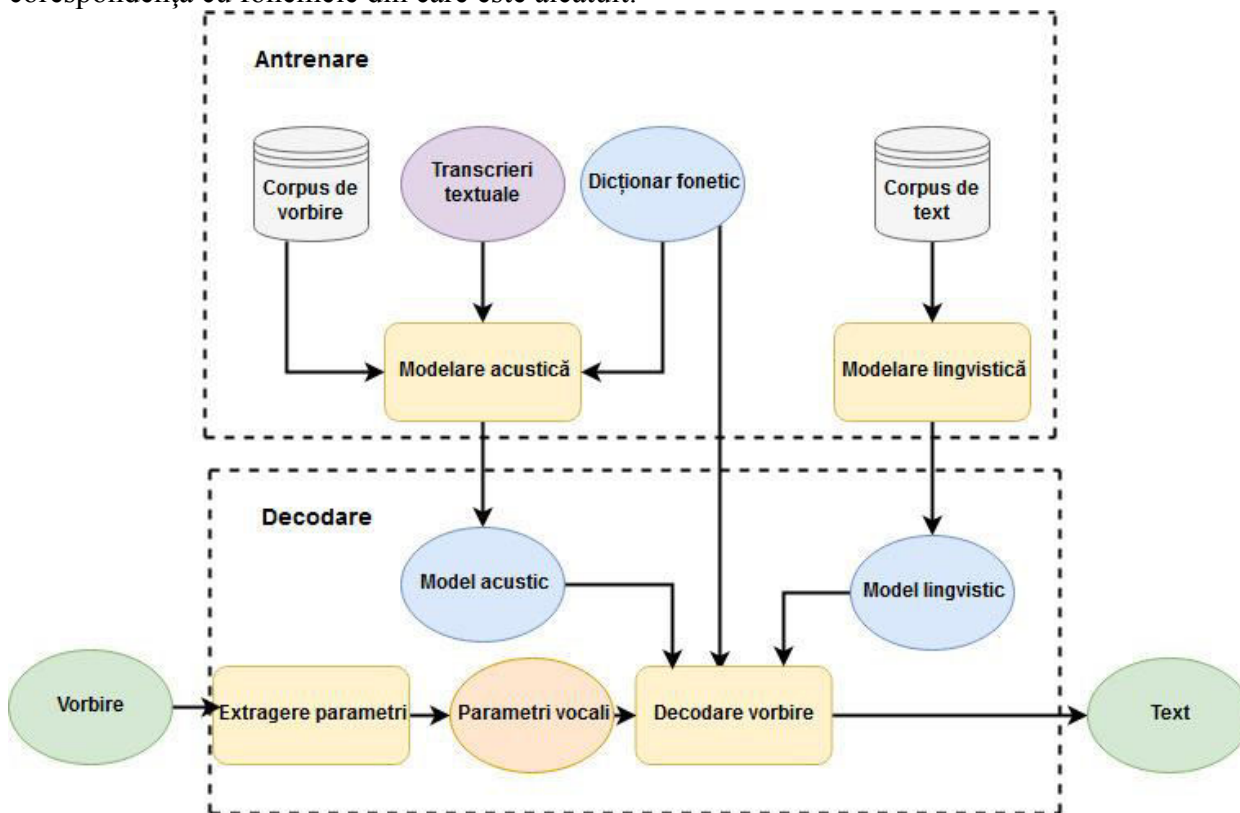


Figura 2.1 Arhitectura unui sistem RAV

Literatura de specialitate prezintă numeroase metode și tehnici de antrenare, dar în contextul acestei lucrări se poate face referire la o abordare statistică, probabilistică, și alta bazată pe rețele neuronale. Prima categorie de metode presupune calculul momentelor statistice, al

probabilităților. Acesta corespunde mixturilor de densități gaussiene (GMM) în problema modelării acustice, respectiv modelelor de tip n-gram, în cazul modelelor lingvistice. Rețelele neuronale vin cu o abordare ușor diferită. Pe baza exemplelor folosite la antrenare, știind atât intrarea cât și ieșirea sistemului, rețeaua are rolul de a găsi o funcție neliniară, care în cazul unei intrări similare cu cea de la antrenare, să fie capabilă să determine cea mai bună ieșire.

2.1 MODELARE ACUSTICĂ

2.1.1 Parametrizarea semnalului vocal

În sistemele RAV, nu se lucrează direct cu forma de undă a semnalului vocal, ci se recurge la parametrizarea sa. Înaintea oricărei alte operații, fiecare sistem are o componentă responsabilă cu preluarea semnalului, trecerea sa printr-un set de operații și obținerea unor vectori de coeficienți care evidențiază cât mai bine anumite trăsături relevante pentru recunoașterea automată a vorbirii. Cele mai robuste astfel de trăsături sunt coeficienții mel-cepstrali (MFCC). Etapele ce duc la obținerea acestora, plecând de la semnalul vocal, sunt evidențiate în Figura 2.2.

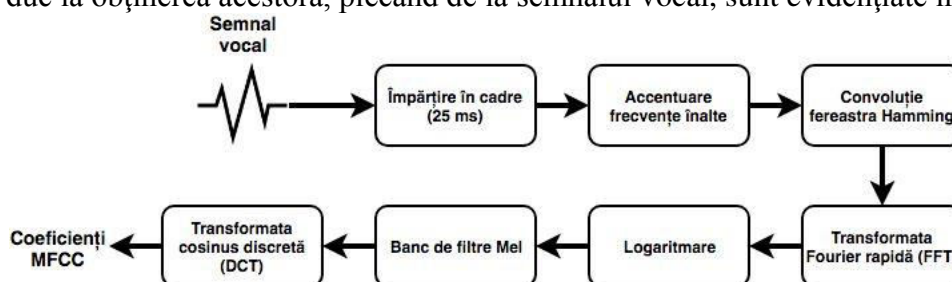


Figura 2.2 Extragerea parametrilor MFCC

Trăsăturile nu sunt extrase la nivel global, o singură dată din întreg semnalul, ci are loc o împărțire în cadre de semnal. Acest fapt este necesar deoarece vorbirea nu are un caracter staționar, parametrii caracteristici nefiind constanți în timp. Variațiile amplitudinii și frecvenței sunt imprevizibile, iar în această situație, se consideră cadre de semnal unde acesta este cvasi-staționar. Cadrele au de obicei o durată de 25 ms și o perioadă de repetiție de 10-15 ms. Scopul acestor suprapuneri este de a păstra informația ce s-ar putea pierde la zona de graniță dintre două cadre consecutive. Deoarece energia semnalului scade odată cu creșterea frecvenței, are loc o etapă de accentuare. Semnalul este trecut printr-un filtru trece-sus, iar energia la frecvențele înalte este crescută. Asupra fiecărui cadru de semnal se aplică o operație de convoluție cu o fereastră de tip trece jos (lob principal la frecvențe joase), cel mai uzual fiind folosită fereastră Hamming:

$$W[n] = \begin{cases} 0.54 - 0.46 * \cos \frac{2\pi n}{N-1}, & 0 \leq n \leq N \\ 0, & \text{în rest} \end{cases}$$

Astfel sunt atenuate discontinuitățile ce pot să apară în punctele de tăietură a cadrelor, fapt ce ar fi determinat artefacte în frecvență. Următorul pas îl reprezintă aplicarea transformatei Fourier rapide (FFT). Semnalul vocal este compus prin convoluția în timp a semnalului de excitație, reprezentat de aerul din plămâni și răspunsul în timp al filtrului reprezentat de tractul vocal. Acesta din urmă este relevant pentru recunoașterea vorbirii și pentru că cele două componente nu pot fi separate în domeniul timp, se trece în domeniul frecvență, unde convoluția devine o operație de înmulțire:

$$S(n) = E(n) * h(n) \xrightarrow{FFT} S(w) = E(w) * H(w)$$

Ulterior, prin aplicarea logaritmului, se ajunge la o operație liniară de tip sumă, ce permite separarea termenilor:

$$S(w) = E(w) * H(w) \xrightarrow{\log} \log(S(w)) = \log(E(w)) + \log(H(w))$$

Sistemul auditiv uman poate realiza o distincție mult mai bună a frecvențelor joase. În gama de frecvențe 0-1000 Hz [82] percepția este liniară, în timp ce peste acest prag, ea devine logaritmică. Se aplică un banc de filtre Mel, ce translatează dintr-o gamă de frecvență inițială mai largă, într-una mai îngustă. Parametrii vocali sunt grupați în jurul frecvențelor mai joase, reprezentarea fiind mai apropiată de modul în care funcționează sistemul auditiv uman. Ultimul pas în extragerea coeficienților MFCC îl reprezintă trecerea înapoi în domeniul timp prin aplicarea transformatei cosinus discrete (DCT). Rolul acesteia este de a reduce dimensionalitatea parametrilor și de a realiza decorelarea lor.

Dintr-un cadru de semnal se extrag de obicei vectori de coeficienți mel-cepstrali 13-dimensional. Deoarece aceștia oferă informații numai despre cadrul curent, iar uneori datele despre context sunt necesare, se calculează derivatele de ordin 1 și 2 ai coeficienților. Acestea captează informații despre tendințele semnalului, despre variația cadrelor vecine.

Un alt tip de trăsături ce sunt folosite cu preponderență pentru sarcina de recunoaștere a vorbitorilor, dar care își găsesc aplicabilitatea și în recunoașterea vorbirii, sunt iVectorii (vectori identitate) [40]. S-a plecat de la Joint Factor Analysis (JFA) [27], unde au fost definiți supervectorii, ce constau în componentele de medie ale modelului GMM ce modelează trăsăturile acustice specifice unui vorbitor. Un supervector trebuie să poată fi descompus în elemente independente de vorbitor, elemente dependente de vorbitor, elemente dependente de canal și elemente reziduale:

$$s = m + Vy + Ux + Dz,$$

Unde s este supervectorul, m este componenta independentă de vorbitor, V este componenta dependentă de vorbitor, U este componenta dependentă de canal, iar z este componenta reziduală.

Fiecărei componente îi corespund un set de factori de dimensiune scăzută, numiți factori proprii, ce corespund direcției de variație a vectorilor proprii. În acest exemplu, x și y sunt factori proprii aferenți vorbitorului, respectiv canalului.

S-a constatat însă că factorii ce modelează canalul în JFA conțin și informații despre vorbitori. Deoarece s-a ajuns la concluzia că cele două componente sunt dificil de separat, s-a recurs la o altă abordare unde acestea sunt tratate împreună. Influența canalului este redusă ulterior prin tehnici ca analiza discriminativă liniară (LDA) și Within-Class Covariance Normalization (WCCN). Astfel a apărut noțiunea de spațiu al variabilității totale. În acest spațiu, un semnal vocal este reprezentat de un vector ce cuprinde factorii spațiului total, numit iVector:

$$s = m + Tw,$$

unde m este supervectorul independent de vorbitor și canal, T este matricea variabilității totale și w este iVectorul.

În recunoașterea automată a vorbirii, iVectorii sunt uneori concatenați cu vectorii ce corespund coeficienților cepstrali. Deoarece iVectorii conțin informații ce individualizează vorbitorul, au rol în a îmbunătăți adaptarea sistemului la vorbirea specifică unui vorbitor anume.

2.1.2 Modelul Markov ascuns (HMM)

Modelul acustic estimează probabilitatea ca un semnal vocal să fi fost generat de o secvență de cuvinte. Pentru acest fapt, este necesară realizarea unor observații pe baza unui set extins de pronunții ale cuvintelor, de la cât mai mulți vorbitori, obținând apoi un model matematic ce estimează probabilitatea fiecărui cuvânt. Această premisă presupune deci alegerea cuvântului ca fiind unitate de bază în modelarea vorbirii, ceea ce se opune însă câtorva principii [30]. Unitatea de bază trebuie să fie precisă, iar reprezentarea ei să înglobeze manifestarea în mai multe contexte acustice. În același timp, trebuie să fie antrenabilă, să existe suficient de multe date pentru ca unitatea să poată fi estimată corect. Nu în ultimul rând, unitatea de bază trebuie să fie generalizabilă, fiind posibil ca orice cuvânt să fie construit dintr-o secvență de unități. Problema în cazul cuvintelor este dată de faptul că ele nu sunt generalizabile și antrenabile în aceeași măsură în care ar fi unele unități sublexicale, cum ar fi fonemele sau chiar unități sub-fonetice (senone).

Așa cum s-a specificat anterior, fonemul este cea mai mică unitate sonoră și corespunde într-o mare măsură cu manifestarea acustică a unei litere. Se poate spune atunci că fiecare cuvânt poate fi descompus într-o secvență de foneme. Având în vedere modul în care cuvintele sunt articulate, fonemele nu sunt rostite în mod independent. Două foneme identice, dar care apar în contexte diferite, având foneme vecine diferite, se vor manifesta diferit. Spunem în această situație ca fonemele sunt dependente de context, iar analiza lor merge mai departe, fiind introdusă noțiunea de trifoneme. Un fonem este compus din 3 stări, unde prima stare depinde de fonemul precedent, a doua stare este caracteristică fonemului propriu-zis, în timp ce a treia stare depinde de fonemul următor. Numărul foarte mare de trifoneme posibile face procesul de antrenare foarte dificil și intens computațional. Pentru a rezolva și această situație, fonemele sunt grupate în funcție de similaritatea lor în grupuri de stări acustice, denumite senone. Ținând cont de faptul că aceste stări sunt dependente de context, iar producerea vorbirii este dată de tranziții între stări, s-a constatat faptul că vorbirea poate fi modelată cu ajutorul modelului Markov ascuns (HMM).

Un HMM este un automat cu stări finite, unde nu este cunoscută secvența de stări, ci numai vectorii acustici corespunzători fiecărei stări, aceștia fiind generați de o funcție densitate de probabilitate.

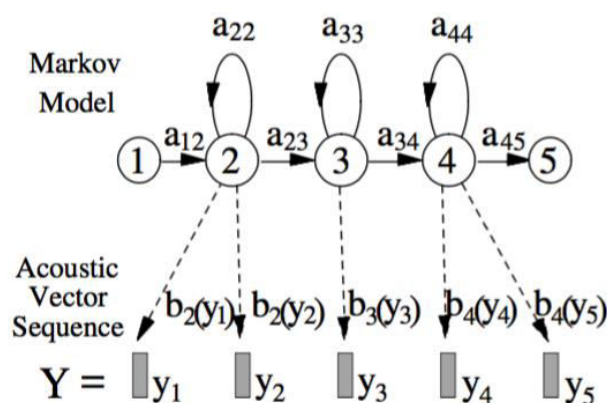


Figura 2.3 Arhitectura HMM.

Sursa: [47]

Un HMM este caracterizat de un set de stări, notate în Figura 2.33 cu numere de la 1 la 5. Fiecare stare presupune o probabilitate de tranziție către starea următoare (a_{12} , a_{23} , a_{34} , a_{45}) și o probabilitate de repetiție a stării actuale (a_{22} , a_{33} , a_{44}). Spunem despre stările 1 și 5 că sunt stări non-emisive, în timp ce stările 2, 3 și 4 sunt stări emisive, ele emit vectori de valori pe baza unei funcții de densitate de probabilitate, modelată cu ajutorul mixturilor de gaussiene sau cu rețele neuronale profunde.

Deși pe caz general, tranzițiile într-un HMM nu au constrângeri, în cazul sistemelor RAV, datorită modului secvențial în care se produce vorbirea, nu sunt permise tranzițiile înapoi și nici salturile peste stări. Fiecare stare poate avea tranziții la ea însăși sau la starea imediat următoare.

2.1.3 Mixturi de densități gaussiene (GMM)

Foarte multe evenimente pot fi descrise de funcția densitate de probabilitate gaussiană, inclusiv modelarea acustică a vorbirii:

$$p(x|\theta) = \frac{1}{\sqrt{2\pi\sigma^2}} * \exp\left(-\frac{1}{2\sigma^2} * (x - \mu)^2\right),$$

unde funcția de densitate de probabilitate este parametrizată prin $\theta = [\mu, \sigma^2]$, μ fiind media și σ^2 fiind varianța (momentele statistice de ordin 1 și 2).

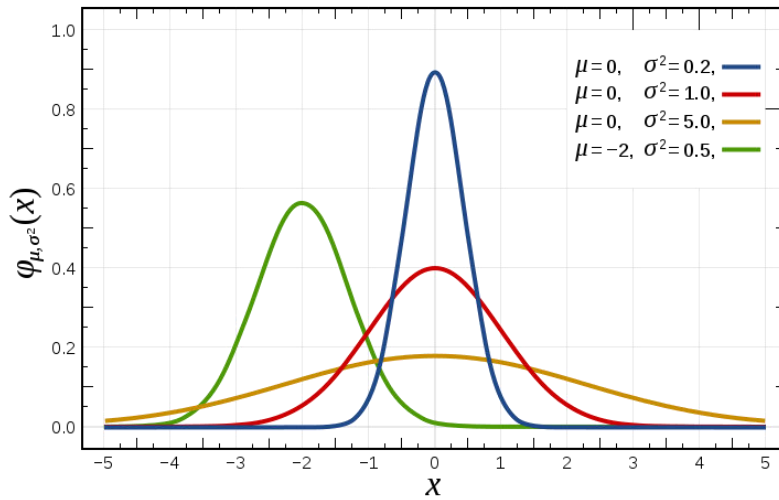


Figura 2.4 Funcția densitate de probabilitate normală.

Sursa: [48]

Funcția densitate de probabilitate normală modelează un spațiu de valori, fiind centrată în punctul cu cea mai mare probabilitate, corespunzător mediei valorilor. Dispersia este o măsură a distanței valorilor față de medie. Fiecare densitate gaussiană modelează probabilitatea ca o valoare să aparțină unei clase. Astfel, prin alegerea unui punct arbitrar din spațiul valorilor, putem determina în ce măsură acesta aparține uneia dintre clase. Estimarea raportului de plauzibilitate are ca scop determinarea clasei din care fac parte un set de observații independente: $x[0], x[1], \dots, x[N-1]$. Date fiind aceste observații, se caută cel mai bun estimat pentru θ . Pentru aceasta, se folosește funcția de plauzibilitate:

$$L(X; \theta = p(x_0, \dots, x_{N-1}; \theta) = \prod_{i=0}^N p(x_i; \theta) = \frac{1}{(2\pi\sigma^2)^{N/2}} \exp\left(-\frac{1}{2\sigma^2} * \sum_{i=0}^N (x_i - \theta)^2\right).$$

Mixturile de gaussiene (GMM) reprezintă sume ponderate ale densităților gaussiene ce încearcă să aproximeze densitățile de probabilitate ale trăsăturilor acustice. În recunoașterea automată a vorbirii, spațiul de valori modelat este format din coeficienții acustici extrași din semnal, în urma parametrizării lui. De exemplu, în cazul unei mixturi de 3 densități gaussiene, funcția ce le aproximează pe acestea este:

$$p(x) = c_1 N(x; \mu_1, \sigma_1) + c_2 N(x; \mu_2, \sigma_2) + c_3 N(x; \mu_3, \sigma_3), \sum c_i = 1.$$

Antrenarea unei mixturi de gaussiene, estimarea parametrilor ei, se realizează cu ajutorul algoritmului *Expectation-Maximization (EM)* [39]. La început, se pleacă de la o estimare inițială a parametrilor funcțiilor densitate de probabilitate. Prima fază a antrenării (E), calculează probabilitatea de apartenență a datelor de intrare la clase. Cea de-a doua fază (M), calculează parametrii modelului folosind datele actuale ale probabilităților de apartenență a datelor de intrare la clase. Algoritmul rulează în mod iterativ, iar în final converge spre maximum local al funcției de plauzibilitate.

Decodarea unui HMM reprezintă sarcina de recunoaștere a vorbirii în sine și se realizează cu algoritmul Viterbi [24]. Acesta determină care este cea mai probabilă secvență de stări acustice care a generat un set de trăsături vocale.

2.1.4 Rețele neuronale cu întârziere în timp (Time delay neural network -TDNN)

Time Delay Neural Network (TDNN) este o arhitectură de rețea neuronală ce s-a dovedit foarte eficientă în sarcinile de recunoaștere a vorbirii. Aceasta a fost introdusă de către Weibel în 1987 [2], fiind folosită la început pentru recunoașterea unui set restrâns de foneme. A obținut o performanță de 98%, pe o bază de date ce conține 5240 cuvinte izolate, adnotate la nivel de fonem [7]. Deoarece sunetele rostite nu au lungime uniformă, iar segmentarea lor precisă este foarte dificilă, TDNN reușește să rezolve acest inconvenient. Analizând momentele anterioare și posterioare producerii unui sunet, TDNN construiește un model ce conține elemente cheie ale producerii aceluși sunet. Figura 2.55 prezintă arhitectura unei rețele TDNN propusă în [16].

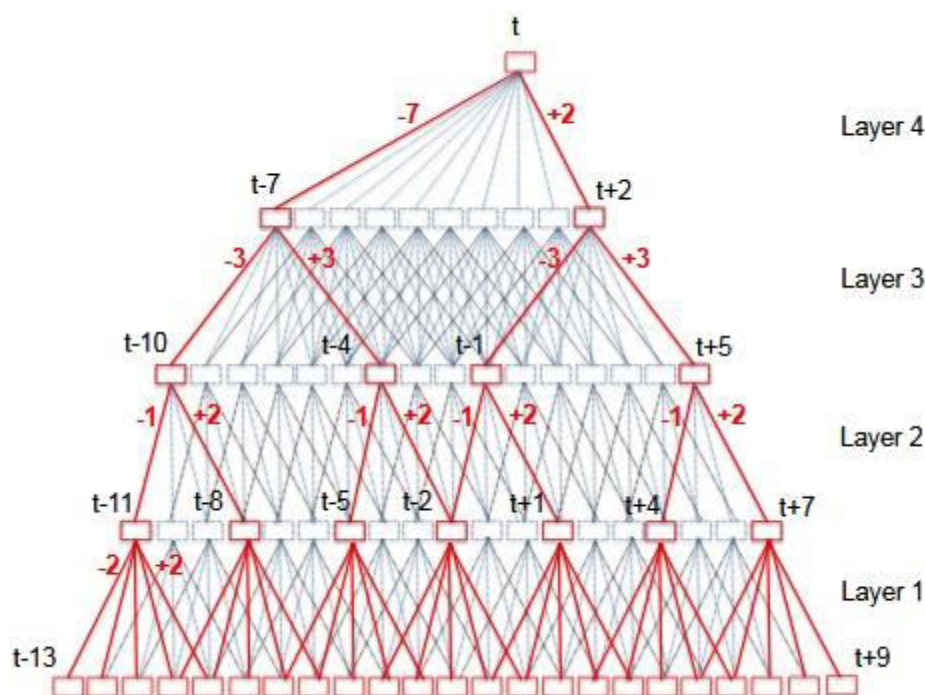


Figura 2.5 Arhitectura unei rețele TDNN.

Sursa: [16]

Principala caracteristică a rețelei este capacitatea de a modela relațiile temporale dintre cadrele de semnal. Atunci când este procesat un context temporal extins într-o rețea neuronală standard, straturile inferioare învață o transformare afină a întregului context. Într-o rețea TDNN, straturile inferioare învață contexte restrânse, în timp ce straturile superioare învață contexte temporale largi. Fiecare strat funcționează la o rezoluție temporală diferită, care crește odată cu avansarea spre straturile superioare ale rețelei.

Într-un TDNN standard se calculează funcțiile de activare pentru fiecare neuron în parte. Ferestre de câte N cadre vor reprezenta intrarea neuronilor de pe stratul următor. Apoi fereastra se mută cu un pas mai departe și alte N cadre vor fi considerate la intrarea următorului neuron de pe stratul următor. Astfel, apar suprapuneri mari între contextele de intrare la momente de timp vecine. Mergând pe presupunerea că activările vecine sunt corelate, se poate aplica o tehnică de optimizare numită subeșantionare, ce presupune că numai pentru anumiți neuroni dintr-un strat să fie calculată funcția de activare. În Figura 2.55, aplicarea subeșantionării a fost marcată folosind culoarea roșie, în timp ce cu negru avem o arhitectură standard, fără subeșantionare. Durata antrenării se reduce astfel de aproximativ 5 ori [16], comparabil cu timpul necesar rețelelor de tip *feed-forward*.

TDNN se aseamănă atât cu rețelele recurente, RNN, deoarece modelează dependențe temporale lungi între evenimentele acustice, dar și cu rețelele convoluționale, CNN, deoarece se folosesc contexte ce presupun suprapuneri ale cadrelor, similar operației de convoluție. TDNN are însă avantajul de a necesita un timp de antrenare mult mai scurt.

În Figura 2.55, pentru fiecare cadru de semnal, se consideră un context de 13 cadre la stânga sa și 9 cadre la dreapta sa. Folosind un context asimetric, cu un număr mai mare de cadre anterioare față de cele posterioare, se obține o latență mai mică la decodarea în timp real, iar acuratețea s-a dovedit a fi mai bună. Un context de 5 cadre, două la stânga și două la dreapta, sunt îmbinate și vor reprezenta intrarea fiecărui neuron din stratul secundar. Următoarele straturi vor considera la intrare numai anumiți neuroni din stratul anterior, fiind luate în calcul contextele $\{-1,2\}$, $\{-3,3\}$ și $\{-7,2\}$.

Trăsăturile de la intrarea rețelei sunt de obicei coeficienți cepstrali, MFCC, 13-dimensional, extrași din fiecare cadru de semnal. Prin aplicarea tehnicii LDA, asupra coeficienților din cadrele unui context, se reduce numărul acestora la 40. Deoarece s-a dovedit că adaptarea la vorbitor aduce îmbunătățiri în sistemele RAV, tehnicile de adaptare ca *feature-space maximum likelihood linear regression* (fMMLR) necesită decodare în doi pași, ceea ce devine dificil în sistemele ce funcționează în timp real. Astfel, la cei 40 de coeficienți cepstrali corespunzători unui cadru de semnal, se adaugă un număr de 100 iVectori, capabili să cuprindă atât informații referitoare la canalul acustic, cât și la vorbitor.

Funcția de activare folosită în această rețea este o generalizare a funcției *maxout*. Funcția *maxout* oferă la ieșire maximul funcției de activare de la un grup de neuroni:

$$y = \max_{i=1}^G(x_i).$$

Generalizarea acesteia fiind soft-maxout:

$$y = \log \sum_{i=1}^G \exp(x_i),$$

Iar funcția p-norm [1], folosită în cadrul rețelei:

$$y = \|x\|_p (\sum_i |x_i|^p)^{1/p}.$$

2.2 MODELARE LINGVISTICĂ

2.2.1 Modele statistice de tip n-gram

Modelul probabilistic (n-gram) oferă o statistică asupra modului în care grupurile de cuvinte se succed. Prin antrenarea pe baza unui volum de text foarte mare, probabilitatea de apariție a unui cuvânt este calculată în funcție de secvență de cuvinte care îl preced:

$$P(w) = P(w_1, w_2, \dots, w_n) = P(w_1) * P(w_2|w_1) * P(w_n|w_1, w_2, \dots, w_{n-1}).$$

Probabilitatea unei secvențe de cuvinte este descompusă în estimarea probabilităților unui singur cuvânt, dată fiind secvența de cuvinte anterioară acestuia. Cel mai uzual se folosesc secvențe cu istorie de două cuvinte (3-gram) sau un cuvânt (2-gram). Pentru un model 2-gram, calculul probabilității unei perechi de cuvinte se calculează în felul următor:

$$P(w_j|w_i) = \frac{\text{count}(w_i, w_j)}{\sum \text{count}(w_i, w)}.$$

Probabilitatea de apariție a perechii de cuvinte (w_i, w_j) este dată de numărul de apariții ale cuvântului w_i , urmat de cuvântul w_j , în raport cu numărul de apariții ale aceluiași cuvânt w_i , urmat de alte cuvinte.

2.2.2 Rețele neuronale recurente pentru modelarea lingvistică (RNN-LM)

Rețelele neuronale recurente (RNN) s-au dovedit a fi foarte eficiente în modelarea evenimentelor cu caracter secvențial. O astfel de sarcină este și prelucrarea limbajului natural, cuvintele dintr-o propoziție formând o înșiruire de stări ce se succed într-o anumită ordine, după anumite reguli. Modelele de limbă au rolul de a oferi o măsură a corectitudinii gramaticale și semantice pentru o propoziție ipotetică, obținută în urma decodării cu un sistem RAV.

Particularitatea principală a rețelelor RNN cu ajutorul cărora sunt antrenate modele de limbă o reprezintă modul în care informația secvențială este utilizată. În rețelele neuronale clasice, nu există neapărat o interdependență a datelor de intrare. Acest lucru nu este însă de folos în modelarea lingvistică, deoarece predicția cuvântului următor este strâns legată de cuvintele precedente. Tocmai de aceea, acest tip de rețele poartă denumirea de "recurente": aceeași sarcină este efectuată pentru fiecare componentă în parte a secvenței. Se mai spune despre rețelele RNN că au "memorie", deoarece pentru fiecare secvență este utilizată informația privind o fereastră de secvențe din trecut. Teoretic, dimensiunea contextului luat în considerare poate fi oricât de mare, dar din motive computaționale, aceasta se rezumă la câteva secvențe anterioare.

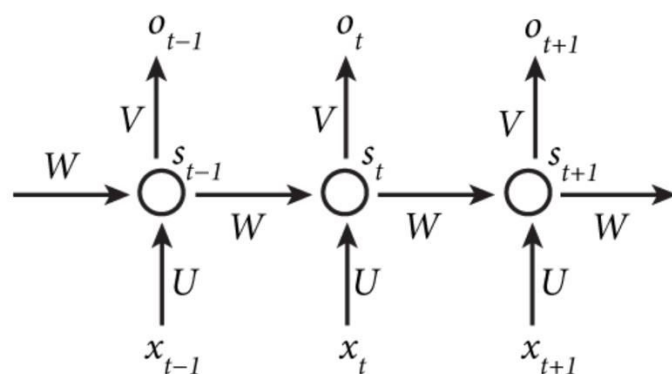


Figura 2.6 Structura unui RNN pentru modelare lingvistică

Sursa: [50]

Figura 2.6 prezintă structura unui RNN folosit la modelarea de limbă. Intrarea la momentul t este x_t și poate consta într-o abordare de tip *one-hot vector* sau *word embedding*. Prima variantă constă în codarea cuvintelor folosind vectori de dimensiunea vocabularului, care conțin valoarea 1 pe poziția corespunzătoare cuvântului în vocabular și 0 în rest. Fiecare cuvânt va fi reprezentat de un astfel de vector, ceea ce nu este deloc eficient din punct de vedere computațional. Cea de-a doua variantă implică o dimensionalitate mai restrânsă a datelor, având capacitatea de a generaliza, vectorii ce codează cuvintele cuprind informații semantice, cum ar fi câmpul lexical din care face parte cuvântul, partea de vorbire pe care o reprezintă, etc. În acest fel, ajung să fie definite relații între cuvinte. Straturile ascunse ale rețelei la momentul t sunt notate cu s_t . Acestea reprezintă memoria rețelei și sunt calculate atât pe baza straturilor ascunse de la momentul anterior de timp, cât și pe baza intrării de la momentul curent:

$$s_t = f(Ux_t + Ws_{t-1}).$$

În general, se folosesc ca funcții de activare tangenta hiperbolică ($\tanh$) sau ReLU. Ieșirea la momentul de timp t este o_t . De observat faptul că parametrii rețelei sunt aceiași la fiecare moment de timp, fiind efectuată aceeași operație, dar cu intrare diferită.

Rezultatele din literatură arată îmbunătățiri considerabile aduse de modelele de limbă antrenate cu rețele recurente în sarcina de recunoaștere automată a vorbirii. În [19] se obține o îmbunătățire a ratei de recunoaștere de 18% față de modelele de tip n-gram. Mai mult, în condițiile în care modelul n-gram a fost antrenat pe o cantitate de 5 ori mai mare de date, îmbunătățirea obținută este de 12%.

2.3 RESURSE FONETICE, ACUSTICE, LINGVISTICE PENTRU RAV

Modelul acustic nu modelează direct cuvinte, ci unități sub-lexicale (foneme) sau chiar unități sub-fonemice. Fonemul este unitatea acustică de bază dintr-o limbă. În limba română, între litere și foneme există o corespondență aproape biunivocă. Se folosesc 7 vocale de bază și două vocale împrumutate, 4 semi-vocale și 22 de consoane, așa cum se poate vedea în Tabelul 2.1 [43]. Tot în tabel se pot observa regulile de fonetizare, existând și câteva situații de ambiguitate, aceeași literă fiind transcrisă fonetic în câteva moduri diferite, în funcție de contextul în care apare. Aceste situații se regăsesc în cazul literelor: *e*, *i*, *o*, *u*, *c*, *g*, *h*, *x*. Componenta unei silabe poate să inducă ambiguitate, unele sunete pot fi vocale sau semivocale, în timp ce alte cazuri de ambiguitate sunt date de grupurile de litere *ce*, *ci*, *che*, *chi*, *ge*, *gi*, *ghe*, *ghi*.

Fonemul			Exemple de cuvinte	
Tip	Simbol IPA	Simbol intern	Forma scrisă	Forma fonetică
vocale	a	a	sat	s a t
	e	e	mare	m a r e
	i	i	lift	l i f t
	o	o	loc	l o c
	u	u	șut	s l u t
	ə	a1	gură	g u r a1
	ɨ	i2	între	i2 n t r e
vocale împrumutate	y	y	ecru	e c r y
	ø	o2	bleu	b l o2
semivocale	e ^	e1	deal	d e1 a l
	j	i3	fiară	f i3 a r a1
	o ^	o1	oase	o1 a s e
	w	w	sau	s a w
consoane	c	k2	chem	k2 e m
	b	b	bar	b a r
	p	p	păr	p a r
	k	k	acum	a k u m
	tʃ	k1	cenușă	k1 e n u s l a1
	g	g	galben	g a l b e n
	dʒ	g1	girafă	g1 r a f a1
	ʒ	g2	unghi	u n g2
	d	d	dar	d a r
	t	t	tot	t o t
	f	f	fața	f a t l a
	v	v	vapor	v a p o r
	h	h	harta	h a r t a
	ʒ	j	ajutor	a j u t o r
	ʃ	s1	coș	k o s1
	l	l	lac	l a c
	m	m	măr	m a l r
	n	n	nas	n a s
	s	s	sare	s a r e
	z	z	zar	z a r
	r	r	risc	r i s k
	ts	t1	țaran	t1 a l r a n
consoană palatalizată	j	i1	țări	t a r i1

Tabelul 2.1 Fonemele limbii române

Dicționarul fonetic este o componentă folosită atât la antrenarea unui sistem RAV, cât și la decodarea unui semnal vocal. El realizează asocierea fiecărui cuvânt din vocabular cu fonemele sale componente. Un cuvânt poate avea mai multe transcrieri fonetice, corespunzătoare mai multor moduri de pronunție. Crearea unui dicționar fonetic presupune cunoașterea regulilor de fonetizare, împreună cu excepțiile de la reguli. În cazul unui sistem RAV cu vocabular redus, fonetizarea poate fi realizată manual. În schimb, pentru sisteme RVC-VE, fonetizarea manuală este extrem de dificilă din cauza numărului foarte mare de cuvinte din vocabular. În această situație, devine necesară utilizarea unei aplicații în care să existe regulile deja implementate, iar procesul să decurgă în mod automat.

În contextul unui sistem RAV, resursele acustice sunt folosite la antrenarea modelului acustic, dar și la evaluarea sistemului. Acestea constau într-un corpus de voce, împreună cu adnotările aferente. Corpusul este format din fișiere audio cu înregistrări de vorbire, în timp ce adnotările reprezintă fișiere text ce conțin transcrierile fișierelor audio. Adnotarea se face la nivel de cuvânt și se poate realiza manual, prin ascultarea înregistrării audio și scrierea textului corespunzător. Există și varianta adnotării automate, folosind metodele prezentate pe larg în Capitolul 3. Crearea unor înregistrări audio plecând de la citirea unui text deja existent este o altă posibilitate de a obține baze de date cu vorbire adnotată. De obicei, în sistemele RAV, se folosesc fișiere audio în format wav, eșantionate la frecvența de 16 KHz, 16 biți pe eșantion.

Pentru obținerea unui sistem cât mai performant, baza de date de vorbire trebuie să cuprindă multe surse de variabilitate. Dimensiunea bazei de date trebuie să fie cât mai mare, cu fișiere de la cât mai mulți vorbitori. Se recomandă ca vorbitorii să fie cât mai variați, atât masculini cât și feminini, de vârste diferite, provenind de pe întreg arealul în care se vorbește limba respectivă. De asemenea, stilul vorbirii trebuie să cuprindă atât vorbire citită, cât și vorbire spontană. Înregistrările pot fi în condiții de liniște, dar și în situații unde există zgomot de fundal. Scopul final este obținerea unei variabilități acustice foarte mari, existând cât mai multe tipuri de pronunții pentru aceleași fonem. Desigur, toate aceste caracteristici ale bazei de date de vorbire depind și de sarcina sistemului RAV. De exemplu, ele sunt necesare în cazul unui sistem de dictare, cu vocabular extins, unde vorbirea este independentă de vorbitor, în orice fel de context. Există însă și sisteme RAV ce își propun transcrierea unor rostiri cu vocabular restrâns, numai dintr-un domeniu anume, dependent de vorbitor, adaptat la stilul de vorbire a câtorva persoane. O analiză a resurselor acustice disponibile pentru limba română se găsește în secțiunea 3.1.

Resursele lingvistice sunt necesare la antrenarea modelului de limbă. Crucială este dimensiunea corpusului de text folosit. Pentru o sarcină de recunoaștere cu vocabular extins este nevoie ca în modelul de limbă să se regăsească toate cuvintele limbii respective, în diferite structuri sintactice, pentru a crește capacitatea de predicție. Procurarea unui astfel de corpus de text de dimensiuni mari se poate face în mod automat, fiind extras din conținutul ce se regăsește în mediul online. Astfel de conținut este reprezentat în special de platformele de știri. Textul descărcat trebuie să treacă printr-o operație de curățare, fiind eliminate toate celelalte caractere ce nu reprezintă litere aflate în componență cuvintelor. De asemenea, dacă se dorește obținerea de text simplu la ieșirea sistemului RAV, se vor înlocui majusculile, se vor elimina semnele de punctuație și alte simboluri speciale, precum și cifrele. Aceste operații sunt necesare deoarece, prin definiție, modelul de limbă estimează numai probabilitatea de apariție a cuvintelor și a secvențelor de cuvinte.

2.4 EVALUAREA RAV

Pentru a evalua un sistem RAV este necesară o bază de date de vorbire de evaluare, pentru care există transcrierea corespunzătoare, având certitudinea că aceasta este corectă. Baza de date este oferită la intrarea sistemului, obținându-se o transcriere ipotetică. Având transcrierea reală, de referință, și transcrierea realizată de sistemul RAV, se poate calcula în ce măsură acesta a transcris corect sau nu. În acest sens, au fost definite câteva metrici de evaluare.

Rata de eroare la nivel de caracter (ChER) folosește caracterul ca unitate de bază pentru comparație. Aceasta măsoară numărul de caractere eronate din totalul caracterelor din text. Caracterele pot fi greșite din 3 motive: inserții (au fost adăugate caractere în plus), substituții (apariția altor caractere decât cele corecte) și ștergeri (lipsa unor caractere):

$$ChER[\%] = \frac{\#Er.inserțiecaract. + \#Er.substitutiecaract. + \#Er.stergerecaract.}{\#Totalcaractereinreferinta} * 100.$$

Rata de eroare la nivel de cuvânt (WER) se calculează asemănător cu cea la nivel de caracter, unitatea de bază comparată fiind cuvântul. Similar, pot exista erori de inserție, substituție și ștergere:

$$ChER[\%] = \frac{\#Er.insertiecuv. + \#Er.substitutiecuv. + \#Er.stergerecuv.}{\#Total\ cuvinte\ in\ referinta} * 100.$$

Rata de eroare la nivel de propoziție (SER) reprezintă raportul dintre numărul de propoziții transcrise greșit și numărul total de propoziții. Dacă o propoziție conține cel puțin un cuvânt greșit, atunci ea este considerată greșită:

$$SER[\%] = \frac{\#Propozitii\ eronate}{\#Total\ propozitii\ in\ referinta} * 100.$$

În funcție de sarcina de recunoaștere, fiecare dintre aceste rate pot fi mai concludente decât celelalte. Acolo unde un caracter greșit anulează valabilitatea întregului cuvânt, ChER este cea mai potrivită metrică ce ar trebui luată în considerare. Similar, dacă un cuvânt greșit anulează valabilitatea întregii propoziții, WER oferă o bună estimare a performanțelor sistemului.

De asemenea, modelele de limbă pot fi și ele evaluate, existând două metrici în acest sens: perplexitatea (PPL) și rata de apariție a cuvintelor necunoscute (OOV). Existența unui corpus de text de evaluare este necesară pentru această sarcină. Corpusul poate fi similar cu cel pe baza căruia s-a realizat antrenarea modelului sau poate fi diferit.

Perplexitatea măsoară cât de bine poate face modelul preziceri asupra textelor. Intuitiv, reprezintă câte opțiuni alternative sunt luate în calcul pentru fiecare cuvânt; cu cât mai puține, cu atât mai bine. Se mai poate spune că este o măsură al gradului de "surprindere" al modelului în fața unei secvențe de cuvinte a cărei probabilitate trebuie estimată. Perplexitatea secvenței $w_1, \dots, w_N$ este calculată ca fiind:

$$PPL = 2^H = 2^{\sum_{i=1}^N -\frac{1}{N} \log_2 P(w_i | w_1, \dots, w_{i-1})},$$

unde H reprezintă cross-entropia.

Uneori, corpusul de evaluare este posibil să conțină unele cuvinte care nu se regăsesc în vocabularul sistemului RAV și implicit nici în modelul de limbă. Aceste cuvinte nu pot fi transcrise de sistemul RAV și poartă denumirea de cuvinte în afara vocabularului (*Out of Vocabulary* - OOV). Probabilitatea lor de apariție este nulă, în timp ce perplexitatea lor este infinită.

Este de dorit ca atât perplexitatea, cât și rata cuvintelor din afară vocabularului, să aibă valori cât mai scăzute.

CAPITOLUL 3

METODE DE ANTRENARE NESUPERVIZATĂ A MODELELOR ACUSTICE PENTRU RAV

Pentru dezvoltarea unui sistem de recunoaștere automată a vorbirii, resursa principală o reprezintă un corpus de vorbire pentru care există transcrierea corespunzătoare. Crearea unui model acustic puternic presupune antrenarea pe baza unui corpus de dimensiune cât mai mare, ce conține pronunții cât mai variate, provenite de la cât mai mulți vorbitori. Uneori, aceste resurse audio adnotate lipsesc sau sunt dificil de procurat. Totuși, odată cu creșterea puterii computaționale și a capacității mediilor de stocare, interesul s-a mutat către corpusurile audio disponibile în online și mass-media. De exemplu, emisiuni de radio și televiziune, înregistrări ale ședințelor publice din cadrul anumitor instituții, lecturi, toate acestea sunt surse ușor accesibile și bogate în vorbire din viața reală, dar neadnotată.

Transcrierea unui astfel de corpus nu este deloc o sarcină trivială. Aceasta poate fi realizată în mod manual, prin ascultarea înregistrărilor, însoțită de scrierea simultană a textului aferent. Efortul în acest caz este considerabil, atât din punct de vedere al timpului necesar, cât și al costului. Mai mult, în cazul unor limbi slab dezvoltate, găsirea unor vorbitori nativi care să realizeze acest proces de adnotare manuală reprezintă în sine o problemă.

O altă modalitate de a obține un corpus de vorbire transcrisă, de data aceasta în mod automat, presupune chiar utilizarea unor sisteme de recunoaștere automată a vorbirii deja existente. Deoarece niciun astfel de sistem nu este perfect, iar transcrierile obținute conțin erori, un mod foarte întâlnit de tratare a acestei probleme presupune utilizarea concomitentă a mai multor sisteme. Figura 3.1 prezintă schema de principiu a unei astfel de abordări.

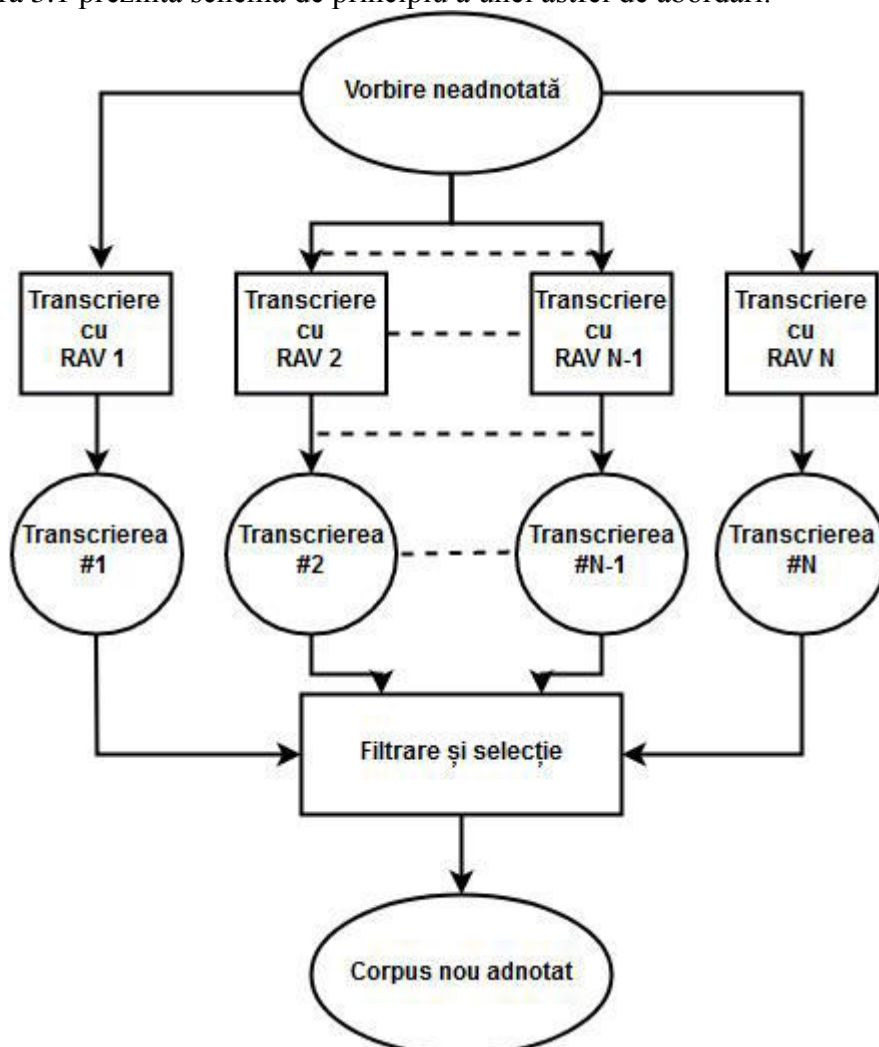


Figura 3.1 Adnotarea automată a corpurilor audio

Transcrierile corpusului audio de la cele N sisteme trec printr-un proces de filtrare și selecție. Această etapă are sens numai în situația în care sistemele sunt diferite, iar erorile făcute de acestea sunt, de asemenea, distincte. Plecând de la această premisă, sunt considerate corecte părțile transcrise identic de către toate sistemele. Altfel, dacă cele N sisteme ar face erori simetrice, este echivalent cu utilizarea unui singur sistem, duplicat de N ori. În concluzie, o astfel de abordare este utilă numai dacă sunt folosite sisteme RAV complementare.

Se poate spune despre două sau mai multe sisteme RAV că sunt complementare, dacă pentru aceeași secvență de vorbire, transcrierile obținute vor conține erori diferite.

În capitolele următoare vor fi prezentate principalele metode prin care se pot obține transcrieri complementare, împreună cu modul în care se realizează filtrarea acestora, în vederea adnotării cât mai precise a unor corpusuri de vorbire.

3.1 PROBLEMA RESURSELOR PENTRU LIMBI SLAB DOTATE

Ultimii ani au adus un avans major în ceea ce privește procesarea semnalului audio, în special dezvoltarea sistemelor de recunoaștere automată a vorbirii și a vorbitorului. Deși metodele și algoritmi trec prin procesul continuu de îmbunătățire, o mare problemă este reprezentată de lipsa

bazelor de date de vorbire, în special pentru limbile cu resurse limitate, așa cum este cazul limbii române. Din câte știm, există relativ puține corpusuri de vorbire în limba română cu vorbire adnotată. Potrivit [13], limba română este a doua cea mai slab dotată limbă dintr-un grup de 5 posibile, în ceea ce privesc resursele de text și vorbire. Totuși există câteva baze de date de vorbire în limba română, unele dintre ele fiind distribuite public.

Un sumar cu cele mai importante baze de date de vorbire în limba română este prezentat în Tabelul 3.1. Așa cum se observă din tabel, cele mai mari corpusuri sunt cele prezentate în [37, 38, 39, 40]. Există și câteva corpusuri mai mici pentru care sunt oferite, de asemenea, detalii.

Achiziția și adnotarea unuia dintre primele corpusuri de vorbire continuă în limba română este prezentat în [21], mai mult de 10 ore fiind înregistrate de către 100 vorbitori. Această bază de date are o structură similară cu corpusul EUROM-1 în limba engleză.

Corpusul SWARA [10] conține 21 ore de vorbire de înaltă calitate, de la 17 vorbitori. A fost adnotat manual la nivel de rostire și semiautomat la nivel de fonem. Scopul principal al corpusului este de a da posibilitatea persoanelor cu deficiențe de vorbire sau care suferă de afonie să folosească o voce sintetizată cât mai apropiată de vocea lor.

În [6] sunt prezentate sisteme RAV ce au fost create folosind buletine de știri transmise online în câteva limbi est-europene. Modelul acustic în limba română a fost antrenat folosind un corpus de vorbire de 31 de ore, adnotat manual.

În [37] este prezentat un corpus de 40 ore ce conține înregistrări ale unor conversații, bazate pe 25 de scenarii ce au legătură cu domeniul bancar.

În [36] se prezintă ultimele noutăți despre expandarea corpusul de vorbire folosit la antrenarea și evaluarea sistemelor RAV ale grupului Speed: două mari seturi de vorbire citită și spontană, ambele având peste 100 ore.

RoDigits [35] este un corpus de vorbire în limba română, compus din clipuri audio de la mai mulți vorbitori, fiecare clip fiind format dintr-o secvență rostită de cifre conectate.

Nume & ref.	Tipul vorbirii	Domeniu	Dimensiuni			Disponibilitate
			#frază	#ore	#vorbitori	
RASC [8]	Citită	articole Wikipedia	3k	4.8	N/A	public
RO-GRID [34]	Citită	General	4.8k	6.6	12	public
IIT [46]	Citită	Literatură	N/A	0.8	3	non-public
n/a [21]	Citită	Traduceri adaptate după Eurom-1	4k	10.0	100	non-public
n/a [5]	Spontană	Internet, TV	N/A	4.0	12	non-public
RSS [11]	Citită	Știri, Literatură	4k	4.0	1	public
SWARA [10]	Citită	Ziare	19k	21.0	17	public
n/a [6]	Spontană	Știri	n/a	31.0	N/A	non-public
n/a [37]	Spontană	Domeniul bancar	N/A	40.0	30	non-public
RSC [36]	Citită	Știri, interviuri, Literatură	147k	105	157	non-public
SSC [36]	Spontană	Emisiuni de radio, TV	227k	135	N/A	non-public
Rodigits [35]	Citită	Secvențe de cifre	15k	38	154	public

Tabelul 3.1 Resurse de vorbire în limba română

Așa cum se poate vedea din acest sumar, nu există o gamă foarte variată de resurse de vorbire adnotată pentru limba română și numai unele dintre ele sunt disponibile gratuit. Alte limbi, cum ar fi limba engleză, dispun de o cantitate de date mult mai mare. De exemplu, Switchboard [33] este un corpus de vorbire conversațională pe linie telefonică și conține 300 ore de vorbire. Librispeech [17], un corpus de vorbire citită în limba engleză de scară largă, are o durată de peste 1.000 ore.

3.2 METODE DE OBTINERE A COMPLEMENTARITĂȚII RAV

Obiectul acestui studiu este analiza complementarității sistemelor RAV, fiind dată de diferențe la nivelul:

- seturilor de date de antrenare;
- trăsăturilor acustice;
- algoritmilor de antrenare;
- algoritmilor de decodare;
- tipul modelului acustic.

Seturile de date de antrenare diferite (acustice/lingvistice) reprezintă un prim factor ce conduce la obținerea unor sisteme RAV complementare. Distincția dintre corpusuri poate fi dată de mai mulți factori:

- tipul vorbirii - citită sau spontană;
- mediul ambiant - în condiții de liniște sau zgomot;
- domeniul vorbirii - vorbire generală sau aparținând unei anumite sfere de activitate;
- modul de împărțire al corpusului în subseturi;
- genul vorbitorilor (masculin/feminin);
- limba utilizată.

Acești factori au un impact direct asupra diversității modelului acustic. Modelele acustice antrenate pe date care diferă în acest mod sunt complementare deoarece fiecare surprinde manifestări acustice diferite ale fonemelor. În [26] și [25] sunt utilizate două sisteme RAV complementare, antrenate pe seturi de date diferite. Scopul a fost transcrierea automată a unui corpus de vorbire achiziționat din transmisiuni de televiziune. În [31] și [44], ambele modele acustice sunt de același tip, dar ele au fost antrenate cu subseturi diferite din corpusul de antrenare: vorbire citită și vorbire spontană. Un alt element ce poate oferi transcrieri complementare în faza de decodare este modelul de limbă aplicat. În [23] se folosesc atât modele de limbă antrenate pe corpusuri generale de text, cât și modele de limbă antrenate pe text provenind dintr-un domeniu specific. Limba pentru care a fost antrenat modelul este de asemenea un element important, deoarece nu toate limbile folosesc același set de foneme, iar în cazul fonemelor ce se regăsesc în mai multe limbi, unele sunt perfect identice, în timp ce în cazul altora, se pot face asocieri de similitudine. Astfel de abordări, în care se folosesc modele acustice antrenate pe limbi diferite, se regăsesc în [3], [4], [18] și [64].

Trăsăturile extrase din semnalul vocal sunt un alt element ce duce la obținere de sisteme RAV complementare. Așa cum este cunoscut, în sistemele RAV, nu se lucrează cu formă de undă brută, ci semnalul este parametrizat. Deși mai toate metodele de extracție a parametrilor vizează păstrarea informației utile, cu rol discriminatoriu în producerea fonemelor, iar majoritatea operațiilor efectuate sunt comune, există mai multe tipuri de trăsături obținute din semnal. De exemplu, în [23] se folosesc atât coeficienți mel-cepstrali (MFCC), cât și coeficienți percepțuali de predicție liniară (PLP). Autorii utilizează în [65] atât coeficienți clasici, de tip MFCC și PLP, cât și coeficienți extrași prin metoda numită *pitch-synchronous analysis*, asupra cărora a fost aplicată normarea lungimii tractului vocal (VTLN). Mai mult, au fost încercate și diferite combinații între aceste trăsături. Cea mai simplă formă de combinare o reprezintă concatenarea vectorilor de trăsături acustice. Această operație se realizează prin aplicarea analizei discriminatorii liniare heteroscedastice (HLDA). Metoda este însă nefezabilă deoarece dimensiunea spațiului ce trebuie modelat este crescută. Reducerea dimensionalității se realizează

prin aplicarea unor tehnici ca analiza componentelor principale (PCA) sau analiza discriminatorie liniară (LDA). În [58] se folosesc două sisteme RAV complementare, ce diferă atât prin tipul modelului acustic, cât și prin tipul trăsăturilor vocale. Unul dintre sisteme este bazat pe coeficienți PLP, 13-dimensional, asupra cărora se efectuează o operație de normare a mediei și variantei (CMVN), în timp ce al doilea este bazat pe coeficienți mel 40-dimensional, extrași cu bancuri de filtre. Aceleași tipuri de trăsături, împreună cu MFCC și combinații între acestea, sunt utilizate în [53].

Tipurile diferite de modele utilizate în sistemele RAV pot reprezenta surse de complementaritate. Astfel, în [20], sunt utilizate modele acustice de tip GMM sau subspații de mixturi gaussiene (SGMM), antrenate în mod convențional, plecând de la vectori de trăsături acustice, sau prin inițializare cu parametri obținuți dintr-un model preantrenat de vorbire generală (UBM). Complementaritatea este dată în [58] de utilizarea modelelor acustice de tip GMM, respectiv modele acustice antrenate cu rețele neuronale profunde (DNN). Aceleași tipuri de modele acustice sunt folosite și în [29]. Autorii prezintă în [53] folosirea mai multor tipuri de modele acustice: SGMM, DNN sau rețele neuronale convoluționale (CNN-DNN). Tot rețele neuronale în diferite configurații sunt utilizate și în [66].

Algoritmii de antrenare ai modelelor fac ca acestea să fie complementare, chiar dacă modelele funcționează pe baza acelorași principii, fiind de același tip. De exemplu, în cazul modelelor HMM-GMM, criteriul de optimizare la antrenare poate fi de tip generativ sau discriminativ. Clasificatorii generativi încearcă să învețe modelul care generează datele, calculând probabilitățile și distribuțiile modelului. În cazul clasificatorilor discriminativi, probabilitățile posterioare ale stărilor sunt modelate direct, astfel încât discriminarea între foneme se face cât mai bine. În [53] se folosesc atât modele SGMM, ce sunt de tip generativ, cât și FBMMI, de tip discriminativ.

Algoritmii de decodare ai semnalului vocal sunt un alt factor ce conduce la obținerea de sisteme complementare. În [62], decodarea este realizată atât în ordine normală, cât și în ordine inversă. Pentru cel de-al doilea caz, s-a recurs la inversarea cadrelor de semnal de la intrarea sistemului, împreună cu inversarea cuvintelor din modelul de limbă și din modelul fonetic. Un element ce poate oferi transcrieri complementare în faza de decodare este modelul de limbă aplicat. Decodarea pot fi de asemenea directă, folosind un model de limbă pentru a obține graful cu transcrieri alternative (latice) sau decodare urmată de reevaluarea lăței, folosind tehnica numită *rescoring* [66].

3.3 ÎMBINAREA ȘI APLICAREA METODELOR DE COMPLEMENTARITATE

În articolul [23] se utilizează diversitatea la nivelul trăsăturilor acustice, tipurilor de vorbitori (masculini/feminini) și a seturilor de date de antrenare pentru modelul de limbă. Se folosesc mai multe sisteme RAV pentru limba slovacă, antrenate pe aceleași baze de date, folosind aceeași paradigmă, platforma HMM-GMM. Fiecare model conține 3 stări pentru HMM și 32 densități gaussiene. Complementaritatea este dată de tipul trăsăturilor vocale utilizate, cât și de datele cu care au fost antrenate modelele de limbă. Se folosesc coeficienți MFCC și PLP, împreună cu derivatele lor de ordin 1 și 2, iar în final a fost aplicată o metodă de normare a mediei acestora. Trei modele acustice au fost antrenate folosind trăsături de tip MFCC, câte unul pentru fiecare gen (masculin, feminin) și unul general. În schimb, un singur model acustic a fost antrenat folosind PLP, folosind întreg setul de date de antrenare. Două modele de limbă au fost utilizate, ambele fiind de tip n-gram. Unul dintre ele a fost antrenat cu text provenind dintr-un domeniu specific, în timp ce al doilea a fost antrenat cu text general. Sistemele RAV finale au fost obținute prin combinarea fiecărui model acustic cu fiecare model de limbă, fiind obținute în total 8 sisteme.

În [65] se combină mai multe tipuri de trăsături pentru a obține sisteme RAV complementare. Modelele acustice au fost antrenate folosind platforma HMM-GMM. Modelul de bază a fost obținut folosind coeficienți MFCC, împreună cu derivatele de ordin 1 și 2. Câte 16 mixturi gaussiene au fost utilizate pentru a modela stările acustice. Alte modele au folosit coeficienți MFCC bazați pe reprezentarea STRAIGHT [28] combinări între vectorii cu coeficienți MFCC standard și MFCC STRAIGHT, coeficienți de tip PLP, coeficienți PLP bazați pe reprezentarea STRAIGHT, combinări între vectorii cu coeficienți PLP standard și PLP STRAIGHT. Alte variațiuni cuprind acești coeficienți în combinație cu aplicarea tehnicii VTLN.

În [20], complementaritatea constă în utilizarea unor algoritmi diferiți pentru antrenarea modelelor, împreună cu varierea numărului total de stări acustice. Toate modelele acustice sunt modelate cu HMM-uri cu 3 stări, dependente de context, fiecare stare fiind modelată de 16 densități gaussiene. Se pleacă de la două sisteme inițiale, ale căror modele acustice sunt de tip GMM, SGMM și SGMM inițializat cu parametri deja antrenați într-un model UBM. Modelele inițiale sunt antrenate cu puține date, ele fiind reantrenate prin adăugarea la corpusul de antrenare a datelor selectate. Trăsăturile acustice extrase sunt de tip PLP, împreună cu derivatele lor de ordin 1 și 2.

În [58] se utilizează sisteme complementare atât din punct de vedere al tipului modelelor, cât și din punct de vedere al trăsăturilor acustice. Este antrenat un model acustic de tip HMM-GMM, ce folosește vectori de trăsături de tip PLP, 39-dimensional, asupra cărora se aplică tehnici de normare a mediei (CMN) și varianței (CVN). Modelul de tip DNN-HMM folosește trăsături de tip *filterbank* împreună cu derivatele lor, 40-dimensionale, luând în calcul câte 5 cadre de semnal la stânga și la dreapta cadrului curent. Stratul de intrare conține 1320 neuroni, stratul de ieșire conține 3000 neuroni, iar straturile ascunse, în număr de 7, conțin 1024 neuroni fiecare.

În [26] și [25] sunt folosite două sisteme RAV, cu modele acustice de tip HMM-GMM, complementaritatea lor fiind asigurată de seturile diferite de date cu care au fost antrenate.

În [31] diversitatea celor două modele acustice este dată de modul de împărțire al setului de date de antrenare, respectiv tipul vorbirii din aceste seturi. Astfel, unul dintre sisteme a fost creat folosind un subset cu vorbire citită, în condiții de liniște, în timp ce al doilea se bazează pe vorbire spontană, uneori în condiții de zgomot.

În [62] se folosește ca sursă de diversitate doi algoritmi diferiți de decodare, aplicați în cadrul aceluiași sistem RAV. Astfel, decodarea semnalului vocal are loc în ordine normală și inversă. Modelul acustic a fost antrenat pe baza platformei HMM-GMM, folosind coeficienți MFCC. S-a utilizat același model fonetic și același model lingvistic, dar pentru situația când decodarea are loc în ordine inversă, aceste modele au fost inversate și ele, fiind scrise invers.

În [53], diversitatea se manifestă la nivelul tipului trăsăturilor extrase din semnalul vocal, tipul modelelor acustice și tipul algoritmilor de antrenare. Au fost utilizate 4 tipuri de trăsături acustice: coeficienți mel-cepstrali (MFCC), cepstrul liniar predictiv (PLP), coeficienți extrași cu ajutorul bancurilor logaritmice de filtre Mel (FBANK) și coeficienți (FBANK) împreună cu trăsături tonale. Din punctul de vedere al modelelor acustice, au fost utilizate atât modele bazate pe HMM-GMM, cât și pe rețele neuronale de tip DNN sau CNN-DNN. Algoritmii de antrenare folosiți au fost atât generativi, SGMM, cât și discriminativi, FBMMI. Pentru antrenarea acestora s-au utilizat alinierile date de un model simplu, bazat pe trifoneme dependente de context, antrenat cu HMM-GMM. Modelarea de limbă s-a făcut atât probabilistic, folosind modele de tip n-gram, cât și pe bază de rețele neuronale recurente (RNN). Modelele n-gram au fost utilizate în două faze: un model general pentru faza de decodare și un model adaptat la domeniu pentru faza de rafinare a textului transcris.

Următoarele 4 articole au utilizat ca metodă de diversitate decodarea cu modele acustice antrenate pe alte limbi, diferite de cea a corpusului decodat. Pentru această sarcină, fonemele din fiecare limbă pentru care există model acustic, au fost asociate la fonemele limbii țintă. În [3] s-au folosit modele acustice în limbile bulgară, rusă, poloneză și croată pentru a decoda o bază de date de vorbire în limba cehă. Transcrierile astfel obținute au fost utilizate mai departe la antrenarea unui sistem RAV în limba cehă. Scopul în [4] este decodarea și crearea unui sistem

RAV pentru limba cehă, folosind de data aceasta modele acustice pentru limbile engleză, franceză, germană și spaniolă. În [18] se utilizează sisteme RAV pentru limbile engleză, franceză și rusă pentru a obține un model acustic în limba letonă. Au fost folosite aproximativ 800 ore de vorbire neadnotată din transmisiuni online. Modelul de limbă a fost antrenat folosind corpus de text adunat online, împreună cu transcrieri ale ședințelor din Parlament. S-au utilizat atât modele probabilistice, de tip n -gram, cât și modele bazate pe rețele neuronale. În [64] se dorește antrenarea nesupervizată a unui sistem RAV pentru limba japoneză, folosind la decodarea corpusului neadnotat, modele antrenate pe limbile mandarină, engleză și coreeană.

În [32], complementaritatea sistemelor este realizată în special la nivelul trăsăturilor acustice. Cele 4 sisteme RAV utilizează coeficienți cepstrali, la care se adaugă diferite transformări: MFCC + CMVN + VTLN + LDA (1), + SAT + MPE (2), + MLLR (3), + LM rescoring (4). Cel de-al patrulea sistem utilizează în plus față de cel precedent tehnica de *rescoring* pentru a îmbunătăți transcrierile inițiale. A fost antrenat un model acustic de tip HMM-GMM pe o bază de date de 85 ore de vorbire în limba engleză, adnotată manual. Un corpus de vorbire neadnotată, numit European Parliament Plenary Sessions (EPPS), ce are o durată de 180 ore, a fost transcris folosind sistemele complementare.

În [29], diversitatea sistemelor este dată de tipul modelelor acustice folosite, HMM-GMM, dar și HMM-DNN, împreună cu tipul algoritmului folosit la antrenarea modelului, generativ (*MLE - Maximum likelihood Estimation*) sau discriminativ (*fMPE+bMMI - boosted maximum mutual information*).

3.4 METODE DE SELECȚIE

Selecția datelor este ultima etapă din cadrul unui proces de adnotare automată a corpusurilor de vorbire. În urma folosirii unor metode ce asigură complementaritatea, sunt create mai multe sisteme RAV cu care este transcrisă vorbirea neadnotată, obținându-se mai multe transcrieri alternative. Scopul selecției este de a combina toate informațiile disponibile pentru a rezulta în final cea mai bună transcriere pentru un segment audio. Se urmăresc două aspecte: o precizie cât mai bună, astfel încât transcrierea stabilită să fie cea corectă și obținerea de transcrieri pentru o cantitate cât mai mare din totalul datelor audio neadnotate. De asemenea, în funcție de caz, există un compromis între aceste aspecte, astfel încât se dorește o transcriere pentru o parte din corpus sau pentru întreg corpusul, respectiv o încredere cât mai mare sau o încredere mai mică pentru transcrierea obținută.

Alinierea completă a două sau mai multe ipoteze obținute în urma procesului de decodare cu sisteme RAV reprezintă o primă și cea mai simplă metodă de selecție. Această aliniere presupune identificarea segmentelor identice și considerarea lor ca fiind corecte, având în vedere premisa că sistemele complementare fac erori diferite. În [31], selecția se face în acest fel, utilizând algoritmul *Dynamic Time Warping (DTW)*. De asemenea, metoda este întâlnită în [26] și în [25].

Scorul de încredere al cuvintelor permite filtrarea părților selectate ca fiind comune în urma alinierii. Aceasta este o metodă similară cu metoda alinierii complete, dar mai eficientă. De obicei, sistemele RAV, scot la ieșire un graf de transcrieri alternative, denumit latice, unde fiecare nod este un cuvânt, iar fiecare arc este o tranziție între stări, cu o anumită probabilitate. Transcrierea finală este cea mai bună cale prin acest graf, calea cu cele mai mari probabilități, acestea purtând numele de scor de încredere. Un prag foarte jos pentru scorurile de încredere va permite acceptarea unor transcrieri eronate, în timp ce un prag foarte ridicat este posibil să ignore cuvinte ce sunt corecte. O astfel de abordare este întâlnită în [20], iar în [26] este aplicată o constrângere, ce vizează numărul minim de cuvinte dintr-o secvență selectată. Similar se regăsește și în [62], unde pe lângă constrângerea precedentă, se cere și o durată minimă de non-vorbire între secvențele selectate. În [32] folosește aliniere și praguirea scorului de încredere la nivel de cuvânt și la nivel de stare acustică.

ROVER [38] (*Recognizer Output Voting Error Reduction*) este un sistem ce are ca scop combinarea mai multor ipoteze rezultate în urma procesului de transcriere, astfel încât se obține o singură ipoteză, cât mai corectă. ROVER utilizează programarea dinamică pentru a construi un graf de tranziții între cuvinte (WTN - *word transition network*). Mai întâi, se selectează prima ipoteză și se consideră a fi ipoteză de bază. Apoi, se aliniază cu aceasta cea de-a doua ipoteză, utilizând programarea dinamică și distanța Levenshtein. În urma alinierii, se obține o secvență ce conține cuvintele comune, regăsite în ambele ipoteze, dar sunt marcate și pozițiile în care au apărut inserții sau ștergeri de cuvinte. Această secvență este aliniată mai departe cu următoarea ipoteză și procesul se repetă până la final. Rezultatul obținut în urma ultimei alinieri este reprezentat sub forma unei rețele de confuzie a cuvintelor, un graf în care trecerea de la un cuvânt la următorul este formată din mai multe arce reprezentând alternative de cuvinte. În cazul în care două cuvinte diferite primesc același vot, ROVER da prioritate primei alternative din graf. Procesul de votare este dat de formula:

$$ScorVot(w) = \alpha * (N(w, i)) + (1 - \alpha) * C(w, i),$$

unde $N(w, i)$ este numărul de apariții ale cuvântului w în setul de transcrieri alternative, $C(w, i)$ este scorul de încredere al cuvântului w , iar α este un parametru de pondere între frecvența de apariție a cuvintelor și scorurile lor de încredere.

ROVER folosește trei scheme de vot ce folosesc:

- frecvența aparițiilor - ignoră informațiile despre scorul de încredere ($\alpha = 1$);
- frecvența de apariție și media scorurilor de încredere - calculează o medie a scorurilor de încredere pentru fiecare cuvânt;
- frecvența de apariție și maximul scorului de încredere - similar cu schema precedentă, doar că este luat în calcul cuvântul cu cel mai mare scor de încredere.

În [23] se întâlnește această abordare de selecție a datelor ce utilizează ROVER sub mai multe înfățișări. Este folosită frecvența de apariție a cuvintelor în transcrieri, dar și scorul de încredere. În [65], [66] și [29] se utilizează de asemenea metoda ROVER pentru a alinia ipotezele obținute de la mai multe sisteme RAV. În [53] se folosește ROVER, iar asupra scorurilor de încredere au fost aplicate mai multe praguri.

Stabilitatea acustică (A-stabil) este o altă metodă de selecție întâlnită în literatura de specialitate. Prin varierea ponderilor modelului acustic, cât și a modelului de limbă, se obțin mai multe transcrieri alternative pentru fiecare sistem. Se alege drept referință, o transcriere pentru care ponderea între modelul acustic și cel lingvistic este cea mai echilibrată.

Se calculează frecvențele de apariție ale fiecărui cuvânt din referință în celelalte transcrieri alternative și se normalizează la numărul de transcrieri alternative. Pentru fiecare propoziție din referință, scorul A-stab este definit ca media scorurilor pentru fiecare cuvânt. Pe baza acestui scor se face selecția datelor. Această metodă este întâlnită în [3], [4] și [64], unde complementaritatea este dată de folosirea unor modele acustice antrenate pe limbi diferite, folosind mai departe tehnica numită *cross-language transfer*, pentru a decoda vorbire dintr-o limbă țintă diferită.

Tehnicile de învățare automată sunt prezente în selecția datelor, utilizând clasificatori de tip *conditional random fields* (CRF). Clasificatorul selector este antrenat pentru a decide care ipoteza este corectă (sau mai bună) dintre cele rezultate din transcrierea cu sistemele complementare. Clasificatorul verificador determină dacă ipoteza selectată este corectă sau nu. În urma acestui proces, se pot obține câteva categorii de aliniere:

- cele două ipoteze corespund și sunt corecte ambele;
- cele două ipoteze nu corespund, dar una dintre ele este corectă;
- cele două ipoteze corespund, dar ambele sunt eronate;
- cele două ipoteze nu corespund, iar niciuna dintre ele nu este corectă.

Scopul clasificatorului este de a accepta datele utile (corespunzătoare categoriilor 1 și 2) și de a elimina datele eronate (categoriile 3 și 4). Metoda a fost utilizată în [58].

Rețelele neuronale sunt de asemenea utilizate în sarcină de selecție a datelor. În [25], o metodă derivată din metoda alinierii simple, pleacă de la premisa că nu întotdeauna ipotezele au aceeași lungime, intervenind posibilitatea ca ipoteze scurte să fie aliniate cu ipoteze lungi. Astfel, se întâmplă uneori ca unele cuvinte să fie aliniate la o altă parte din ipoteză, decât cea din care face parte. Acest inconvenient se rezolvă prin limitele temporale obținute de la sistemul RAV în etapa de decodare. În acest fel, se consideră incorrect aliniate cuvintele ai căror indici temporali sunt foarte depărtați. În urma alinierii, se selectează secvențe de cuvinte consecutive de lungime n (n -grame), ce sunt verificate mai departe de către o rețea neuronală, pentru a stabili dacă secvența respectivă este corectă sau nu.

3.5 ÎMBINAREA ȘI APLICAREA METODELOR DE SELECȚIE

În [23] este folosit sistemul ROVER în câteva configurații. Prima dată este luat în calcul numai numărul de apariții ale cuvintelor în transcrierile alternative, apoi acesta este considerat împreună cu probabilitățile posterioare. A treia situație ține cont de numărul de apariții și de scorul de încredere al cuvintelor. La final, se reconsideră cazurile precedente, aplicându-se în plus un proces de netezire a scorurilor.

În [29], trei sisteme RAV ce diferă prin tipul modelelor acustice și prin modelele de limbă sunt folosite pentru a genera ipoteze inițiale, cu scor de încredere la nivel de cuvânt și frază. Combinarea ipotezelor se face folosind un sistem ROVER, ce duce la obținerea unor ipoteze mai bune și a unor noi scoruri de încredere. Acestea sunt obținute prin interpolarea liniară a scorurilor inițiale și a gradului de acord asupra ipotezei. Apoi, o procedură de votare ce cuprinde un sistem principal (obținut în urma combinării cu ROVER) și trei sisteme suplimentare (sistemele inițiale) sunt utilizate pentru a recalibra scorurile de încredere. Se va considera, pe rând, că fiecare sistem este cel principal. Numai ipoteza care provine de la sistemul principal este considerată ca fiind potențial corectă, în timp ce sistemele suplimentare au rolul de a calibra scorul de încredere al sistemului principal.

În [53], reducerea erorilor de transcriere prin aplicarea metodei ROVER a condus la îmbunătățirea performanțelor. Transcrierile obținute de la diferite sisteme au fost combinate, selecția datelor făcându-se pe baza unui scor de încredere. S-a testat aplicarea mai multor praguri asupra acestor scoruri.

În [58], un set de clasificatori de tip *conditional random fields (CRF)* sunt folosiți pentru a selecta și verifica cea mai bună ipoteză dintre cele rezultate din transcrierea cu sistemul GMM-HMM sau DNN-HMM la nivel de caracter/cuvânt. Se folosește un clasificator selector, care alege una dintre ipoteze, împreună cu un clasificator verificator care determină corectitudinea ipotezei selectate. Deoarece se observa un tipar diferit între ipotezele obținute cu GMM față de DNN, clasificatorul CRF combină această diversitate și determină ce ipoteză ar trebui selectată pentru a reantrena modelul acustic. La început, textele sunt aliniate în perechi la nivel de caracter. O ipoteză corectă sau îmbunătățită este selectată pe baza ipotezelor complementare. În urma acestui proces, se pot obține 5 categorii de aliniere: cele două ipoteze corespund și sunt corecte ambele (1), cele două ipoteze corespund, dar ambele sunt eronate (2), cele două ipoteze nu corespund, iar niciuna dintre ele nu este corectă (3) sau cea bazată pe DNN este corectă (4) sau cea bazată pe GMM este corectă (5). Setul de clasificatori utilizează mai multe tipuri de trăsături, separate în două mari clase: trăsături bazate pe sistemul RAV (scorul de încredere al cuvântului, durata cuvântului, scorul cuvântului dat de modelul de limbă, scorul cuvântului dat de modelul acustic, numărul de cuvinte din lățiș conectate la cuvântul curent, numărul de cuvinte suprapuse în lățiș peste cuvântul actual) și trăsături bazate pe text (partea de vorbire reprezentată de cuvânt, probabilitatea dată de un model de limba 5-gram, comportamentul de tip back-off rezultat dintr-un model de limba 5-gram). În [20] se realizează selecția transcrierilor obținute pe baza scorurilor de încredere. Se caută un prag al acestui scor, punând în balanță calitatea datelor obținute și cantitatea lor. S-a observat că scorul de încredere la nivel de propoziție este mai relevant decât scorul la nivel de cuvânt.

În [26] ipotezele sunt aliniate complet și se consideră că părțile comune nu conțin erori, deși în realitate acestea au apărut, dar într-un procent scăzut. Mai departe, s-a încercat scăderea erorii pe baza scorului de încredere. Au fost testate mai multe praguri, iar cu cât pragul a fost mai înalt, cu atât eroarea a fost mai mică. De asemenea, au fost propuse câteva metode de normare a pragului și s-a încercat stabilirea unor formule pentru a obține rezultate mai bune. Combinația a două constrângeri, minimul de cuvinte consecutive și utilizarea unui prag pentru scorul de încredere, a dus la o scădere a erorii. Cel mai mare câștig a fost obținut prin utilizarea pragului variabil.

În [31] se realizează alinierea ipotezelor și selecția părților comune. Aceeași abordare se întâlnește în [62], iar mai departe se folosesc informațiile date de scorul de încredere și se aplică diferite praguri, pentru a efectua selecția finală.

În [25], ipotezele sunt aliniate, fiind stocate informații despre limitele temporale, forma ortografică, scorul dat de modelul acustic și scorul de încredere. Selecția ipotezelor se face cu ajutorul unui clasificator bazat pe rețele neuronale, ce încearcă să stabilească un model pentru secvențele de cuvinte consecutive (clasificarea n-gramelor). Acesta este mai eficient decât stabilirea unui prag pentru scorul de încredere, deoarece această metodă din urmă omite cuvintele cu un scor mic. Totodată, există cazuri când cuvinte transcrise incorect au un scor de încredere înalt, sarcină de decizie fiind dificilă. Pentru a antrena rețeaua neuronală, este necesar un corpus de date pentru care există informații despre corectitudinea fiecărui n-gram existent. Este important că datele de antrenare să fie echilibrate, având aceeași cantitate de n-gram corecte și incorecte. Ordinul n-gramelor este cuprins între 1 și 6. Într-un prim experiment, vectorul de trăsături de la intrarea rețelei a fost compus din scorul dat de modelul acustic, împreună cu scorul de încredere, obținute de la cele două sisteme RAV, pentru fiecare cuvânt. Al doilea experiment a exclus scorul dat de modelul acustic. Al treilea experiment este similar cu al doilea, diferența fiind reprezentată de scalarea scorului de încredere prin înmulțirea lui cu un factor constant. Cel de-al patrulea experiment a folosit un singur scor de încredere, obținut ca o medie a celor furnizate de cele două sisteme RAV.

În [32], selecția ipotezelor în scopul adnotării automate a fost realizată folosind două criterii: alinierea și stabilirea unui prag pentru scorul de încredere la nivel de cuvânt, precum și alinierea și stabilirea unui prag pentru scorul de încredere la nivelul stărilor acustice.

În [3], selecția datelor se face pe baza scorurilor de încredere, calculate fie cu algoritmul *gamma*, fie pe baza stabilității acustice (A-stabil). Prima metodă presupune calculul probabilității unei căi din latică, în același fel în care algoritmul *forward-backward* funcționează la decodarea HMM-urilor. Un nod în latică poate fi văzut ca o stare HMM, iar o cale din latică este o posibilă tranziție între stări. Deoarece nodurile sunt asociate cuvintelor din ipoteză, probabilitatea de emisie a unui nod este scorul acustic pentru acel segment temporal. Probabilitatea de tranziție între cuvinte este dată de modelul de limbă. Aceste două probabilități, date de modelul acustic și de modelul de limbă, compun scorul de încredere, puternic corelat cu eroarea de recunoaștere. Metoda stabilității acustice presupune generarea mai multor ipoteze. Fiecare ipoteză este aliniată cu o ieșire de referință, definită ca ipoteza cu cea mai bună pondere între modelul acustic și cel de limbă. Pentru fiecare cuvânt din referință, scorul de încredere este definit ca numărul de apariții ale cuvântului în ipotezele alternative, raportat la numărul de ipoteze alternative.

3.6 PERFORMANȚE ALE METODELOR DE ADNOTARE AUTOMATĂ ȘI ANTRENARE NESUPERVIZATĂ

Experimentele ce au utilizat metodele de diversitate și selecție prezentate sunt destul de eterogene, iar o comparație directă între acestea este dificil de realizat. Totuși, unii autori au făcut specificații cu privire la performanțele unor metode, în comparație cu altele. De asemenea, au fost oferite unele date numerice cu privire la performanțele obținute, astfel încât se poate face o privire de ansamblu despre care metode sunt mai eficiente.

În [20] se face o apreciere asupra scorurilor de încredere, observându-se că acestea sunt mai relevante la nivel de propoziție decât la nivel de cuvânt.

În [25] se folosește o rețea neuronală pentru a stabili corectitudinea secvențelor n-gram selectate din transcrieri. Această metodă a obținut o acuratețe de 84.55%. Utilizarea scorului dat de modelul acustic a condus la rezultate mai slabe față de situațiile în care acesta a fost exclus. De asemenea, s-a observat că este mai util ca la intrarea rețelei neuronale să existe două valori ale scorului de încredere, în locul uneia singure.

În [62] se observă că în experimente anterioare, părțile comune din cele două transcrieri sunt corecte în proporție de aproximativ 90%. Transcrierile au fost obținute prin decodarea normală și inversă, folosind același sistem RAV. Două constrângeri suplimentare au fost adăugate la selecția datelor: segmentele trebuie să fie de cel puțin 10 cuvinte și să fie precedate sau succedate de cel puțin 300 ms de non-vorbire. Reantrenarea sistemului cu datele inițiale împreună cu datele nou obținute, a condus la o scădere absolută cu 0.4% a erorii la nivel de cuvânt. Metoda selecției pe baza transcrierilor obținute prin decodare în ordine normală și inversă a surclasat selecția datelor pe baza scorului de încredere.

În [53], rezultatele au arătat că pentru un prag jos al scorului de încredere, cuprins între 0.1 -0.3, procentul datelor selectate este de aproape 100%, obținându-se pentru acestea un WER de aproximativ 7%.

În [32] se reușește ca prin aplicarea complementarității la nivelul modelării acustice să obțină o selecție a datelor în proporție de 80%, iar îmbunătățirea relativă adusă sistemului RAV final este de 7%.

În [58], selecția și verificarea ipotezelor utilizând clasificatori de tip *conditional random fields* surclasează alte încercări bazate pe scor de încredere sau combinarea sistemelor folosind ROVER.

3.7 CONCLUZII

Ideea utilizării sistemelor RAV complementare este una fiabilă în contextul adnotării automate a corpusurilor de vorbire. Diversitatea sistemelor poate fi dată de mai mulți factori, cei mai importanți fiind compoziția seturilor de antrenare acustică/lingvistică, tipul trăsăturilor extrase, tipul modelelor antrenate, tipul algoritmilor utilizați la antrenare/decodare. Nu se poate afirma cu certitudine că vreuna dintre metode oferă complementaritate mai bună, fiecare poate fi eficientă într-o situație particulară.

În ceea ce privește metodele de combinare și selecție, se observă faptul că ROVER este o procedură clasică, ce a fost aplicată cu succes în multe situații. O altă metodă clasică este cea a utilizării scorurilor de încredere, deși nu întotdeauna sistemele RAV scot la ieșire scoruri corelate cu acuratețea transcrierii. Totuși, alternativele ce se bazează pe tehnici de învățare automată (clasificatori CRF, rețele neuronale) au început să fie utilizate în ultima vreme, iar rezultatele sunt mai bune în comparație cu metodele clasice. Metoda stabilității acustice este derivată din ROVER și se folosește preponderent în situațiile în care se folosesc sisteme în limbi sursa diferite față de limba sursă.

CAPITOLUL 4

EXPERIMENTE DE RECUNOAȘTERE AUTOMATĂ A VORBIRII ÎN LIMBA ROMÂNĂ

Capitolul curent prezintă un set extins de experimente de RAV pentru limba română. Acestea au venit sub forma unei munci complementare, o continuare a unui efort susținut de-a lungul timpului [64, 65], în cadrul grupului de cercetare Speed. Acest efort a dus la crearea primului sistem de RVC-VE, independent de vorbitor pentru limba română. Experimentele din capitolul curent au avut ca scop extinderea acestui demers, principalul element de noutate fiind utilizarea tehnicilor și algoritmilor de învățare automată, atât pentru modelarea acustică, cât și pentru modelarea lingvistică. Astfel, a fost creat primul sistem de transcriere a vorbirii în text pentru limba română bazat pe rețele neuronale profunde [36].

4.1 STADIUL ANTERIOR ȘI ELEMENTE INOVATIVE

Deși principalul element inovativ este reprezentat de antrenarea sistemului utilizând rețele neuronale, în locul abordărilor clasice, probabilistice, în care modelul acustic este de tip HMM-GMM, iar modelul de limbă este de tip n-gram, există și o serie de alte aspecte ce aduc noutate față de sistemul precedent. În primul rând, corpusul de vorbire folosit la antrenare a cunoscut o extindere masivă. Apoi, procesarea trăsăturilor extrase din semnalul vocal s-a făcut cu ajutorul

unor algoritmi suplimentari. Modele acustice bazate pe platforma HMM-GMM au cunoscut îmbunătățiri aduse la nivelul algoritmului de antrenare. Nu în ultimul rând, vocabularul folosit la antrenarea modelelor de limbă a fost extins, la fel și complexitatea acestora.

4.1.1 Extinderea corpusului de vorbire

Sistemul RVC-VE pentru limba română [42], creat anterior în cadrul Speed în anul 2014, a fost antrenat pe două mari corpusuri de vorbire. *RSC-train* însumează 100 de ore de vorbire citită, în condiții de liniște. *SSC-train1* însumează 27.5 ore de vorbire spontană, extrasă din emisiuni, uneori în condiții de zgomot, dar și din buletine de știri, unde vorbirea este citită, fără zgomot și cu o pronunție curată.

Un corpus nou de vorbire a fost achiziționat și adnotat automat între timp. Aproximativ 350 de ore de vorbire din buletine de știri și vorbire conversațională, fără nicio transcriere aferentă, au fost colectate din mediul online. Acest corpus neadnotat a fost transcris cu două sisteme RAV complementare, diferența dintre ele fiind dată de tipul vorbirii cu care au fost antrenate. Transcrierile obținute de la cele două sisteme au fost aliniate și comparate, iar părțile identice au fost considerate corecte. Metodologia a fost descrisă pe larg în [67], unde s-a arătat că aceasta funcționează corect pentru aproximativ 90% din datele selectate. Segmentele audio corespunzătoare părților identice au fost tăiate din corpusul inițial. În acest fel, a fost creat un nou corpus, denumit *SSC-train2*, ce constă în 103 ore de vorbire. Acesta a fost utilizat ulterior la antrenarea modelelor acustice.

4.1.2 Extragerea trăsăturilor acustice

Sistemul RVC-VE creat anterior a folosit trăsături vocale de tip MFCC, extrase cu ajutorul unei ferestre Hamming cu durata de 25 ms și o perioadă de repetiție de 10 ms. Vectorul final cu trăsături este format din 40 de elemente: 13 coeficienți cepstrali standard, împreună cu derivatele lor de ordinul 1 și 2, la care se adaugă și un coeficient de energie.

Sistemul nou utilizează de asemenea coeficienții cepstrali, asupra cărora aplica unele tehnici de prelucrare a acestor trăsături acustice. Normarea mediei și variantei coeficienților cepstrali (CMVN) a fost aplicată pentru a aduce coeficienții la medie zero și varianță egală cu unitatea. Această transformare a avut ca scop eliminarea variațiilor induse de fiecare vorbitor, precum și cele induse de mediul acustic în care are loc vorbirea. De asemenea, a fost aplicată tehnica analizei discriminatorii liniare (LDA) și metoda verosimilității maxime (MLLT). Acestea sunt utilizate ca o alternativă la derivatele de ordinul 1 și 2 ale coeficienților acustici. LDA are ca scop reducerea dimensionalității și păstrarea informației discriminatorii cât mai mult posibil. Uneori, prin utilizarea datelor din surse multiple, se ajunge la creșterea spațiului în care sunt reprezentate trăsăturile. LDA reduce datele la un set mai ușor de gestionat și aliniat pe direcția principală de variație a acestora [16]. MLLT aplică o transformare asupra coeficienților acustici în încercarea de a captura corelația dintre componentele vectorilor de trăsături [19].

4.1.3 Îmbunătățiri la nivelul modelului acustic de tip HMM-GMM

Ca un pas intermediar în crearea modelului acustic bazat pe rețele neuronale, dar necesar, deoarece antrenarea acestuia a plecat de la alinierile date de un model probabilistic, au fost antrenate și modele acustice de tip HMM-GMM, folosind algoritmi discriminativi. A fost aplicată tehnica maximizării informației mutuale (MMI), ce încearcă să maximizeze probabilitățile posterioare pentru vorbirea folosită la antrenare. Tot în acest context a fost aplicată și tehnica de adaptare la vorbitor (SAT). Aceasta încearcă să adapteze modelul acustic la caracteristicile fiecărui vorbitor, folosind ca abordare regresia bazată pe verosimilitatea maximă [14].

4.1.4 Îmbunătățiri la nivelul modelului lingvistic și al vocabularului

Din punctul de vedere al modelului de limbă de tip n-gram utilizat în sistemul RVC-VE anterior, acesta a fost creat pe baza unui vocabular de 64.000 cuvinte, restricția fiind impusă de utilitarul cu care se realiza decodarea semnalului vocal. În implementarea curentă, deoarece utilitarul cu

care se face decodarea este altul, această limitare a dispărut, iar dimensiunea vocabularului a putut fi variată, fiind create modele de limba cu 100.000, 150.000 și 200.000 cuvinte. Acest fapt este unul benefic, mai ales că limba română este o limbă bogată, existând foarte multe structuri morfologice diferite.

Un alt element inovativ introdus în noul sistem RVC-VE este dat de utilizarea tehnicii de reevaluare a laticelor (*lattice rescoring*) sub forma unei componente finale a procesului de decodare. Metoda presupune decodarea semnalului vocal utilizând un model de limbă simplu, iar graful cu transcrieri alternative obținut este apoi reevaluat cu autorul unui model de limbă complex, rezultând astfel o transcriere cu un grad de acuratețe mai mare.

4.1.5 Modelarea acustică folosind rețele neuronale

Modelul acustic antrenat cu rețele neuronale este bazat pe arhitectura de tip *time delay neuronal network*, ce modelează dependențe temporale lungi între cadrele de semnal. Efectul este similar cu acela din cazul rețelelor neuronale recurente (RNN), fiind învățată dinamica temporală a semnalului, dar timpul de antrenare este mai scăzut, fiind comparabil cu o rețea standard de tip *feed-forward*. Funcția de activare folosită în rețeaua neuronală este o funcție *p-norm* descrisă în [70, 71]. Funcția de activare nu este calculată pentru toți neuronii la toate momentele de timp. O metodă de optimizare, denumită subeșantionare [16], ce presupune faptul că activările vecine sunt corelate, este folosită pentru a reduce complexitatea computațională din etapa de antrenare. Parametrii de la intrarea rețelei sunt coeficienți cepstrali 40-dimensional, câte un vector de coeficienți pentru fiecare cadru de semnal, la care nu s-a aplicat tehnica CMVN. Împreună cu aceștia, se folosesc și coeficienți de tip *iVectors*, 100-dimensional, ce furnizează informații privitoare la vorbitor și mediul ambiant în care are loc vorbirea, fiind utili la antrenarea adaptată la vorbitor [16].

4.1.6 Modelarea lingvistică folosind rețele neuronale

Deși modelele de limbă probabilistice, de tip n-gram, se bucură în continuare de un mare succes, rezultatele îmbunătățite oferite de rețelele neuronale au indicat trecerea la folosirea acestora și în cazul modelării lingvistice. Cele mai indicate tipuri de rețele neuronale pentru această sarcină s-au dovedit a fi rețelele neuronale recurente. Ele permit procesarea unor secvențe de lungime arbitrară, corelate temporal. Principalul avantaj îl reprezintă faptul că au memorie, ieșirea în urma prelucrării secvenței curente este influențată de ieșirea în urma prelucrării secvențelor anterioare. Acest fapt poate aduce îmbunătățiri semnificative în cazul modelării probabilității de apariție a cuvintelor dintr-o propoziție, modelele de tip n-gram fiind limitate la a lua în considerare un istoric mai scurt, de obicei de 2-3 cuvinte.

În noul sistem RVC-VE, au fost utilizate modele lingvistice antrenate cu RNN. Deși nu au fost efectuate experimente exhaustive, existând în continuare direcții ce ar putea fi exploatate, rezultatele obținute până acum certifică faptul că acest demers este unul de perspectivă.

4.1.7 Utilitarul Kaldi

Sistemul RVC-VE anterior a fost creat cu ajutorul utilitarului CMU Sphinx [49]. Îmbunătățirile specificate în secțiunile 4.1.2- 4.1.6 se regăsesc implementate în utilitarul Kaldi [19]. Kaldi este un utilitar cu sursă deschisă, scris în C++, specializat în sarcini de recunoaștere automată a vorbirii și recunoașterea vorbitorului. Kaldi depinde de două mari librării externe, una pentru implementarea arhitecturilor cu stări finite, iar alta pentru suportul operațiilor algebrice.

Kaldi folosește în mod implicit trăsături vocale de tip MFCC și PLP, asupra cărora se pot aplica diverse transformări, enumerate în secțiunea 4.1.2.

Din punct de vedere al modelării acustice, Kaldi permite crearea modelelor de tip HMM-GMM, împreună cu unele îmbunătățiri care pot fi aduse acestor modele, așa cum a fost descris în secțiunea 4.1.3. Pentru a crea modele acustice în Kaldi pe baza platformei HMM-GMM, mai întâi este antrenat un model independent de context (*monophone*), pornind de la alinierea inițială ale datelor de antrenare, fiecare fonem având aceeași probabilitate de apariție, cu aceeași medie și varianță. Acest model folosește 13 coeficienți MFCC și derivatele lor de ordin 1 și 2. Asupra

modelului de monofoneme este aplicată tehnica alinierii forțate. Apoi, o nouă etapă de antrenare, dependentă de context, generează un model acustic bazat pe trifoneme (*Tri1*), ce folosește aceleași trăsături acustice ca în cazul precedent. O nouă aliniere forțată este efectuată asupra modelului *Tri1*, urmată de aplicarea transformatorilor LDA+MLLT, generând astfel un nou model de trifoneme, numit *Tri2*. Folosind alinierea forțată efectuată pe modelul *Tri2* și antrenând apoi utilizând tehnica adaptării la vorbitor, modelul acustic *Tri3* este obținut. Acest model este îmbunătățit aplicând o tehnică de antrenare discriminativă (MMI).

Cel mai mare avantaj al utilitarului Kaldi este prezența algoritmilor de antrenare bazați pe rețele neuronale profunde, existând totodată posibilitatea rulării lor pe procesoare grafice (GPU). Implementarea *NNET2* cu rețele neuronale din Kaldi este descrisă în detaliu în [1] și [16]. O altă implementare, denumită *NNET3*, suportă mai multe tipuri de rețele neuronale (RNN, LSTM).

4.2 PREGĂTIREA EXPERIMENTELOR

4.2.1 Modele acustice

Așa cum a fost explicat în secțiunea 4.1.7, Kaldi oferă o largă varietate de algoritmi și permite antrenarea modelelor acustice atât pe baza platformei HMM-GMM, cât și pe baza rețelelor neuronale profunde (DNN). Experimentele prezentate în continuare au fost efectuate adaptând o rețetă existentă în Kaldi. Deoarece rețetă pentru limba engleză pe baza corpusului de vorbire TED-LIUM [12] antrenează date cu o dimensiune similară ca durată cu datele pentru limba română, parametrii principali folosiți în etapa de antrenare din această rețetă au fost folosiți și în testele pentru limba română.

Din punct de vedere al trăsăturilor vocale, atât în modelele HMM-GMM, cât și în modelele DNN se folosesc coeficienți MFCC extrași cu o fereastră Hamming cu durată de 25 ms (vectori de 13 coeficienți pentru fiecare cadru, împreună cu derivatele lor de ordin 1 și 2). În plus, în cazul modelelor DNN, 100 coeficienți de tip *iVectors*, extrași din fiecare cadru, sunt atașați la vectorul de trăsături.

Pentru modelele acustice bazate pe HMM-GMM, experimentele s-au derulat în 5 configurații. Un model acustic pentru foneme independente de context (*Monophone*) a fost antrenat pe parcursul a 40 iterații folosind 1.000 densități gaussiene. Apoi, un model acustic pentru foneme dependente de context, derivat din modelul anterior și denumit *Tri1*, a fost antrenat folosind 2.500 senone și 30.000 densități gaussiene. Modelul *Tri1* a fost utilizat ca date de intrare la antrenarea celui de-al doilea model acustic dependent de context, *Tri2*. În *Tri2*, numărul de senone a fost crescut mai departe la 4.000, în timp ce numărul de densități gaussiene a fost crescut la 50.000. În final, *Tri2* a fost utilizat ca punct de plecare pentru antrenarea unui model mult mai complex cu 5.000 senone și 100.000 densități gaussiene. Numărul densităților gaussiene reprezintă numărul total raportat la toate senonele din modelul acustic.

Trei experimente cu rețele neuronale au fost efectuate, folosind implementarea *NNET2* și variind numărul neuronilor din stratul de intrare și cel de ieșire, numărul de straturi ascunse și numărul epocilor de antrenare. Primul experiment a utilizat o rețea neuronală cu 6 straturi ascunse, un strat de intrare cu 3.500 neuroni și un strat de ieșire cu 350 neuroni, procesul de antrenare având loc pe parcursul a 8 epoci. Similar cu parametrii utilizați la antrenarea modelului HMM-GMM, aceste numere se regăsesc în rețetă standard bazată pe corpusul TED-LIUM în limba engleză. În al doilea experiment, numărul neuronilor de pe straturile de intrare și ieșire au fost crescuți cu 25% (la 4.400, respectiv 440). În final, cel de-al treilea experiment a presupus reducerea numărului de straturi ascunse la 3 și creșterea numărului de epoci de antrenare la 15.

Deoarece am dorit compararea rezultatelor obținute pe platforma HMM-GMM, antrenate cu Kaldi, respectiv cu CMU Sphinx, raportate în [42] de membrii grupul Speed, a fost reantrenat cel mai bun model acustic obținut la momentul respectiv, folosind o versiune actualizată a utilitarului CMU Sphinx. Astfel, au fost efectuate experimente ce implică 13 coeficienți PNCC (Power Normalized Cepstral Coefficients) și derivatele lor de ordin 1 și 2. Fonemele au fost

modelate cu 4.000 senone și 128 densități gaussiene per senone (un total de 512.000 densități gaussiene).

Toate modelele acustice obținute atât cu Kaldi, cât și cu CMU Sphinx, au implicat antrenarea a 36 foneme dependente de context, specifice limbii române. Corpusul de antrenare are o durată de aproximativ 225 ore și constă în subseturile *RSC-train*, *SSC-train1*, *SSC-train2*, ce vor fi descrise în secțiunea 4.2.3.

4.2.2 Modele lingvistice

Modelele de limbă create au implicat două abordări: una probabilistică, de tip n-gram, folosind utilitarul SRI-LM [9] și alta bazată pe rețele neuronale recurente, RNN. Două corpusuri de text au fost utilizate pentru această sarcină, interpolate cu ponderea 0.5: unul ce conține 315 M cuvinte, adunate de pe site-urile de știri din mediul online, iar altul ce conține 40 M cuvinte, ce reprezintă transcrieri ale unor conferințe. Modelele de limbă diferă prin ordinul n-gram (de la 1-gram la 5-gram) și prin numărul de cuvinte din vocabular: 64.000, 100.000, 150.000 și 200.000 cuvinte. Deoarece corpusul de text a fost colectat de pe Internet, au fost situații când nu existau diacritice. Acestea au fost restaurate pentru unele părți de text, conform metodei prezentate în [56].

Evaluarea modelelor de limbă urmărind raportarea metricilor de perplexitate și cuvinte în afara vocabularului. Pentru acest proces, au fost utilizate 3 corpusuri de evaluare: transcrierile de la corpusul audio *RSC-eval*, ce conțin 43.000 de cuvinte, transcrierile de la corpusul audio *SSC-eval*, ce conțin 36.000 de cuvinte (ambele descrise pe larg în secțiunea 4.2.3) și un corpus de 4 milioane de cuvinte, ce conține transcrieri ale unor emisiuni de radio și televiziune.

Rezultatele raportate în secțiunea 4.3.1 au fost obținute folosind un model de limbă 3-gram cu 64.000 de cuvinte în vocabular. Secțiunea 4.3.3 prezintă rezultatele obținute prin combinarea a două modele de limbă, unul folosit la decodarea semnalului vocal și obținerea grafului cu transcrieri alternative, iar alt model de limbă pentru reevaluarea acestui graf și obținerea transcrierii finale.

4.2.3 Corpusul de vorbire pentru antrenare și evaluare

Corpusul de vorbire prezentat în [42] și extins ulterior prin achiziții de noi date, a fost utilizat pentru a crea noul sistem RVC-VE. Acesta constă în două subseturi: primul subset (*RSC*) conține vorbire citită înregistrată într-un mediu acustic fără zgomot, în timp ce al doilea subset (*SSC*) conține vorbire spontană, preluată din transmisiuni de radio și televiziune, unele fiind înregistrate în mediu acustic fără zgomot, în timp ce altele sunt afectate de zgomot de fundal (muzică, râsete, etc.). Ambele au fost împărțite în seturi de antrenare și evaluare, folosite pentru a antrena și evalua sistemul de recunoaștere automată a vorbirii. Tabelul 4.1 sumarizează toate aceste seturi și furnizează detalii despre caracteristicile lor.

Scop	Set	Durată	
Antrenare	RSC-train	94h 46m	225h 31m
	SSC-train1	27h 27m	
	SSC-train2	103h 17m	
Evaluare	RSC-eval	5h 29m	8h 58m
	SSC-eval	3h 29m	

Tabelul 4.1 Corpusul de vorbire

În conformitate cu [42], *RSC-train* constă în 145.000 rostiri de la 157 de vorbitori diferiți, însumând un total de 100 ore de vorbire. *RSC-eval* constă în 2604 rostiri de la 22 de vorbitori diferiți, însumând un total de 5.5 ore de vorbire. Ambele seturi conțin atât rostiri ale unui singur cuvânt, cât și rostiri de fraze.

SSC-train1 conține vorbire spontană în condiții de zgomot, preluată din emisiuni de radio și televiziune, precum și vorbire citită, fără zgomot, preluată din buletine de știri. Setul este format din 54.000 rostiri și însumează un total de 27.5 ore de vorbire. *SSC-train2* este un subset nou

achiziționat, adăugat la seturile deja existente în [42]. Acesta conține 170.000 rostiri spontane și însumează 103 ore de vorbire. *SSC-eval* constă în 3035 rostiri, atât în condiții de liniște, cât și în condiții de zgomot și însumează 3.5 ore de vorbire.

4.3 REZULTATE EXPERIMENTALE DE RECUNOAȘTERE AUTOMATĂ A VORBIRII

4.3.1 Modele acustice de tip HMM-GMM și HMM-DNN antrenate cu Kaldi

Toate experimentele prezentate în această secțiune au fost efectuate folosind același model de limbă, un 3-gram cu un vocabular de 64.000 cuvinte.

Tabelul 4.2 sumarizează rezultatele obținute cu Kaldi folosind platforma HMM-GMM. Se observă că îmbunătățirile aplicate asupra trăsăturilor acustice, împreună cu creșterea numărului de senone și numărul densităților gaussiene, tind să reducă eroarea la nivel de cuvânt (WER). Modelul acustic cu 5.000 senone și 100.000 densități gaussiene, care beneficiază de aplicarea transformatelor LDA+MLLT asupra trăsăturilor vocale, precum și adaptarea la vorbitor (SAT) și antrenarea de tip discriminativ (MMI), obține cele mai bune rezultate. Pentru vorbire citită, scăderea relativă a erorii la nivel de cuvânt (WER) este semnificativă: 27.1%, 20.5% și 7.5%, în comparație cu primele 3 linii din tabel. Pentru vorbire spontană, scăderea relativă a WER este mai puțin semnificativă: 11%, 8.4% și 3.8%.

Modelul acustic			WER [%]	
# senone	# Gaussiene	Transf. trăsături și tehnici antrenare	RSC-eval	SSC-eval
2.500	30.000	n/a	12.3	29.7
4.000	50.000	LDA+MLLT	11.3	28.9
5.000	100.000	+SAT	9.7	27.5
5.000	100.000	+MMI	9.0	26.4

Tabelul 4.2 Rezultate pentru modelele acustice bazate pe HMM-GMM

Tabelul 4.3 sumarizează rezultatele obținute pentru cele 3 experimente cu modele acustice antrenate pe bază de DNN, așa cum a fost descris în secțiunea 4.2.1. Deoarece antrenarea a avut loc pe parcursul a câtorva epoci, seturile de evaluare au fost decodate și cu modele intermediare, obținute după un anumit număr de epoci de antrenare.

Tabelul 4.3 arată că modelele acustice antrenate cu 1, 2 până la 15 epoci obțin rezultate similare atât pe vorbire citită, cât și pe vorbire conversațională. Mai mult, diverse experimente efectuate prin creșterea dimensiunii numărului de straturi de intrare sau ieșire sau scăderea numărului de straturi ascunse, par să nu aducă îmbunătățiri semnificative. Totuși, comparând cu rezultatele obținute pe baza modelelor acustice antrenate cu HMM-GMM, rezultatele bazate pe DNN sunt mult mai bune: o scădere relativă de 30% a WER pe vorbire citită și o scădere relativă de 20% a WER pe vorbire conversațională.

Configurație DNN	# epoci antrenare	WER [%]	
		RSC-eval	SSC-eval
3500 neuroni intrare 350 neuroni ieșire 6 straturi ascunse	1	6.4	21.7
	2	6.2	21.0
	3	6.3	20.7
	4	6.4	21.0
	5	6.4	21.2
	8	6.9	22.1
4400 neuroni intrare 440 neuroni ieșire 6 straturi ascunse	1	6.5	21.4
	2	6.3	21.0
	3	6.3	21.0
	4	6.6	21.1
	5	6.8	21.4
	8	7.3	22.4
4400 neuroni intrare 440 neuroni ieșire 3 straturi ascunse	1	6.5	22.3
	2	6.4	21.9
	3	6.4	22.0
	4	6.5	22.1
	5	6.5	22.0
	10	6.6	22.3
	15	6.6	22.5

Tabelul 4.3 Rezultate pentru modele acustice bazate pe DNN

Un alt aspect important ce privește modelele acustice bazate pe DNN este complexitatea computațională a algoritmilor de antrenare. Este foarte bine cunoscut faptul că etapa de antrenarea a unei rețele neuronale durează un timp substanțial și ridică probleme legate de resursele de calcul necesare. În acest sens, experimentele prezentate în Tabelul 4.3 au fost efectuate pe o singură stație de lucru, ce are un procesor Intel Xeon 3.20 GHz cu 16 nuclee, 192 GB memorie RAM și o placă grafică NVIDIA Quadro M4000. Placa grafică a putut fi utilizată grație implementării bazate pe CUDA [79] ce se regăsește în algoritmi din Kaldi. În acest context, etapa de antrenare pentru experimentele prezentate în Tabelul 4.3 a durat 57 ore, 70 ore și respectiv 57 ore.

Pentru a reduce timpul necesar antrenării și pentru a crea posibilitatea de a efectua cât mai multe experimente simultane, a fost extinsă infrastructura de calcul. Astfel, a fost creat un grup de 4 stații de lucru, cu aceeași configurație ca cea descrisă mai sus, ce au fost utilizate pentru procesare paralelă. Interconectarea fizică s-a realizat prin rețea, în timp ce interconectarea software este gestionată cu ajutorul aplicației Grid Engine [83]. Pe acest grup de stații, procesarea paralelă pentru antrenarea modelelor acustice de tip DNN a durat între 3x și 3.3x mai puțin. Nu s-a obținut o accelerare de 4x deoarece nu toate operațiile permit paralelizare. În concluzie, crearea acestui grup de stații a fost un pas foarte important, deoarece au putut fi efectuate experimente în mai multe configurații, iar timpul necesar a fost relativ scurt.

4.3.2 Modele de limbă complexe de tip *n*-gram

Toate experimentele prezentate în această secțiune au fost efectuate folosind cel mai bun model acustic bazat pe DNN (evidențiat în Tabelul 4.3).

Mai multe modele de limbă au fost create pentru a găsi cel mai bun compromis între dimensiunea modelului și performanța lui în termeni de perplexitate (PPL), procentul de cuvinte în afara vocabularului (OOV) și rata de eroare la nivel de cuvânt (WER). Tabelul 4.4 compară ratele OOV pentru diverse modele de limbă cu diverse dimensiuni ale vocabularului. Cu cât

vocabularul este mai bogat, cu atât scade probabilitatea de apariție în setul de evaluare a unor cuvinte care nu sunt în modelul de limbă. Sistemul RVC-VE anterior utiliza un model de limbă cu dimensiunea de 64.000 cuvinte. Acest model de limbă are un OOV de 1.4% pe *RSC-eval* și 2.6% pe *SSC-eval*. În comparație cu acesta, modelul de limbă cu 100.000 cuvinte are un OOV mult mai scăzut: 57% mai scăzut pe *RSC-eval*, 15% mai scăzut pe *SSC-eval*. Crescând și mai mult dimensiunea vocabularului, la 200.000 cuvinte, se obțin noi câștiguri în ceea ce privește rata OOV: 93% pe *RSC-eval* și 23% pe *SSC-eval*.

Dimensiune vocabular	OOV [%]		
	RSC-eval	SSC-eval	meet-eval
64k cuvinte	2.6	1.4	0.6
100k cuvinte	2.2	0.6	0.3
150k cuvinte	2.0	0.3	0.2
200k cuvinte	1.9	0.1	0.2

Tabelul 4.4 Ratele OOV în raport cu creșterea vocabularului

Tabelul 4.5 prezintă diverse modele de limbă al căror ordin n-gram a fost variat (de la 1-gram la 5-gram), împreună cu dimensiunea vocabularului (100.000, 150.000 și 200.000 cuvinte), evaluate în termeni de PPL și WER. Se poate observa că pentru același ordin, dar dimensiuni diferite ale vocabularului, ratele de perplexitate sunt aproximativ egale.

Având în vedere rezultatele discutate mai sus, am preconizat o îmbunătățire semnificativă în ceea ce privește WER pentru modelul de limbă cu 200.000 cuvinte, în comparație cu modelul de limbă cu 100.000 cuvinte. Totuși, Tabelul 4.5 arată că extinderea vocabularului este benefică atunci când evaluarea se face pe setul de evaluare *RSC-eval* (între 5% și 11% scădere a WER, în funcție de ordinul n-gram al modelului de limbă), în timp ce evaluarea pe setul de evaluare *SSC-eval* oferă un WER identic.

Tabelul 4.5 arată de asemenea că tranziția de la 1-gram la 2-gram aduce o scădere foarte mare în ceea ce privește perplexitatea (84%), în timp ce tranziția de la 2-gram la 3-gram aduce o scădere mai puțin semnificativă (39%). Tranzițiile de la 3-gram la 4-gram și de la 4-gram la 5-gram aduc și ele îmbunătățiri semnificative (10%, respectiv 2%, scădere în ceea ce privește perplexitatea).

În cazul modelelor de limbă de tip 4-gram și 5-gram nu au putut fi efectuate experimente de recunoaștere automată a vorbirii din cauza constrângerilor de memorie. Generarea grafului de decodare pentru aceste cazuri depășește cantitatea de memorie RAM disponibilă (192 GB).

	Ordin LM	PPL			WER[%]	
		RSC-eval	SSC-eval	meet-eval	RSC-eval	SSC-eval
100k cuvinte	1-gram	1345	1938	1046	15.0	36.5
	2-gram	216	300	179	6.44	23.4
	3-gram	129	184	117	5.18	20.6
	4-gram	116	166	109	n/a	
	5-gram	113	162	108		
150k cuvinte	1-gram	1368	1984	1057	14.6	36.4
	2-gram	307	220	181	6.26	23.3
	3-gram	132	189	118	5.00	20.5
	4-gram	118	170	110	n/a	
	5-gram	115	167	109		
200k cuvinte	1-gram	2036	1379	1060	14.2	36.4
	2-gram	317	308	181	5.90	23.2
	3-gram	133	195	118	4.62	20.5
	4-gram	119	176	111	n/a	
	5-gram	116	172	110		

Tabelul 4.5 Evaluarea modelelor de limba în contextul RAV

4.3.3 Reevaluarea laticelor folosind modele de limba n -gram

Modelele de limbă descrise și evaluate în secțiunea 4.3.2, împreună cu cel mai bun model acustic bazat pe DNN, au fost utilizate pentru experimentele prezentate în această secțiune. Scopul a fost acela de a folosi tehnica reevaluării grafului cu transcrieri alternative (*lattice rescoring*) pentru a obține un sistem RVC-VE care este mult mai precis (WER scăzut) și decodează vorbirea mult mai eficient (folosește mai puțină memorie și decodarea are loc mai rapid). Un alt scop a fost acela de a găsi cea mai bună combinație de modele de limbă: unul pentru decodare și altul pentru reevaluare. Decodarea s-a făcut cu modele de limbă de ordinul 1,2 sau 3, în timp ce reevaluarea laticelor s-a realizat cu modele de limbă cu grad mai mare. De exemplu, pentru situația când decodarea a fost făcută cu un model de limba de tip 2-gram, reevaluarea s-a făcut cu modele de limba de tip 3-gram, 4-gram și 5-gram.

Modelele de limbă folosite la reevaluare au întotdeauna aceeași dimensiune a vocabularului cu cele folosite la decodare, proces în urma căruia se obține graful de transcrieri alternative (lattice). Folosirea unor modele de limba cu vocabular de dimensiuni diferite nu ar fi ajutat: cuvintele suplimentare ce s-ar fi găsit în latticea creată cu un model de limba mare, s-ar fi pierdut atunci când reevaluarea ar fi fost făcută cu un model de limba mai mic. Această observație este valabilă și invers: cuvintele suplimentare dintr-un model de limba mare, folosit la reevaluare, ar fi fost inutile, dacă latticea nu le conține.

Tabelul 4.6 compară rezultatele obținute direct la decodare cu rezultatele obținute după ce s-a aplicat tehnica de reevaluare a laticelor. Comparăția are loc prin varierea mai multor tipuri de modele de limbă folosite la decodare (diverse dimensiuni ale vocabularului și diverse ordine n -gram). Ordinul modelelor de limbă folosite la reevaluare nu este listat în Tabelul 4.6 deoarece rezultatele obținute prin varierea ordinului modelului de limbă folosit la reevaluare sunt foarte similare (tabelul arată cel mai bun rezultat care este cu cel puțin 5% mai bun decât cel mai slab rezultat). Observația precedentă este validă în toate cazurile, cu o singură excepție: reevaluarea cu un model de limba 2-gram, a unei lattice produsă cu un model de limba 1-gram, generează un rezultat care este mult mai slab decât toate celelalte opțiuni de reevaluare.

Date fiind rezultatele din Tabelul 4.6, se pot face câteva observații. Dacă decodarea se realizează cu un model de limba de tip 1-gram sau 2-gram, aplicarea tehnicii de reevaluare a laticelor este

foarte importantă: WER după reevaluare este cu 23% - 73% mai scăzut pe vorbire citită și cu 13% - 38% mai scăzut pe vorbire conversațională. Dacă decodarea este făcută cu un model de limbă 3-gram, reevaluarea laticelor nu aduce o îmbunătățire semnificativă.

		WER[%]		WER[%]	
		RSC-eval	SSC-eval	RSC-eval	SSC-eval
Model de limbă		Fără reevaluare		Cu reevaluare	
100k cuvinte	1-gram	15.0	36.5	6.06	22.5
	2-gram	6.44	23.4	5.04	20.3
	3-gram	5.18	20.6	5.05	20.1
150k cuvinte	1-gram	14.6	36.4	5.81	22.4
	2-gram	6.26	23.3	4.85	20.3
	3-gram	5.00	20.5	4.85	20.1
200k cuvinte	1-gram	14.2	36.4	5.39	22.4
	2-gram	5.90	23.2	4.49	20.2
	3-gram	4.62	20.5	4.48	20.0

Tabelul 4.6 WER[%] folosind tehnica de reevaluare a laticelor cu modele n-gram

Comparând rezultatele de la reevaluare, obținute în urma decodării cu modele de limbă 2-gram și 3-gram, indiferent de dimensiunea vocabularului, se observă că acestea sunt foarte similare. Prin urmare, nu are rost ca decodarea să fie făcută cu un model 3-gram; din punct de vedere al acurateții este aproximativ similar cu situația în care decodarea s-ar face cu un model de limbă 2-gram, urmat de reevaluare cu un model de limbă de ordin mai mare. Mai mult, din punct de vedere al resurselor de calcul și al vitezei de decodare, Tabelul 4.7 arată că decodarea cu model de limbă 2-gram este mult mai eficientă. Memoria maximă necesară pentru a stoca graful de decodare este de aproximativ 8.5 GB la decodarea cu model de limbă 2-gram și în jur de 30 GB la decodarea cu model de limbă 3-gram. Aceste valori sunt foarte similare pentru vocabular cu dimensiunea de 100.000 cuvinte până la 200.000 cuvinte. Viteza de decodare este măsurată în Tabelul 4.7, metrica utilizată fiind factorul de timp real (xRT). Acesta este calculat ca timpul necesar la decodarea corpusului de evaluare, împărțit la durata totală a acestuia. Valorile factorului de timp real afișate în Tabelul 4.7 au fost obținute prin decodarea pe o singură stație de lucru cu procesor Intel Xeon 3.20 GHz, 16 nuclee și 192 GB memorie RAM, fără a utiliza și procesorul grafic NVIDIA Quadro M4000. În acest context, valorile factorului de timp real de la decodare variază pentru *RSC-eval* între 0.04 și 0.06, în funcție de ordinul modelului de limbă, respectiv 0.08 și 0.10, în funcție de ordinul modelului de limbă, pentru *SSC-eval*. Reevaluarea laticelor crește factorul de timp real cu 10% până la 20%.

Model de limbă	Memorie maximă decodare	Factor de timp real [xRT]	
		RSC-eval	SSC-eval
1-gram	~ 1.5 GB	0.04	0.08
2-gram	~ 8.5 GB	0.05	0.08
3-gram	~ 30 GB	0.06	0.10

Tabelul 4.7 Consumul de memorie și factorul de timp real

4.3.4 Reevaluarea laticelor folosind modele de limbă RNN

Reevaluarea grafului cu transcrieri alternative s-a realizat de asemenea și cu ajutorul modelelor de limbă antrenate cu rețele neuronale recurente. Deoarece, cel puțin teoretic, modelele de limbă RNN codează o istorie a cuvintelor de lungime infinită, compilarea lor într-un graf de decodare este imposibilă. Din acest motiv, modelele RNN nu pot fi folosite direct în etapa de decodare,

dar în schimb ele sunt utilizate la reevaluarea laticelor. Deși modelele RNN au cu totul alt mod de antrenare față de modelele probabilistice, un parametru ce a fost variat în aceste experimente îl reprezintă ordinul n -gram maxim. Această valoare are rolul de a limita expansiunea exponențială a laticelor în urma procesului de reevaluare. Astfel, dacă pentru două stări din graf există un istoric identic de n cuvinte, cele două ramuri identice din graf sunt contopite în una singură. Tabelul 4.8 prezintă o comparație între WER [%] obținut cu reevaluarea laticelor folosind modele RNN, al căror ordin n -gram maxim a fost variat.

		WER[%]		WER[%]		WER[%]		WER[%]	
		RSC-eval	SSC-eval	RSC-eval	SSC-eval	RSC-eval	SSC-eval	RSC-eval	SSC-eval
Model de limbă decodare		2-gram RNN		3-gram RNN		4-gram RNN		5-gram RNN	
100k cuvinte	1-gram	7.79	27.07	7.07	25.41	6.90	25.26	6.86	25.26
	2-gram	5.24	21.84	4.63	20.64	4.59	20.43	4.57	20.38
	3-gram	4.66	20.48	4.33	19.51	4.27	19.43	4.26	19.27
150k cuvinte	1-gram	7.63	27.07	6.96	25.38	6.73	25.20	6.68	25.08
	2-gram	5.20	21.86	4.72	20.58	4.56	20.33	4.59	20.24
	3-gram	4.48	20.29	4.23	19.52	4.04	19.21	4.00	19.16
200k cuvinte	1-gram	7.22	27.06	6.41	25.41	6.26	25.22	6.24	25.05
	2-gram	4.54	21.77	4.04	20.48	3.86	20.17	3.89	20.15
	3-gram	4.02	20.13	3.69	19.41	3.53	19.21	3.48	19.15

Tabelul 4.8 WER[%] folosind tehnica de reevaluare a laticelor cu modele RNN

Se observă că varierea dimensiunii vocabularului, precum și a ordinului n -gram al modelului folosit la decodare, conduce la îmbunătățiri relative între modelul 2-gram RNN și 5-gram RNN, cu valori între 8% - 14% pe setul *RSC-eval* și 5% - 7% pe setul *SSC-eval*. Reevaluarea cu un model mai bun ajută mai mult pentru setul de evaluare cu vorbire citită, dar nu și pentru cel cu vorbire spontană. Acest lucru indică faptul că modelul de limbă nu are o influență atât de mare asupra rezultatelor pe vorbire spontană, modelul de limbă fiind cel care ar trebui îmbunătățit în acest sens. De asemenea, între cel mai bun rezultat obținut în urma reevaluării cu modele probabilistice și cel mai bun rezultat obținut cu RNN, îmbunătățirea relativă este de 22% pe setul *RSC-eval* și 4% pe *SSC-eval*, ambele în favoarea modelelor antrenate cu rețele neuronale recurente.

4.3.5 Sumarizarea rezultatelor finale și concluzii

Tabelul 4.9 prezintă o comparație între sistemul RVC-VE precedent, creat cu ajutorul utilitarului CMU Sphinx și sistemul RVC-VE actual, creat atât pe baza utilitarului Kaldi (folosind platforma HMM-GMM și DNN) cât și CMU Sphinx (HMM-GMM).

O scădere semnificativă a WER poate fi observată comparând cea de-a doua și prima linie din tabel, în special în cazul setului *SSC-eval* (17.5% îmbunătățire relativă). Această îmbunătățire a fost în special obținută datorită extinderii corpusului de antrenare: *SSC-train2*, ce conține 103 ore de vorbire conversațională a fost folosit la antrenare. Astfel, adăugând un nou corpus de vorbire spontană, acest fapt a condus la o scădere pronunțată a WER pe setul de evaluare de vorbire spontană.

O diferență semnificativă în WER poate fi de asemenea observată comparând rezultatele obținute de modelul acustic antrenat cu HMM-GMM folosind Kaldi, față de CMU Sphinx (liniile 2 și 3 în Tabelul 4.9). Modelele antrenate cu Kaldi au obținut îmbunătățiri relative ale WER de 28.4% (pe vorbire citită) și 17.9% (pe vorbire conversațională). Îmbunătățirile au fost

obținute datorită transformărilor suplimentare aplicate trăsăturilor acustice (LDA și MMLT) și datorită tehnicilor de antrenare (SAT, MMI) disponibile în Kaldi.

Folosind modele acustice bazate pe DNN, în contrast cu modelele acustice bazate pe HMM-GMM, în Kaldi s-a obținut o scădere relativă a WER cu 30.8% (pe vorbire citită) și 20.7% (pe vorbire conversațională).

Utilizarea modelelor de limbă complexe și a tehnicii de reevaluare a laticelor folosind modele n-gram (liniile 4 și 5 în Tabelul 4.9) au adus îmbunătățiri semnificative numai pe vorbire citită (27.4% scădere relativă). Reevaluarea laticelor cu modele antrenate pe baza rețelelor neuronale recurente a condus la o nouă scădere a WER, fiind obținută o îmbunătățire relativă de 22% pe vorbire citită și 5% pe vorbire spontană.

Per total, îmbunătățirile relative ale sistemului RVC-VE actual, față de cel din 2014, datorate extinderii corpusului de antrenare, a tuturor factorilor ce țin de modelarea acustică și a tuturor factorilor ce țin de modelarea de limbă, sunt de 76.5% pe vorbire citită, respectiv 51% pe vorbire spontană.

Sistemul RVC-VE al Speed		WER [%]	
Model acustic	Model de limbă	RSC-eval	SSC-eval
HMM-GMM (CMU Sphinx, 2014)	64k cuvinte, 3-gram	14.8	39.1
HMM-GMM (CMU Sphinx, 2017)	64k cuvinte, 3-gram	12.6	32.3
HMM-GMM (Kaldi, 2017)	64k cuvinte, 3-gram	9.0	26.4
DNN (Kaldi, 2017)	64k cuvinte, 3-gram	6.2	21.0
	200k cuvinte, 2-gram (dec), 4-gram (rescr)	4.5	20.2
	200k cuvinte, 3-gram (dec), 5-gram RNN (rescr)	3.48	19.15

Tabelul 4.9 Compararea sistemelor RVC-VE

4.4 APLICAȚIA DEMONSTRATIVĂ DE RVC-VE

Cea mai bună configurație a modelului acustic bazat pe rețele neuronale profunde (Tabelul 4.3) și cea mai bună pereche de modele de limbă pentru decodare, respectiv reevaluare (Tabelul 4.6) împreună cu platforma de tip client-server (creată de Tanel Alumăe și distribuită sub licența BSD simplificată [67]) au fost integrate într-o aplicație de transcriere în timp real a vorbirii în text pentru limba română, disponibilă online [75]. Interfața vizuală a aplicației în timpul unei sesiuni de transcriere este ilustrată în Figura 4.1.

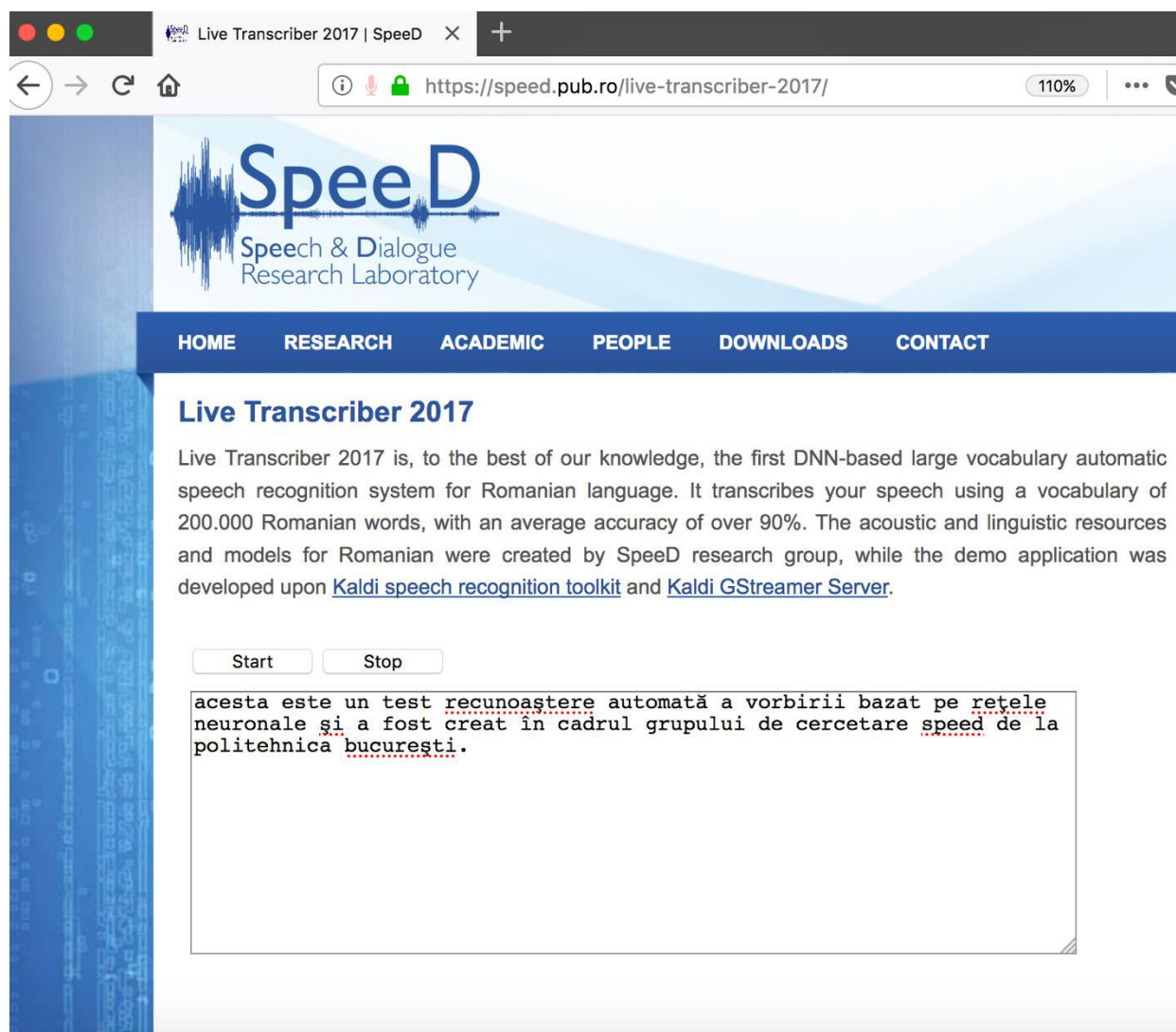
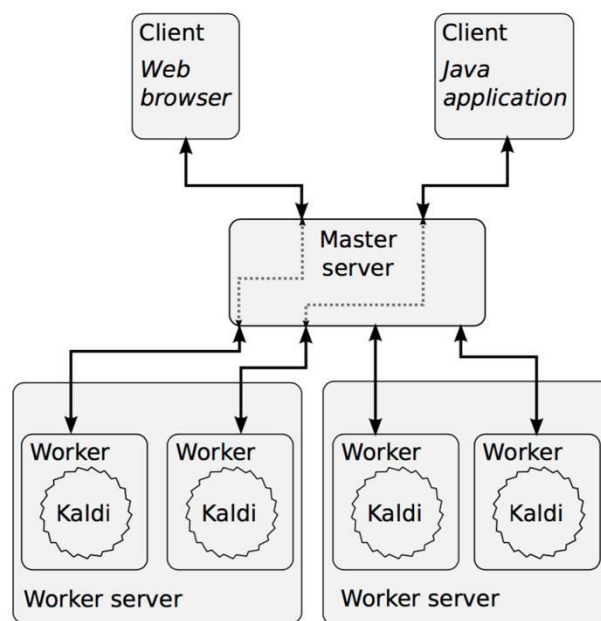


Figura 4.1 Interfața aplicației demonstrative de RAV

Aplicația are o arhitectură de tip client-server, unde clientul rulează într-un navigator web, fiind responsabil de capturarea fluxului audio, trimiterea lui spre server și afișarea transcrierilor primite înapoi de la server. Serverul este compus dintr-un server central (*master*), ce realizează comunicația cu aplicația client și cu grup de servere ce realizează transcrierea propriu-zisă (*workers*), folosind modelul acustic, fonetic și lingvistic. Acestea pot rula pe aceeași mașină fizică sau pe mai multe mașini din rețea. Serverul central funcționează ca un intermediar: redirecționează fluxul audio către un server de transcriere disponibil și textul obținut înapoi către client. Aceste componente software (aplicația client, serverul central, serverele de transcriere) nu trebuie să se afle obligatoriu pe aceeași mașină fizică, ele fiind capabile să lucreze și în mod distribuit. Numărul clienților care pot fi serviți simultan este egal cu numărul serverelor de transcriere. Arhitectura sistemului este ilustrată în Figura 4.2:

**Figura 4.2 Arhitectura aplicației RAV**

Sursa: [67]

4.5 EXPERIMENTE DE ANTRENARE NESUPERVIZATĂ A MODELELOR ACUSTICE

Deoarece crearea manuală a corpusurilor de vorbire adnotată implică o investiție semnificativă de timp și efort, interesul față de tehnicile de adnotare automată a crescut. Adnotarea automată a vorbirii presupune colectarea vorbirii în format brut, fără etichetări, și folosirea unei metode automate pentru a produce transcrieri precise pentru cel puțin o parte din corpusul inițial. Părțile din corpus care au fost transcrise corect pot fi folosite în viitor la antrenarea sistemelor RAV. Metode variate de adnotare automată, dezvoltate de-a lungul timpului, folosesc transcrieri aproximative, scoruri de încredere la nivel de cuvânt sau sisteme RAV complementare.

4.5.1 Metoda de adnotare automată

Metoda folosită la adnotarea automată a avut ca scop obținerea unui corpus de vorbire adnotat într-un mod automat, nesupervizat, oferind în același timp o transcriere de înaltă calitate. S-a intenționat ca noul corpus obținut să fie folosit la reantrenarea sistemelor RAV deja existente, crescând astfel variabilitatea acustică a modelelor și îmbunătățind astfel calitatea transcrierilor. Pașii corespunzători metodei sunt ilustrați în Figura 4.2.

Ideea principală a metodei de adnotare automată este următoarea:

- două sisteme RAV inițiale sunt folosite pentru a produce transcrieri pentru corpusul de vorbire neadnotată;
- transcrierile sunt aliniate și părțile identice sunt selectate ca fiind corecte;
- transcrierile selectate împreună cu părțile audio corespunzătoare sunt utilizate pentru a crea un nou corpus de vorbire adnotat.

Pentru că metoda de mai sus să funcționeze, factorul cheie este reprezentat de complementaritatea celor două sisteme RAV inițiale. Complementaritatea poate fi obținută prin folosirea unor modele acustice și lingvistice diferite, antrenarea pe seturi de date distincte, metode de antrenare sau decodare diferite, etc. Mai exact, este necesar ca cele două sisteme să facă numai greșeli necorelate. În aceste experimente, complementaritatea a fost dată de:

1. tipul modelului acustic folosit (HMM-GMM și DNN);
2. dimensiunea vocabularului (64.000 cuvinte și 200.000 cuvinte);
3. modelul de limbă folosit la decodare (3-gram și 2-gram);
4. utilizarea/non-utilizarea tehnicii de reevaluare a laticelor.

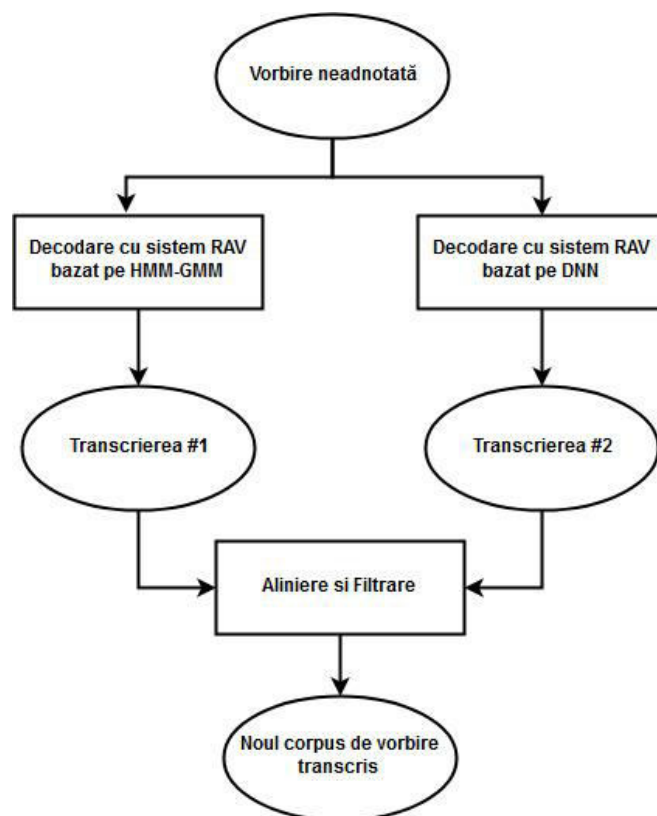


Figura 4.3 Metodă de adnotare automată a vorbirii

În acest context, primul pas în metoda propusă este crearea sistemelor RAV complementare. Al doilea pas presupune colectarea unui corpus de vorbire neadnotat. Mass-media produce material nou în fiecare zi, ce constă în spectacole, buletine de știri, interviuri, reportaje, acestea fiind o foarte bună sursă de vorbire. Publicarea lor în mediul on-line permite o implementare automată a procesului; strângerea unei cantități semnificative de date devine numai o problemă de timp, fără niciun alt efort. Al treilea pas presupune transcrierea noului corpus folosind cele două sisteme RAV complementare. Ipotezele sunt aliniate cu ajutorul unui algoritm bazat pe Dynamic Time Warping (DTW), selectând părțile comune. Așa cum o arată experimentele, probabilitatea ca ambele sisteme să facă erori comune este foarte mică. Secvențe consecutive de cuvinte, ce conțin un număr de caractere mai mare decât un prag determinat experimental, sunt considerate ca fiind transcrise corect. Un alt criteriu folosit este durata secvenței audio, ce trebuie să depășească un anumit prag. Totodată, distanța între două cuvinte trebuie să respecte o anumită încadrare temporală, pentru a elimina posibilitatea existenței unor cuvinte intermediare netranscrise.

În final, după ce se realizează selecția secvențelor de cuvinte corecte, etichetele de timp oferite de cel mai bun sistem RAV sunt folosite pentru a tăia părțile audio corespunzătoare secvențelor aliniate din semnalul audio inițial.

Din punct de vedere al evaluării metodei de adnotare, criteriul de performanță al acestei metode de adnotare automată este dat de doi factori: cantitatea de vorbire selectată după aliniere, raportat la dimensiunea totală a corpusului neadnotat și calitatea adnotării, măsurabilă în rata de eroare la nivel de cuvânt (WER) și rata de eroare la nivel de caracter (ChER). Aceste metrici de performanță pot fi calculate folosind un corpus de vorbire pentru care deja există o adnotare de referință, împreună cu etichete temporale la nivel de cuvânt. Adnotarea de referință poate fi obținută în două moduri: creată manual sau folosind un sistem RAV care efectuează aliniere forțată pe corpusul de evaluare. Deși a doua metodă este mai predispusă la a genera erori dacă sistemul RAV nu este destul de performant, alinierea forțată are o acuratețe destul de bună. Astfel, corpusul este considerat la început ca fiind un corpus neadnotat, ce va fi transcris cu ambele sisteme RAV, obținând ipotezele ce vor fi aliniate. Bazându-se pe etichetele temporale

ale părților aliniate, se selectează părțile corespunzătoare din transcrierea de referință. În acest fel, WER și ChER pot fi calculate între aceste două seturi de text. De asemenea, pe baza etichetelor de timp ale părților aliniate, durata datelor selectate poate fi calculată.

4.5.2 Pregătirea experimentelor

Sistemele RAV complementare, folosite la experimentele de adnotare automată a vorbirii sunt două dintre sistemele descrise în secțiunile 4.2 și 4.3. Pentru antrenarea și evaluarea acestora, au fost utilizate corpusurile de vorbire pentru limba română, prezentate anterior în secțiunea 4.2.3: *Read Speech Corpus* (RSC) ce conține vorbire citită colectată în condiții de laborator, fără zgomot de fundal și *Spontaneous Speech Corpus* (SSC), ce conține vorbire continuă, spontană, preluată din emisiuni de radio și televiziune, uneori afectată de zgomot. Ambele corpusuri sunt compuse din fișierele audio împreună cu transcrierile corespunzătoare și sunt împărțite în seturi de antrenare și evaluare.

Corpusul de vorbire neadnotată a fost procurat din mass-media românească, mai exact de la două web site-uri de știri și un post de radio. A fost implementată o aplicație care parcurge fluxul RSS al acestor web site-uri, extrage în mod automat fișiere audio și realizează eșantionarea la 16 KHz, 16 biți pe eșantion. Durata totală a corpusului este de 130 ore. Detalii despre toate aceste corpusuri de vorbire se găsesc în Tabelul 4.10.

Scop	Set	Durată	
Antrenare	RSC-train	94 h 46 m	225 h 30 m
	SSC-train	130 h 44 m	
Evaluare	RSC-eval	5 h 29 m	8 h 58 m
	SSC-eval	3 h 29 m	
Adnotare	Sursa #1	65 h 55 m	136 h 41 m
	Sursa #2	51 h 27 m	
	Sursa #3	19 h 19 m	

Tabelul 4.10 Corpusurile de vorbire folosite în procesul de adnotare automată

Modelele acustice ale celor două sisteme RAV complementare au fost antrenate pe baza a două paradigme distincte: HMM-GMM și DNN. Ambele sisteme folosesc ca trăsături acustice vectori de coeficienți cepstrali, 13-dimensional, împreună cu derivatele de ordinul 1 și 2. Coeficienții au fost extrași cu o fereastră Hamming cu o durată de 25 ms. Modelul acustic bazat pe DNN este de asemenea antrenat cu iVectori de dimensiune 100. Ambele sisteme modelează 36 de foneme ale limbii române, dependente de context.

Modelul acustic bazat pe HMM-GMM a fost antrenat folosind 4.000 stări acustice (senone), fiecare dintre ele fiind modelate de un număr de 128 densități gaussiene. Modelul acustic bazat pe DNN este construit pe baza alinierilor obținute de la un model de tip HMM-GMM. Arhitectura DNN folosită este TDNN, stratul de intrare fiind alcătuit din 3.500 neuroni și procesează 9 cadre de semnal la un moment de timp. Rețeaua are 6 straturi ascunse cu 1.200 neuroni pe fiecare. Stratul de ieșire constă în 350 neuroni. Modelul a fost antrenat pe durata a 5 epoci, cu o rată de învățare inițială de 0.015 și o rată de învățare finală de 0.00015. Setul de antrenare a fost divizat în loturi cu dimensiunea 512.

Modelele de limbă folosite în ambele sisteme RAV sunt probabilistice, de tip n-gram. La crearea lor s-au folosit texte din mediul online, transcrieri ale unor buletine de știri și conferințe. Acestea au fost interpolate cu ponderea 0.5, iar primul corpus conține 315M cuvinte, în timp ce al doilea conține 40M cuvinte. Sistemul RAV bazat pe HMM-GMM folosește un model de limba 3-gram cu 64.000 cuvinte, în timp ce sistemul RAV bazat pe DNN folosește un model de limba 2-gram cu 200.000 cuvinte la decodare și un model de limba 4-gram cu 200.000 cuvinte pentru reevaluarea laticelor.

Tabelul 4.11 prezintă detalii despre cele două sisteme RAV folosite la procesul de adnotare automată, precum și performanțele lor.

Model acustic	Model de limbă	WER[%]	
		RSC-eval	SSC-eval
HMM - GMM	64K cuvinte, 3-gram	12.6	32.3
HMM - DNN	200k cuvinte, 2-gram (dec), 4-gram (rescr)	4.5	20.2

Tabelul 4.11 Sistemele RAV folosite la procesul de adnotare automată

4.5.3 Rezultate experimentale

Așa cum a fost descris anterior, evaluarea metodei de adnotare automată a fost efectuată prin compararea transcrierii ipotetice rezultată în urma aplicării metodei pe corpusul de evaluare cu transcrierea de referință. Transcrierile generate automat nu acoperă întreg corpusul de evaluare. Prin urmare, comparația propusă mai sus trebuie să se efectueze pe o selecție a corpusului de referință.

Corpusul de evaluare a fost aliniat forțat și etichetele de timp rezultate au fost folosite (împreună cu etichetele de timp ale transcrierilor ipotetice) pentru a selecta părțile corespunzătoare din transcrierea de referință.

Tabelul 4.12 prezintă detalii despre părțile aliniate: rate de eroare la nivel de cuvânt și caracter, durata datelor aliniate și numărul cuvintelor aliniate. Rezultatele arată că metoda de adnotare automată produce corpus de vorbire de o calitate înaltă: 99% dintre cuvintele selectate sunt corecte. Ratele de eroare sunt cu un ordin de mărime mai mici decât cele raportate în [44] și [62]. Acest fapt confirmă eficiența celor două sisteme RAV: erorile făcute de acestea sunt identice numai într-o mică măsură, iar părțile transcrise corect pot fi considerate corecte în proporție de 99%. În concluzie, folosirea a două sisteme RAV care diferă prin tipul modelului acustic (HMM-GMM și DNN) este mult mai eficientă decât folosirea sistemelor RAV care diferă prin alte caracteristici (seturi de antrenare diferite sau metodă diferită de decodare).

Corpus	WER [%]	ChER [%]	# ore aliniate	% ore aliniate
RSC-eval	1.0	0.3	2.62	48%
SSC-eval	1.3	0.4	0.69	20%

Tabelul 4.12 Evaluarea metodei de adnotare automată

Tabelul 4.13 prezintă cantitatea de date obținută după ce metoda a fost aplicată asupra corpusului de vorbire neadnotată, nou achiziționat. De la prima sursă, în jur de 50% din cantitatea totală a datelor procurate au fost adnotate. Pentru cea de-a doua sursă, avem un procent de 31%, în timp ce în cazul celei de-a treia surse, 20% din date au fost păstrate.

Corpus	# ore aliniate	% ore aliniate
Sursa #1	32 h 53 m	50%
Sursa #2	10 h 00 m	31%
Sursa #3	06 h 20 m	20%

Tabelul 4.13 Rezultatele adnotării automate asupra corpusului nou achiziționat

Corpusul de vorbire nou adnotat (49 ore, cum se poate observa în Tabelul 4.13) a fost adăugat la corpusul de antrenare al RAV existent deja. Ambele sisteme RAV au fost reantrenate și evaluate, rezultatele fiind prezentate în Tabelul 4.14. În mod surprinzător, creșterea corpusului de antrenare a fost benefică numai pentru modelul bazat pe HMM-GMM: pe ambele seturi de evaluare, noul sistem HMM-GMM a obținut îmbunătățiri relative minore. Situația diferă însă în cazul sistemului bazat pe DNN, care a obținut rezultate mai slabe.

Model acustic	WER[%]		Relative WER improvement [%]	
	RSC-eval	SSC-eval	RSC-eval	SSC-eval
HMM-GMM	11.7	31.3	7.14	3.10
DNN	5.0	20.9	-10.67	-3.47

Tabelul 4.14 Performanțele noilor sisteme RAV după reantrenare

4.5.4 Concluzii

Secțiunea 4.5 a avut ca scop prezentarea unei metode de adnotare automată a corpusurilor de vorbire, folosind transcrieri de la două sisteme RAV complementare. Experimentele au arătat că sistemele RAV ce folosesc modele acustice de tip HMM-GMM și DNN, produc semnificativ

mai puține erori identice decât erorile comparate cu sistemele RAV complementare care folosesc numai modele de tip HMM-GMM. În primul caz, WER este în jur de 1%, în timp ce experimentele anterioare [44] prezintă un WER de 10%.

Se poate observa faptul că sistemele RAV inițiale au o acuratețe foarte bună, fiind antrenate cu o mare cantitate de date. Acesta poate fi motivul pentru care renatrenarea sistemelor, adăugând nou corpus obținut prin această abordare nesupervizată, nu aduce îmbunătățiri semnificative. Din cantitatea totală de date achiziționate, un procent de 36% au fost adnotate corect. După reantrenarea modelelor acustice, sistemul RAV bazat pe HMM-GMM a obținut o mică îmbunătățire, în timp ce sistemul RAV bazat pe DNN a pierdut puțin în ceea ce privește performanța. Acest fapt va fi investigat ulterior, prin experimente în care sistemele inițiale vor fi antrenate cu mai puține date.

CONCLUZII FINALE

Lucrarea de față a avut ca scop principal îmbunătățirea unui sistem RVC-VE anterior pentru limba română, folosind rețele neuronale pentru antrenarea modelului acustic și lingvistic. Totodată, s-a dorit creșterea corpusurilor în limba română ce conțin vorbire adnotată, folosind o procedură automată de colectare și selecție. Evaluarea acurateții și a performanței, precum și implementarea celei mai bune configurații a sistemului într-o aplicație de transcriere a vorbirii în timp real, au reprezentat unul dintre obiectivele urmărite.

Capitolul 1 a prezentat o introducere de ansamblu în domeniul rețelilor neuronale. Au fost prezentate caracteristicile de bază, modul de funcționare, precum și câteva arhitecturi de rețele, alături de particularitățile lor.

Capitolul 2 a amintit noțiunile fundamentale de la baza unui sistem de recunoaștere automată a vorbirii. Au fost prezentate modulele componente și modul în care acestea sunt create.

Capitolul 3 a constatat în studiul asupra stării artei în ceea ce privește antrenarea nesupervizată a modelelor acustice. Au fost descrise principalele metode de adnotare automată a vorbirii. Plecând de la premisa că două sisteme RAV care transcriu același corpus de vorbire vor face erori diferite, s-a ajuns la concluzia că folosirea sistemelor RAV complementare poate produce cu succes corpusuri de vorbire. Condiția este ca aceste sisteme să prezinte un grad de diversitate crescut. Cei mai importanți factori care pot face ca sistemele să fie complementare sunt seturile diferite de date de antrenare, tipul trăsăturilor extrase din semnalul vocal, tipul modelelor utilizate, algoritmii de antrenare și decodare. În ceea ce privește alinierea și filtrarea transcrierilor obținute, se observă tendința folosirii unor tehnici de învățare automată.

Capitolul 4 s-a axat pe un set extins de experimente de recunoaștere automată a vorbirii pentru limba română. S-a insistat asupra îmbunătățirilor aduse la nivelul trăsăturilor vocale, al modelelor acustice și al modelelor de limbă. Folosirea rețelilor neuronale a adus cele mai mari performanțe. La nivelul modelului acustic, s-a înregistrat o scădere a ratei de eroare la nivel de cuvânt de 30%, în cazul evaluării sistemului pe vorbire citită, respectiv 20% pentru vorbire spontană. Extinderea vocabularului și a complexității modelului de limbă a avut un impact major în ceea ce privește performanța. Cel mai bun model lingvistic bazat pe rețele neuronale a înregistrat îmbunătățiri de 22%, respectiv 4%, față de modelele statistice.

Îmbunătățirea relativă a sistemului RVC-VE actual față de sistemul anterior, creat în 2014 este de 76.5% pe vorbire citită și 51% pe vorbire spontană. Ea se datorează extinderii corpusului de vorbire și a tuturor optimizărilor ce țin de modelarea acustică și lingvistică.

Metoda de adnotare automată a vorbirii folosită în această lucrare a dus la selecția a 20% -50% din durată totală a corpusurilor audio colectate, în funcție de sursa de la care acestea provin. Evaluarea meotdei a demonstrat că adnotarea se face cu o precizie de 99%.

CONTRIBUȚII PERSONALE

Lucrarea de față a reușit să îndeplinească cu succes obiectivele stabilite. Contribuțiile personale ale autorului pot fi sumarizate după cum urmează:

1. Crearea sistemului RVC-VE pentru limba română

- 1.1 Crearea infrastructurii de calcul necesară lucrului distribuit folosind procesoare grafice
- 1.2 Modelare acustică
 - 1.2.1 extinderea corpusului de vorbire
 - 1.2.2 utilizarea unor noi tehnici de prelucrare a trăsăturilor acustice
 - 1.2.3 aplicarea unor noi tehnici de optimizare în procesul de antrenarea a modelului acustic de tip HMM-GMM
 - 1.2.4 antrenarea modelului acustic folosind rețele neuronale profunde
- 1.3 Modelare lingvistică
 - 1.3.1 extinderea dimensiunii vocabularului în vederea obținerii unui model de limbă mai robust
 - 1.3.2 crearea de modele de limbă complexe, ce țin cont de un istoric mai extins al cuvintelor
 - 1.3.3 antrenarea modelului de limba folosind rețele neuronale profunde
 - 1.3.4 utilizarea tehnicii de reevaluare a laticelor
- 1.4 Reantrenarea sistemului RVC-VE anterior folosind algoritmi actualizați
- 1.5 Compararea sistemului RVC-VE anterior cu sistemul curent
- 1.6 Evaluarea sistemului RVC-VE curent
- 1.7 Analiza rezultatelor și a resurselor de calcul necesare
- 1.8 Configurarea aplicației demonstrative de RVC-VE

2. Experimente de antrenare nesupervizată a modelelor acustice

- 2.1 Studiul literaturii de specialitate în problematica adnotării automate a vorbirii
- 2.2 Evaluarea metodei de adnotare
- 2.3 Implementarea aplicației de extragere a datelor audio de pe Internet
- 2.4 Colectarea corpusului de vorbire de pe Internet în mod automat
- 2.5 Transcrierea corpusului folosind două sisteme RAV complementare
- 2.6 Alinierea și filtrarea transcrierilor, obținerea unui nou corpus adnotat automat
- 2.7 Reantrenarea și evaluarea modelelor acustice folosind noul corpus de vorbire

BIBLIOGRAFIE

- [1] X Zhang, J Trmal, Daniel Povey, and S Khudanpur, "Improving Deep Neural Network Acoustic Models using Generalized Maxout Networks," *Icassp*, 2014.
- [2] Alexander Waibel, Toshiyuki Hanazawa, Geoffrey E Hinton, Kiyohiro Shikano, and Kevin J Lang, Phoneme recognition using time-delay neural networks, 1989.
- [3] Ngoc Thang Vu, Franziska Kraus, and Tanja Schultz, "Multilingual A-stabil: A new confidence score for multilingual unsupervised training," in *2010 IEEE Workshop on Spoken Language Technology, SLT 2010 - Proceedings*, 2010.
- [4] Ngoc Thang Vu, Franziska Kraus, and Tanja Schultz, "Cross-language bootstrapping based on completely unsupervised training using multilingual A-stabil," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2011.
- [5] A. Buzo V. Popescu, C. Petrea, D. Haneș and C. Burileanu, "Spontaneous Speech Database for Romanian Language," 2008.
- [6] B. Tarján et al., "Broadcast news transcription in Central-East European languages," in *3rd IEEE International Conference on Cognitive Infocommunications, CogInfoCom 2012 - Proceedings*, 2012.
- [7] Takeda, Kazuya, Yoshinori Sagisaka and Shigeru Katagiri, "Acoustic-phonetic labels in a Japanese speech database," *European Conference on Speech Technology*, 1987.
- [8] T. Boroș Ș. D. Dumitrescu and R. Ion, "Crowd-Sourced, Automatic Speech-Corpora Collection-Building the Romanian Anonymous Speech Corpus," *CCURL 2014: Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era*, 2014.
- [9] Andreas Stolcke, "Srilm - an Extensible Language Modeling Toolkit," *Interspeech*, 2002.
- [10] Adriana Stan et al., "The SWARA speech corpus: A large parallel Romanian read speech dataset," in *2017 9th International Conference on Speech Technology and Human-Computer Dialogue, SpeD 2017*, 2017.
- [11] Adriana Stan, Junichi Yamagishi, Simon King, and Matthew Aylett, "The Romanian speech synthesis (RSS) corpus: Building a high quality HMM-based speech synthesis system using a high sampling rate," *Speech Communication*, 2011.
- [12] Anthony Rousseau, Paul Deléglise, and Yannick Estève, "TED-LIUM: an Automatic Speech Recognition dedicated corpus," *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, 2012.
- [13] Georg Rehm et al., "An Update and Extension of the META-NET Study "Europe's Languages in the Digital Age"," in *Proceedings of the Workshop on Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era. Workshop on Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era (CCURL-2014), May 26, Reykjavik, Icelan*, 2014.
- [14] Daniel Povey, Hong Kwang J. Kuo, and Hagen Soltau, "Fast speaker adaptive training for speech recognition," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2008.

- [15] D. Povey et al., "The Kaldi speech recognition toolkit," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.
- [16] Vijayaditya Peddinti, Daniel Povey, and Sanjeev Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2015.
- [17] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2015.
- [18] Ilya Oparin, Lori Lamel, and Jean Luc Gauvain, "Rapid development of a Latvian speech-to-text system," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2013.
- [19] T Mikolov, M Karafiat, L Burget, J Cernocky, and S Khudanpur, "Recurrent Neural Network based Language Model," *Interspeech*, 2010.
- [20] Zejun Ma, Xiaorui Wang, and Bo Xu, "Unsupervised training of subspace Gaussian mixture models for conversational telephone speech recognition," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2012.
- [21] C. Munteanu, M. Boldea and A. Doroga, "Design, Collection and Annotation of a Romanian Speech Database," *In Proceedings of the First LREC-Workshop on Speech Database Development for Central and Eastern European Languages*.
- [22] C. Munteanu, M. Boldea and A. Doroga, "Collection and Annotation of a Romanian Speech Database," *In Proceedings of the First LREC-Workshop on Speech Database Development for Central and Eastern European Languages*.
- [23] Martin Lojka and Jozef Juhár, "Hypothesis combination for Slovak dictation speech recognition," in *Proceedings Elmar - International Symposium Electronics in Marine*, 2014.
- [24] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition," *Bell System Technical Journal*, 1983.
- [25] Jozef Juhár Koctúr, Tomáš, Stanislav Ondáš, "Speech corpus generation based on N-gram confidence measure classification," *ELMAR International Symposium*, 2017.
- [26] Tomáš Koctúr, Ján Staš, and Jozef Juhár, "Unsupervised acoustic corpora building based on variable confidence measure thresholding," in *Proceedings Elmar - International Symposium Electronics in Marine*, 2016.
- [27] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel, "Joint factor analysis versus eigenchannels in speaker recognition," *IEEE Transactions on Audio, Speech and Language Processing*, 2007.
- [28] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigné, "Restructuring speech representations using a pitch-adaptive time-frequency smoothing and an instantaneous-frequency-based F0 extraction: Possible role of a repetitive structure in sounds," *Speech Communication*, 1999.
- [29] Yan Huang, Dong Yu, Yifan Gong and Chaojun Liu, "Semi-supervised GMM and DNN acoustic model training with multi-system combination and confidence re-calibration," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2013.
- [30] Xuedong Huang, Alex Acero, and Hsiao-Wuen Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development.*, 2001.
- [31] Andi Buzo, Horia Cucu and Corneliu Burileanu, "Unsupervised acoustic model training

- using multiple seed ASR systems," *in the Proceedings of the 4th International Workshop on Spoken Language Technologies for Under-resourced Languages (SLTU)*, pp. 124-130, 2014.
- [32] Gollan, Christian, et al. "An improved method for unsupervised training of LVCSR systems," *Eighth Annual Conference of the International Speech Communication Association*, 2007.
- [33] J J Godfrey, E C Holliman, and J McDaniel, "SWITCHBOARD telephone speech corpus for research and development," *Proceedings of the 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-92)*, 1992.
- [34] A. Kabir and M. Giurgiu, "A romanian corpus for speech perception and automatic speech recognition," vol. The 10th I, p. 2011.
- [35] Georgescu, A.L., Caranica, A., Cucu, H., Burileanu, C. "RoDigits - a Romanian Connected-Digits Speech Corpus for Automatic Speech and Speaker Recognition," *University Politehnica of Bucharest Scientific Bulletin (submitted to)*.
- [36] Alexandru Lucian Georgescu, Horia Cucu, and Corneliu Burileanu, "Speed's DNN approach to Romanian speech recognition," in *2017 9th International Conference on Speech Technology and Human-Computer Dialogue, SpeD 2017*, 2017.
- [37] Ș. A. Toma, G. Suciuc and R. Chevereșan, "Towards a continuous speech corpus for banking domain automatic speech recognition," *In Speech Technology and Human-Computer Dialogue (SpeD)*, 2017.
- [38] Jonathan G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," *Automatic Speech Recognition and Understanding*, 1997.
- [39] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society. Series B*, 1977.
- [40] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, 2011.
- [41] Horia Cucu, Andi Buzo and Corneliu Burileanu, "Unsupervised acoustic model training using multiple seed ASR systems," *Spoken Language Technologies for Under-Resourced Languages*, 2014.
- [42] Horia Cucu, Andi Buzo, Lucian Petrică, Dragoș Burileanu, and Corneliu Burileanu, "Recent improvements of the Speed Romanian LVCSR system," in *IEEE International Conference on Communications*, 2014.
- [43] Horia Cucu, "Proiect de cercetare-dezvoltare în Tehnologia Vorbirii," *Îndrumar de proiect*
- [44] Horia Cucu, Andi Buzo, Laurent Besacier, and Corneliu Burileanu, "Enhancing ASR systems for under-resourced languages through a novel unsupervised acoustic model training technique," *Advances in Electrical and Computer Engineering*, 2015.
- [45] H. Cucu, "Towards a speaker-independent, large-vocabulary continuous speech recognition system for Romanian," *PhD Thesis*, 2011.
- [46] L. A. Scutelnicu, A. D. Bibiri, D. Cristea, L. Pistol and A. Turculeț, "Romanian Corpus For Speech-To-Text Alignment," *In Proc. of the 9th International Conference on Linguistic Resources And Tools For Processing The Romanian Language*.
- [47] "https://mi.eng.cam.ac.uk/~mjfg/mjfg_NOW.pdf". Accesat la data de 21.06.2018.
- [48] "https://en.wikipedia.org/wiki/Normal_distribution". Accesat la data de 21.06.2018.
- [49] "<https://cmusphinx.github.io>". Accesat la data de 21.06.2018.
- [50] "<http://www.wildml.com/2015/09/recurrent-neuronal-networks-tutorial-part-1-introduction->

to-rnns/". Accesat la data de 21.06.2018.

- [51] Christian Szegedy et al., "Going deeper with convolutions," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
- [52] Karen Simonyan and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *International Conference on Learning Representations (ICRL)*, 2015.
- [53] Peng Shen et al., "Combination of multiple acoustic models with unsupervised adaptation for lecture speech transcription," *Speech Communication*, 2016.
- [54] Reza Sahraeian, Dirk Van Compernelle, and Febe De Wet, "Using generalized maxout networks and phoneme mapping for low resource ASR- a case study on Flemish-Afrikaans," in *Proceedings of the 2015 Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, PRASA-RobMech 2015*, 2015.
- [55] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychological Review*, 1958.
- [56] Petrică, L., Cucu, H., Buzo, A., & Burileanu, C., "A Robust Diacritics Restoration System Using Unreliable Raw Text Data," In *Spoken Language Technologies for Under-Resourced Languages*.
- [57] Warren S. McCulloch and Walter Pitts, "A logical calculus of the ideas immanent in nervous activity," *The Bulletin of Mathematical Biophysics*, 1943.
- [58] Sheng Li, Yuya Akita, and Tatsuya Kawahara, "Data selection from multiple ASR systems' hypotheses for unsupervised acoustic model training," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2016.
- [59] Jing Li, Ji-hang Cheng, Jing-yuan Shi, and Fei Huang, "Brief Introduction of Back Propagation (BP) Neural Network Algorithm and Its Improvement," in *Advances in Computer Science and Information Engineering: Volume 2.*, 2012.
- [60] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 1998.
- [61] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances In Neural Information Processing Systems*, 2012.
- [62] Denis Juvet and Dominique Fohr, "About combining forward and backward-based decoders for selecting data for unsupervised training of acoustic models," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2014.
- [63] Geoffrey Hinton et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, 2012.
- [64] Yao Haitao, Xu Ji, and Liu Jian, "Multi-lingual unsupervised acoustic modeling using multi-task deep neural network under mismatch conditions," in *Proceedings of 2016 8th IEEE International Conference on Communication Software and Networks, ICCSN 2016*, 2016.
- [65] Giulia Garau and Steve Renals, "Combining spectral representations for large-vocabulary continuous speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, 2008.
- [66] Peter Bell et al., "A lecture transcription system combining neural network acoustic and language models," in *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2013.
- [67] Tanel Alumäe, "Full-duplex Speech-to-text System for Estonian," in *Frontiers in Artificial Intelligence and Applications*, 2014.

- [68] Zeiler, Matthew D., and Rob Fergus "Visualizing and understanding convolutional networks" European conference on computer vision, Springer, Cham, 2014.
- [69] "<https://www.apple.com/shop/accessories/all-accessories/homekit>". Accesat la data de 21.06.2018.
- [70] "<https://www.safaribooksonline.com/library/view/deep-learning/9781491924570/ch04.html>". Accesat la data de 21.06.2018.
- [71] "https://www.researchgate.net/figure/Graph-of-a-recurrent-neural-network_fig3_234055140". Accesat la data de 21.06.2018.
- [72] "<https://www.microsoft.com/en-us/cortana>". Accesat la data de 21.06.2018.
- [73] "<https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html>". Accesat la data de 21.06.2018.
- [74] "<https://www.apple.com/ios/siri/>". Accesat la data de 21.06.2018.
- [75] "<https://speed.pub.ro/live-transcriber-2017/>". Accesat la data de 21.06.2018.
- [76] "<https://speed.pub.ro>". Accesat la data de 21.06.2018.
- [77] "<https://s3.amazonaws.com/slice-intelligence-prod/whitepapers/downloads/Amazon-Echo-Slice-White-Paper.pdf>". Accesat la data de 21.06.2018.
- [78] "<https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>". Accesat la data de 21.06.2018.
- [79] "<https://developer.nvidia.com/cuda-toolkit>". Accesat la data de 21.06.2018.
- [80] "<https://developer.amazon.com/alexa?cid=a>". Accesat la data de 21.06.2018.
- [81] "<http://www.scrigroup.com/educație/biologie/Elemente-de-baza-ale-calcululu13888.php>". Accesat la data de 21.06.2018.
- [82] "http://kom.aau.dk/group/04gr742/pdf/MFCC_worksheet.pdf". Accesat la data de 21.06.2018.
- [83] "<http://gridscheduler.sourceforge.net>". Accesat la data de 21.06.2018.