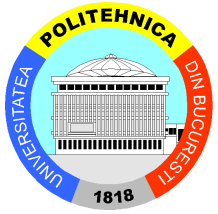

UNIVERSITATEA "POLITEHNICA" din BUCUREŞTI

FACULTATEA Electronică, Telecomunicații și Tehnologia Informației

CATEDRA Dispozitive, Circuite și Aparate Electronice

TEZĂ DE DOCTORAT

Recunoaşterea Limbajului Vorbit în Reţele de Telecomunicaţii Mobile

Autor: Ing. Andi Buzo

Conducător de doctorat: Prof. dr. ing. Corneliu Burileanu

București

2011

Mulțumiri

In primul rând, vreau să-i mulțumesc conducătorului tezei mele de doctorat, domnului prof. dr. ing. Corneliu Burileanu pentru că m-a orientat în carieră dându-mi sfaturi pe care astăzi sunt fericit că le-am ascultat. Îi mulțumesc pentru ajutorul pe care mi l-a dat în fiecare etapă de realizare a tezei de doctorat.

Mulțumesc membrilor distinși ai comisiei, doamnei prof. dr. ing. Inge Gavăt, domnului prof. dr. ing. m.c. al Academiei Române Horia-Nicolai Teodorescu, domnului prof. dr. ing. Gavril Toderean și domnului prof. dr. ing. Teodor Petrescu (președintele comisiei) pentru observațiile făcute și sfaturile utile la imbunătățirea calității tezei de doctorat.

Mulțumesc colegilor care m-au ajutat la realizarea tezei cu diverse materiale, sugestii și indicații. În mod deosebit vreau sa-i mulțumesc domnului ing. Horia Cucu (în curând doctor inginer) cu care am lucrat cot la cot și împreună am realizat aplicația de recunoaștere a limbajului vorbit în limba română.

Mulțumesc familiei mele care, deși a fost departe, am simțit-o foarte aproape. M-au sprijinit necondiționat ușurându-mi foarte mult munca: ”*ju dua shumë*!”.

În final aș dori să le mulțumesc prietenilor care au colorat timpul meu liber din această perioadă. Astfel, peste câțiva ani o să-mi aduc aminte de doctorat ca o perioada frumoasă a vieții mele.

Cuprins

Lista de figuri

Lista de tabele

CAPITOLUL 1

INTRODUCERE

1.1. OBIECTIVUL LUCRĂRII

Această teză de doctorat are două obiective principale. Primul obiectiv este dezvoltarea unui sistem de recunoaştere a limbajului vorbit (SRLV) în limba română. Problema principală, în acest sens, sunt resursele limitate pentru această limbă. În particular, nu există baze de date etichetate de dimensiuni mari şi nici baze de date standard care pot fi folosite ca referinţă. Din acest motiv, pentru a obţine rezultate satisfăcătoare în termeni de rată de recunoaştere se sacrifică dimensiunea vocabularului, mai ales în cazul recunoaşterii vorbirii continue. O altă problemă în recunoaşterea limbajului vorbit (RLV) în limba română este lipsa unor experimente amănunţite care să folosească ca referinţă de comparaţie pentru celelalte soluţii oferite. În această teză, sunt propuse câteva strategii de achiziţie a bazei date şi de antrenare a modelelor acustice. O serie de experimente detaliate au fost rulate variind diverşi parametri ai modelului acustic. Rezultatele experimentale permit găsirea configuraţiei optime, dar pot fi folosite şi ca referinţă pentru evaluarea imbunătăţirilor introduse în sistem prin diferite optimizări sau prin mărirea bazei de date.

Al doilea obiectiv este analiza comportării sistemului în condiţiile în care el, sau o parte din el, este implementat pe un dispozitiv portabil, iar semnalul vocal este trimis printr-o reţea de telecomunicaţii. Problemele principale care apar la implementarea pe dispozitive de telecomunicaţii, sunt cauzate de resursele limitate pe care le au aceste dispozitive. Din acest motiv, trebuie aduse optimizări care reduc complexitatea SRLV. La transmisia semnalului vocal printr-o reţea de telecomunicaţii, el poate suferi distorsiuni sub formă de pierderi de pachete şi de zgomot (zgomotul de fond este şi el inclus şi analizat în acest caz). În această lucrare, sunt propuse câteva soluţii care reduc puterea de calcul necesară fără să afecteze rata de recunoaştere. Experimentele făcute variind parametrii acustici ajută la determinarea configuraţiei optime, de data aceasta din punct de vedere al complexităţii sistemului. Un număr mare de experimente au fost făcute pentru evaluarea sistemului în condiţii adverse de pierderi de pachete şi de zgomot.

O parte importantă a acestei lucrări o ocupă studiul incertitudinii modelelor acustice şi a confuziei între foneme. Aceste informaţii au fost folosite atât pentru optimizarea sistemului din punct de vedere al ratei de recunoaştere, cât şi din punct de vedere al reducerii numărului de parametrii ai semnalului vocal şi implicit al reducerii numărului de calcule. Această soluţie a fost aplicată şi în cazul unui semnal afectat de zgomot şi de pierderi de pachete şi rezultatele experimentale au confirmat creşterea acurateţei de recunoaştere.

1.2. APLICAŢII ALE RLV

RLV găseşte o arie de aplicabilitate foare mare. Aceste aplicaţii se pot grupa în trei categorii mari. Dictarea este cea mai evidentă aplicaţie a RLV, pentru că scopul explicit al acestei sarcini este acela de a transcrie un semnal vocal. Ea presupune o recunoaştere a vorbirii continue şi poate fi generală sau dependentă de subiectul discuţiei. De obicei, la dictare se consideră că materialul care urmează a fi citit este preparat dinainte, iar limba folosită este cea literară. Condiţiile de înregistrare şi calitatea achiziţionării semnalului vocal sunt bune. Deşi în ziua de azi RLV la dictare este independentă de vorbitor, se foloseşte adaptarea sistemului la vorbitor, fie printr-o sesiune prestabilită, fie direct în timpul citirii. Astfel, există sisteme complete de RLV utilizate în scop de dictare şi aplicate în produse comerciale [208], [40], [179].

A doua categorie de aplicaţii a RLV este indexarea audio. Ea presupune transcrierea şi indexarea materialului audio înregistrat la conferinţe, la emisiuni TV/Radio, etc. Problemele principale care apar în acest caz sunt legate de natura neuniformă a limbajului vorbit folosit. Ea tinde mai mult spre vorbire spontană. Pentru limba engleză, au fost elaborate sisteme performante şi au fost construite tabele de referinţă pentru compararea rezultatelor [158], [159], [160], [69]. Folosind aceeaşi tehnologie astfel de sisteme au fost construite şi pentru alte limbi cum ar fi franceza, germana, italiana, japoneza, chineza (mandarin) şi spaniola [25], [111], [115], [154], [160].

A treia categorie de aplicaţii se bazează pe dialog om-maşină. Cele mai simple, care au intrat şi pe piaţă, sunt similare celor bazate pe răspunsuri prin DTMF (din engleză Dual Tone Multi-Frequency). Acestea necesită un nivel scăzut de înţelegere a limbajului natural, de exemplu navigarea printr-un meniu. Aceste aplicaţii folosesc, de regulă, un vocabular mic (100-500) de cuvinte. Există şi aplicaţii (servicii) bazate pe dialog care folosesc vocabulare mari (mii de cuvinte). Domeniul în care RLV este cea mai răspândită este acela de informaţii de călătorie. Alte aplicaţii sunt apelarea prin nume [191], [79], informaţii despre starea vremii [215], nume, adrese şi numere de telefon, accesarea de conturi financiare [50], etc. Aceste aplicaţii pot fi servicii bazate pe telefonie sau cu interfeţe multimedia. În primul caz, pot apărea probleme suplimentare legate de banda îngustă a semnalului vocal şi de eventualele perturbaţii în canalul de telecomunicaţie. În ambele cazuri, aplicaţiile nu pot presupune şi impune dinainte calitatea microfonului şi a sistemului de achiziţie a semnalului vocal. De aceea, SRLV trebuie să fie cât mai robust. Pentru interfeţele multimedia, s-au obţinut progrese mari datorită unor standarde stabilite de către marile companii. Unul din ele este VoiceXML [141], definit de către VoiceXML Forum (creat de AT&T, IBM, Lucent, Motorola).

RLV este aplicată şi cu alte scopuri care nu sunt menţionate mai sus. Câteva dintre ele sunt bazate pe căutarea/urmărirea unui cuvânt anume într-o secvenţă de semnal vocal (în

engleză, *word-spotting*) [6], [5]. Extragerea versurilor dintr-un cântec se bazează tot pe RLV [70]. Din ce în ce mai mult, RLV este folosit și la recunoașterea de vorbitor [161].

1.3. STADIUL ACTUAL DE DEZVOLTARE ÎN RLV ȘI TENDINȚE NOI DE CERCETARE

Recunoașterea limbajului vorbit (RLV) se confruntă, în ziua de azi, cu noi provocări care au stimulat dezvoltarea câtorva direcții principale de cercetare. Acestea sunt RLV robustă, RLV de vocabular mare, modele de limbă avansate care ajută în procesul de recunoaștere și implementarea SRLV *embedded*, adică pe dispozitive electronice portabile. O altă direcție de cercetare este RLV în condițiile în care semnalul vocal este trimis printr-un canal de telecomunicații. În acest caz, se folosesc toate rezultatele obținute în celelalte direcții, dar intervin alte constrângeri care necesită soluții. Sursa acestor provocări este, printre altele, nivelul ridicat de pretenții al utilizatorilor de dispozitive electronice. Într-adevăr, limbajul vorbit este cel natural și este cel mai ușor mod de a comunica, dar pe de altă parte, interfețele cu utilizatorul au devenit așa de prietenoase încât SRLV introdus trebuie să fie rapid, să aibă acurateţe mare, și să poată lucra în diferite condiții de zgomot.

În prezent, pentru modelarea acustică se folosesc modelele Markov ascunse (HMM, din engleză Hidden Markov Models). Deși au fost încercate și DTW (din engleză Dynamic Time Warping), rețele neuronale (RN) și modele *fuzzy*, care au dat rezultate modeste, cele mai bune performanțe au fost obținute prin utilizarea HMM-urilor. Există și soluții hibride care combină HMM cu alte modele, dar rezultatele au fost modeste [184], [46].

RLV robustă înseamnă recunoaștere în condiții variate și adverse. Variabilitatea în RLV provine din vocea caracteristică vorbitorului, starea psihică și emoțională a acestuia, dialectul, tipul propoziției, etc. Pentru a rezolva problema variabilității se folosesc tehnicilie de adaptare prin MLLR (din engleză Maximum-Likelihood Linear Regression) [76], [55], [136] și MAP (din engleză Maximum A-Posteriori Probability) [81], [186]. Condițiile adverse sunt create în principal de către zgomotul de fond și de calitatea înregistrării semnalului vocal. În ziua de azi, nu se poate imagina o aplicație în care vorbitorul să vorbească într-un studio de înregistrare. Cercetările în această direcție constau în îmbunătățirea estimării parametrilor vocali și a modelului acustic. Astfel, cele mai bune rezultate sunt obținute cu parametrii MFCC [50] și PLP [93] și cu variante îmbunătățite ale acestora [94]. Mai nou, analiza semnalului vocal se face pe o succesiune de cadre prin metoda numită SPLICE [58]. Studii importante s-au făcut asupra modelării duratei unității de limbă alese pentru modelare [175], [209]. Protecția față de zgomot este realizată fie prin eliminarea zgomotului din semnalul vocal fie prin compensarea modelelor acustice astfel încât sa se adapteze la noile condiții de zgomot. Eliminarea zgomotului din semnalul vocal a fost obiectul multor cercetări și multe soluții au fost găsite pentru diferite tipuri de zgomot [1], [148], [77], [71]. Un caz particular de perturbație este reverberația. Ea a fost studiată în [87] și în [119] unde s-au dat și câteva soluții pentru eliminarea acesteia. În timp ce aceste metode au ca scop îmbunătățirea calității percepției semnalului vocal, compensarea modelelor acustice are ca scop creșterea capacității de discriminare a modelelor și adaptarea acestora la noile condiții de zgomot și, de aceea, poate da rezultate mai bune [3], [118], [206]. Există două metode care care se bazează indirect atât pe eliminarea zgomotului din semnalul vocal cât și pe compensarea modelelor. Acestea

sunt metoda „parametrilor lipsă” și decodarea sub incertitudine [59], [126], [127]. O tehnică importantă care operează la nivel acustic, este cea a mașinilor cu vectori suport (SVM, din engleză Support Vector Machines) [78], [187], [156]. Prin ea se pot determina mai eficient granițele între modelele diferite, ceea ce face ca metoda să fie eficientă și la recunoașterea de vorbitor [37], [205].

RLV de vocabular mare (LVCSR, din engleză Large Vocabulary Continuous Speech Recognition) prezintă dificultăți suplimentare. Problemele principale sunt achiziția unei baze de date mare, perplexitatea ridicată a unei asemenea baze de date și reducerea vitezei de rulare, care pentru o rețea de cuvinte de dimensiuni mari tinde să aibă valori mici. În acest caz, dimensiunea vocabularului este de cel puțin câteva zeci de mii de cuvinte. În aceste condiții, și modelul acustic trebuie adaptat și optimizat pentru a modela variația mare a bazei de date [83]. O optimizare făcută în acest sens, ține cont de similitudinea dintre diferite modele ale aceluiași fonem și este implementată prin legarea distribuțiilor HMM-urilor. Această idee de bază este aplicată în diverse sisteme în care diferența între modelele folosite este de nuanță. Aceste modele sunt *senone* [105], *genone* [52], *PEL* [9], *stări-legate* [214], [192], [213]. O metodă care reduce perplexitatea și crește viteza de rulare este reducerea adaptivă a vocabularului [39], [82]. Vocabularul este redus funcție de subiectul discuției, dialectul utilizat, etc. Această metodă, însă, introduce erori de tipul ”cuvinte în afara dicționarului” (OOV, din engleză out-of-vocabulary) [157], [82]. La utillizarea vocabularului mare, apar și probleme de pronunție. Acestea sunt legate atât de devieri de pronunție general acceptate în limbajul literar vorbit cât și de confuzia produsă în cazul în care cuvinte diferite sunt pronunțate la fel sau vice versa [155]. Utilizarea *regulilor fonologice* are ca scop rezolvarea acestor probleme [44], [88], [122]. Alte metode de reducere a timpului de rulare se bazează pe reducerea spațiului de căutare folosind informația provenită de la un model de limbă [147], [182], [188], [173], [199], [207]. Toate modelele de limbă folosite în aceste lucrări sunt bazate pe modele statistice.

Modelele de limbă sunt bazate pe faptul că unele succesiuni de cuvinte sunt mai des întâlnite decât altele, iar unele nu pot să apară deloc. Cele mai răspândite modele de limbă sunt cele bazate pe *n*-grame care încearcă să estimeze probabilitatea de apariție a unui cuvânt cunoscându-se cele *n*-1 cuvinte care il preced. Problema principală la această abordare este ponderarea scorurilor pe baza probabilităților de apariție, mai ales pentru succesiunile rare. Câteva soluții s-au dat în [42], [114], [120]. Cele mai utilizate sunt bigramele și trigramele (unde *n* este egal cu 2 respectiv cu 3). Câteva îmbunătățiri au fost aduse prin utilizarea 4-gramelor [8], [210] și a 5-gramelor [132], [177]. Pentru aplicații specifice se pot utiliza gramatici cu stări finite [216], [4], [98]. Acestea dau rezultate bune pentru unele aplicații, însă pentru aplicații de vocabular mare, este greu de elaborat o asemenea gramatică.

În ultimii ani, cercetările în direcția RLV *embedded* au luat o amploare foarte mare, motivul principal fiind dezvoltarea telefoanelor mobile inteligente, a dispozitivelor portabile și a altor dispozitive auxiliare. Cu toate acestea resursele acestor dispozitive rămân inferioare resurselor PC-urilor, iar algoritmii care oferă o acuratețe mare de recunoaștere sunt costisitori din punct de vedere al puterii de procesare și al memoriei necesare. Astfel, există atât încercări software de RLV embedded [104], [18], cât și încercări hardware [129], [130], [181]. RLV *embedded* nu are o direcție principală de cercetare, ci orice optimizare care reduce complexitatea și resursele necesare, contribuie la implementarea unui RLV *embedded*.

Optimizările pot fi în prelucrarea şi extragerea parametrilor semnalului vocal, în modelul acustic, în modelul de limbă sau în decodor. Câteva dintre aceste optimizări constau în tehnici de cuantizare [190], [16], [54], legarea parametrilor HMM-ului [17], [11] şi dezvoltarea algoritmilor în virgulă fixă [113], [202]. În urma acestor cercetări, s-au construit sisteme pentru dispozitive cu resurse limitate cum ar fi PocketSphinx [104], PocketSUMMIT [95] şi un sistem dezvoltat de IBM [152]. O altă problemă care se pune la implementarea unui RLV *embedded*, este eficienţa în consumarea bateriei dispozitivului portabil. Câteva studii care au ca scop optimizarea algoritmilor folosiţi la RLV, în vederea utilizării eficiente a bateriei, sunt [125], [121], [150], [138].

Problemele care apar la RLV *embedded* sugerează implementarea unui sistem în care doar achiziţia, prelucrarea şi extragerea semnalului vocal se face pe dispozitivul portabil (client), iar recunoaşterea se face pe un calculator puternic (server). Această soluţie implică un canal de comunicaţie de date între client şi server. Există şi standarde care definesc această soluţie şi cele mai importante sunt cele de la ETSI (institutul european de standarde în telecomunicaţii, din engleză European Telecommunications Standards Institute) [61], [62], [63], [64]. Aceste standarde stabilesc care funcţii trebuie implementate pe dispozitivul portabil şi care de către server. Ele definesc setul de parametrii vocali care trebuie utilizaţi şi modul de transmisie. Cât timp există transmisie pe un canal de telecomunicaţii, există şi riscul ca datele să fie perturbate de diferite imperfecţiuni ale canalului, cum ar fi pierderea de pachete, întârzierea, etc. Există multe studii care analizează efectul acestor imperfecţiuni asupra performanţelor unui SRLV. Efectul pierderilor de pachete asupra acurateţei SRLV este analizat în [15], [142], [140], [195], [196], [144], [145]. Toate aceste cercetări ajung la concluzii similare, adică performanţele unui SRLV scad cu creşterea pierderii de pachete, dar există un prag sub care performanţele sistemului se menţin, mai ales dacă se foloseşte un model de limbă adecvat. Experimentele arată că erorile care apar în canalul de telecomunicaţie degradează rata de recunoaştere mai mult decât pierderea de pachete [13], [193], [139]. De aceea, multe studii s-au concentrate pe metode de corecţie/mascare a erorilor, cum ar fi interpolarea [144], [108], [162], [194], estimarea [23], [163], [164], [106], [91], şi corecţia făcută în decodor. Aceasta din urmă foloseşte aceleaşi principii folosite în algoritmii de „parametri lipsă” şi de decodare sub incertitudine. Aplicarea acestor metode în cazul erorilor de transmisie a fost studiată în [117], [51], [38], [145], [109], [171]. Alegerea codorului de voce are şi ea o importanţă deosebită pentru că influenţează direct asupra calităţii semnalului. Studiile arată, însă, că nu întotdeauna codecul care oferă calitate mai bună a semnalului vocal obţine şi o rată de recunoaştere mai mare [14], [67], [97]. Efectul eventualului zgomot care poate avea loc în canalul de telecomunicaţii este tratat la fel ca în cazul RLV robustă.

1.4. STADIUL ACTUAL ÎN RLV ÎN LIMBA ROMÂNĂ

În domeniul recunoaşterii limbajului vorbit, s-au făcut studii şi pentru limba română, dar aceste studii nu au ajuns la un stadiu avansat datorită resurselor limitate în această limbă. În primul rând, lipseşte o bază de date bogată şi variată. În timp ce pentru alte limbi (engleză, franceză, etc.) o bază de date completă este de câteva sute de ore de înregistrări, bazele de date în limba română nu sunt mai mari decât câteva ore. În al doilea rând, numărul de

cercetători în acest domeniu este mic. Performanța RLV este influențată de mulți factori atât din punct de vedere al modelului acustic cât și din punct de vedere al modelului de limbă. De aceea este nevoie de un număr mare de experimente.

În recunoașterea vorbirii continue, s-au realizat câteva teze de doctorat [19], [149]. În toate cazurile, au fost folosite HMM. În [19], s-a elaborat o metodă pentru achiziția unei bazei de date de vorbire continuă în limba română și s-a analizat performanța recunoașterii pentru diferite unități de vorbire și pentru un număr diferit de componente de mixturi gaussiene la modelarea ieșirii stării unui HMM. Baza de date are 60 vorbitori și aproximativ 200 de fraze. În [149], s-a făcut o analiză mai amănunțită, în care au fost variați parametrii modelului acustic. Mai mult decât atât, s-a încercat reducerea complexității decodorului, în principal, prin micșorarea fascicolului de căutare și creșterea robusteții recunoașterii prin metoda normalizării prin medie cepstrală și prin metoda antrenării multiple. S-a folosit și un model de limbă determinist relativ simplu. Un model de limbă mai elaborat a fost propus în [56] și este bazat pe *n*-grame (cu particularizare pe 3-grame) și pe metode de netezire a probabilităților acestora. În [56], au fost folosite și rețelele neuronale pentru clasificarea fonemelor în vederea modelării acustice.

Alte lucrări privind recunoașterea vorbirii în limba română pe care le putem menționa sunt [151], [26], [57], [86], [153]. În particular, în [176] și în [107] se folosește de DTW. Rețelele neuronale (RN) au fost folosite în [200], [32], [90]. S-au făcut și încercări de modele *fuzzy* [84] și de modele hibride HMM-RN [85]. În [43], a fost propusă o soluție pentru implementarea unui SRLV de vocabular mic pe un procesor de semnal. În [185], se folosesc RN pentru RLV folosind semnale de pe canalul telefonic.

Există și alte studii pentru limba română care ajută în mod indirect la recunoașterea vorbirii. În particular, analiza prozodiei combinată cu un model de limbă reduce numărul de ipoteze posibile. În această direcție, putem menționa [198], [34], [7]. Identificarea cuvintelor cheie poate de asemenea ajuta procesul de recunoaștere pentru că segmentează semnalul vocal [33].

1.5. POZIȚIONAREA LUCRĂRII RELATIV LA STADIUL ACTUAL DE DEZVOLTARE

Având în vedere tendințele de dezvoltare în domeniul RLV și stadiul de dezvoltare a RLV în limba română, în acest paragraf se prezintă poziționarea acestei lucrări relativ la acestea. În primul rând, teza de doctorat este un studiu în domeniul RLV în limba română și una dintre preocupările principare a fost mărirea bazei de date în limba română. Vocabularul utilizat este cel mai mare până în prezent cu peste 10000 de cuvinte. S-a construit o infrastructură pentru o achiziție simplă a unei bazei de date etichetate pentru scopuri multiple. Pentru bazele de date achiziționate s-a măsurat distribuția fonemelor limbii române și s-a făcut o analiză din punct de vedere fonetic. De asemenea s-a construit o infrastructură bazată pe utilitarele HTK (Hidden Markov ToolKit) [68], pentru rularea rapidă a software-urilor de antrenare, recunoaștere și de prelucrarea rezultatelor. Pentru antrenarea sistemului s-a propus o strategie care ajută la alinierea semnalului vocal cu transcrierea fonetica (etichetarea bazei de date este un proces lent și de aceea se încearcă o etichetare cât mai rară. Însă, cu cât

etichetele sunt mai rare cu atât alinierea este mai greu de realizat). S-a făcut o analiză amănunțită a performanței SRLV funcție de variația parametrilor modelului acustic (HMM) cum ar fi: numărul de stări, numărul de parametri vocali, numărul de componente ale mixturii gaussiene, etc. Prin aceste variații, se urmărește găsirea unei configurații optime atât din punct de vedere al complexității sistemului cât și din punct de vedere al creșterii ratei de recunoaștere. Alte studii constau în variația unităților de vorbire, în mărirea graduală a bazei de date pentru a vedea evoluția ratei de recunoaștere și în variația bazei de date de teste pentru a măsura intervalul de încredere a ratei de recunoaștere.

În al doilea rând, s-au implementat o serie de optimizări care aduc o îmbunătățire a ratei de recunoaștere și o reducere a numărului de calcule (în vederea implementării RLV pe un dispozitiv portabil). S-au încercat metodele de legarea stărilor, reducerea fascicolului de căutare, variația penalizării de inserție, diferite constrângeri gramaticale simple, etc. În particular, s-au implementat două metode: una bazată pe o decizie prin combinarea scorurilor obținute cu unități de limbă diferite și cea de a două bazată pe alegerea adaptivă a dicționarului fonetic funcție de fonemele cu cea mai mare rată de recunoaștere. În aceasta din urmă, s-a măsurat rata OOV și sa încercat reducerea acesteia.

O analiză detaliată s-a făcut și pentru RLV în rețele de telecomunicații. Experimentele au fost făcute prin simularea condițiilor adverse de pierdere de pachete și zgomot. Pierderile de pachete au fost modelate astfel încât să poată simula condiții reale. În alte experimente, s-a măsurat efectul utilizării codecului în RLV. Rata de recunoaștere a fost măsurată pentru diferite codecuri și s-a analizat performanța fiecăruia.

Cea mai importantă parte a acestei lucrări o constituie studiul incertitudinii modelului acustic și a confuziei între foneme. S-au făcut măsurări cantitative pentru a vedea care sunt fonemele (și modelele mai în general) care se confundau cel mai des și studii calitative pentru a vedea care parametrii vocali nu evaluează corect modelele la decodare. Folosind aceste informații, s-au propus câțiva algoritmi atât pentru eliminarea parametrilor inutili cât și pentru ponderarea corectă a acestor parametrii funcție de relevanța lor. Acești algoritmi au fost apoi testați și pentru baza de date afectată de pierdere de pachete și de zgomot. Incertitudinea modelului acustic, așa cum este definită în această lucrare, este o informație importantă care poate fi folosită pentru re-evaluarea modelelor și nu impune limitări în aplicarea ei împreună cu celelalte tehnici de decodare sub incertitudine.

1.6. ORGANIZAREA LUCRĂRII

Capitolul 1 prezintă scopul lucrării care este RLV în limba română și RLV în rețele de telecomunicații mobile. În acest capitol, este făcut un mic rezumat cu stadiul actual de dezvoltare al RLV și cu tendințele noi de dezvoltare atât pe plan internațional cât și pe plan național. Un subcapitol este dedicat poziționării acestei lucrări în stadiul actual de dezvoltare.

Capitolul 2 prezintă o descriere sumară a arhitecturii unui SRLV și a problemelor care sunt ridicate de către RLV bazată pe HMM. Sunt descriși, de asemena, algoritmii de extragere a parametrilor semnalului vocal, de antrenare și de decodare (cu varianta particulară de implementare ”Pasarea Jetonului”). Ținând cont de restricțiile particulare impuse de implementarea pe dispozitive mobile, sunt prezentate și niște metode de optimizare cum ar fi

cea a stărilor legate și utilizarea HMM-urilor semi-continue. **Capitolul 2** se termină cu o scurtă descriere a modelului lingvistic bazat pe *n*-grame.

Capitolul 3 este dedicat RLV în rețele de telecomunicații și implementării acesteia pe dispozitive mobile. El începe cu descrierea problemelor care apar în acest caz particular și conține un rezumat a metodelor existente pentru rezolvarea acestor probleme. Astfel, sunt descrișii algoritmi pentru robustețea RLV față de zgomot bazați pe metode de compensarea parametrilor, compensarea modelelor și metode bazate pe incertitudine. De asemenea, sunt descrise efectele pierderii de pachete și efectele codării asupra performanțelor unui SRLV și sunt prezentate câteva metode care reduc aceste efecte negative.

Capitolul 4 prezintă metodele utilizate pentru achiziționarea bazelor de date în limba română, strategia de antrenare și infrastructura SRLV construită cu scopul de a ușura rularea utilitarelor și de a oferi flexibilitate la variația configurației. Sunt descrise în detaliu criteriile pentru achiziționarea bazelor de date cu scopuri diferite, avantajele și dezavantajele pe care le prezintă fiecare. Sunt descrise problemele care se întâlnesc la transcrierea fonetică și la construirea dicționarului fonetic. În acest capitol, s-a încercat o caracterizare din punct de vedere fonetic a bazei de date. Este abordată problema alinierii la antrenare și sunt oferite câteva soluții care ajută la rezolvarea acestei probleme. **Capitolul 4** se încheie cu descrierea infrastructurii SRLV.

Capitolul 5 prezintă un număr mare de experimente făcute cu scopul de a găsi configurația optimă a SRLV în limba română și pentru validarea unor metode de optimizare atât pentru acuratețea recunoașterii cât și pentru timpul de rulare. Astfel, se începe cu variația parametrilor modelului acustic (număr de stări, de componente ale mixturii, de parametri, tipul de unități de limbă, etc.) și se continuă cu analiza acurateței de recunoaștere la utilizarea restricților de limbă, la mărirea bazei de date și variația bazei de date de test. În partea a doua a **Capitolului 5**, se propun câteva metode de reducere a timpului de calcul cum ar fi reducerea fascicolului de căutare și micșorarea adaptivă a dicționarului fonetic. Tot în acest capitol, este propusă și o metodă de creștere a acurateței de recunoaștere bazată pe combinarea celor două unități de limbă fonem-alofon.

În **Capitolul 6** este formulată teoria bazată pe incertitudine și sunt prezentate rezultatele experimentale pentru validarea teoriei și aplicarea acesteia în cazul condițiilor adverse întâlnite în rețelele de telecomunicații (zgomot și pierderi de pachete). Se definește o mărime nouă numită incertitudinea modelului acustic și pe baza acestei mărimi se face o re-evaluare a modelului acustic. **Capitolul 6** prezintă două metode construite pe baza acestei mărimi, una cu scopul de a scoate parametrii de voce irelevanți, reducând astfel numărul de calcule necesare, iar cea de-a doua cu scopul de a ameliora rata de recunoaștere. Metodele sunt aplicate atât la nivel de model cât și la nivel de stare. O evaluare a metodelor este făcută și cu baza de date TIMIT [80] cu scopul de a valida generalitatea lor.

Capitolul 7 este dedicat concluziilor, contribuțiilor personale și dezvoltării viitoare a acestei lucrări.

CAPITOLUL 2

RECUNOAŞTEREA LIMBAJULUI VORBIT BAZATĂ PE MODELE STATISTICE

2.1. ARHITECTURA UNUI SISTEM DE RECUNOAŞTERE A LIMBAJULUI VORBIT

Pentru a obține performanțe ridicate, un sistem de recunoaştere trebuie să se folosească atât de informaţia acustică obţinută din datele de antrenare cât şi de informaţii de limbă. De obicei, procesele sunt organizate în module astfel încât să se poată dezvolta separat şi să se ofere extensibilitate (Fig. 2.1.). Modulele se pot executa secvenţial, dar pot exista şi bucle. Deşi recunoaşterea modelelor acustice este cea mai dificilă problemă de rezolvat, celelalte module sunt şi ele importante la îmbunătăţirea recunoaşterii.

Modulul de prelucrare de semnal are ca funcţie principală extragerea parametrilor acustici relevanţi pentru procesul de recunoaştere. Ideal aceşti parametri trebuie să conţină doar informaţia de vorbire (nu şi cea de vorbitor, emoţie, etc). Acest modul poate, în unele cazuri, să realizeze şi alte funcţii cum ar fi: detecţie de activitate vocala (VAD), reducere de zgomot, etc.

Modulul acustic este nucleul sistemului şi foarte des se foloseşte termenul de *recunoaştere* referindu-se doar la acest modul. El se foloseşte de modele acustice antrenate în prealabil şi de parametrii acustici ai semnalului vocal pentru a decide care este unitatea de limbă pronunţată de către vorbitor. Unitatea de limbă care poate fi fonemul, difonemul, trifonemul, alofonul, silaba, etc. este caracterizată din punct de vedere statistic de către modelul acustic (în cazul acestei lucrări sub forma de HMM). Rezultatul acestui proces este o succesiune sau un număr de succesiuni de unităţi de limbă cu probabilitatea cea mai mare să fie generate de către modelele de acustice respective.

Modulul lexical introduce restricţii în procesul de decizie ştiindu-se dinainte că nu toate succesiunile de foneme sunt posibile. El se foloseşte de modelele fonetice care în unele cazuri pot fi o listă de cuvinte posibile pentru recunoaştere. Acest modul poate fi separat sau

integrat cu modulul acustic cum este cazul algoritmului “Pasarea Jetonului” (Token Passing). Rezultatul acestui proces este o succesiune de cuvinte.

Modulul sintactic introduce restricţii suplimentare plecând de la premiza că nu orice succesiune de cuvinte este posibilă. Aceste restricţii sunt reguli gramaticale sau o informaţie statistică despre succesiuni de cuvinte, care reduc semnificativ numărul de combinaţii posibile ale cuvintelor la recunoaştere.

Această lucrare vizează în mod special modelul acustic. Acesta este analizat pentru a evidenția factorii care influențează performanța sistemului atât din punct de vedere al acurateței cât și din punct de vedere al complexității.

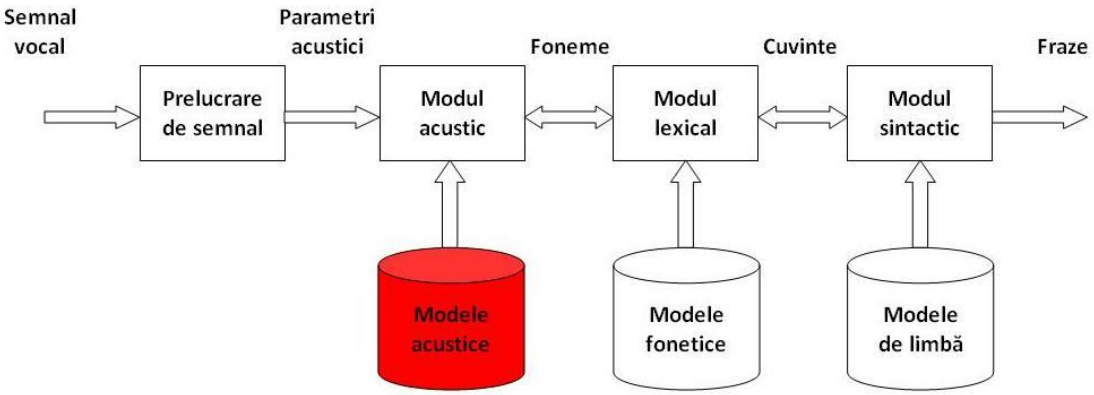

Fig. 2.1. Arhitectura unui sistem de recunoaștere a limbajului vorbit

2.2. MODELUL ACUSTIC CONSTRUIT PE HMM

2.2.1. Modelarea acustică

Modelarea acustică este fundamentală în recunoaşterea de voce. Pentru o secvenţă dată de observaţii acustice $O=o_1\ o_2 \ldots\ o_n$, scopul recunoaşterii este acela de a găsi secvenţa de cuvinte corespunzătoare $W=w_1\ w_2 \ldots\ w_m$ care are probabilitatea maximă a posteriori P(W|O):

$$\hat{W} = \arg\max P(W \mid O) = \arg\max \frac{P(W)P(O \mid W)}{P(O)} \tag{2.1}$$

Cum secvenţa X este una dată, problema recunoaşterii se reduce la:

$$W = \arg\max P(W)P(O \mid W) \tag{2.2}$$

Problema se pune cum putem construi modele acustice de acurateţe înaltă, P(O|W) şi modele lingvistice P(W), care să reflecte cât mai bine semnalul vocal care trebuie recunoscut. Pentru recunoastere de vocabolare mari, datorită numarului mare de cuvinte, este nevoie să se clasifice cuvintele într-o secvenţă de *subcuvinte,* în foneme sau alofone (în această lucrare alofon se numeşte fonemul pentru care se specifică fonemul care îl precede şi cel care il urmează). Aşadar, P(O|W) este strâns legat de modelul fonetic. P(O|W) trebuie să ţină cont de

variaţiile vorbitorilor, variaţiile de pronunţie, variaţiile mediului şi variaţiile coarticulării fonemelor dependente de context (după tipul propoziţiei, accentul într-un cuvânt, dialectul, etc.). De aceea este important în a adapta atât P(W) cât şi P(O|W) pentru a maximiza P(O|W). Procesul de recunoaştere a vorbirii continue este o problema mai dificilă decât o recunoaştere simplă de formă, pentru că în recunoaşterea de vorbire continuă avem un număr mare de cuvinte de căutat.

La baza modelării acustice stă HMM care oferă un mod eficient pentru a integra segmentarea, maparea în timp, potrivirea formelor şi contextul într-un mod unificat [103].

2.2.2. Definiția HMM

Modelele Markov Ascunse (HMM) sunt folosite în mod eficient în recunoaşterea de forme. Privind recunoaşterea vorbirii ca un caz particular, această metodă statistică poate fi folosită cu succes datorită avantajelor pe care le oferă. Faţă de modelele Markov simple, în cazul HMM-urilor nu există o bijecţie între stare şi ieşire. Acesta oferă flexibilitate mai mare şi se potriveşte foarte bine semnalului vocal, în care acelaşi fonem poate avea durate diferite după caz. Un exemplu de HMM este prezentat în Fig. 2.2.

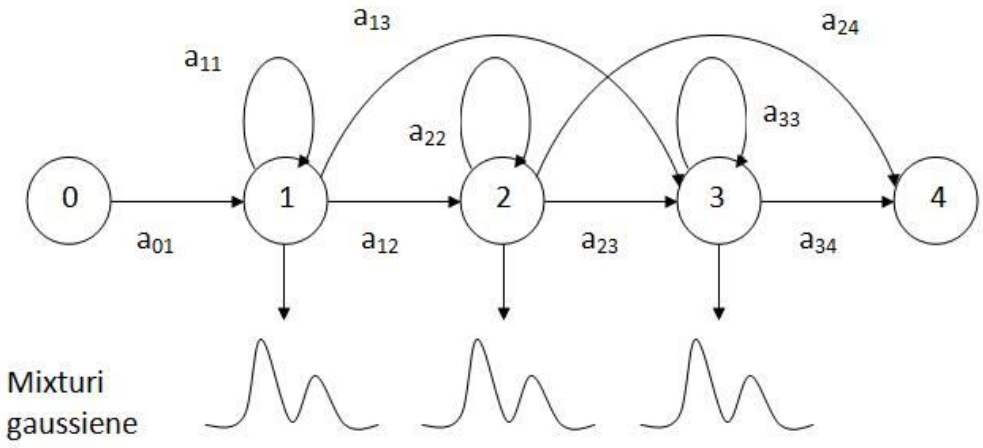

Fig. 2.2. Un exemplu de HMM cu 5 stări din care prima şi ultima nu sunt emisive

Prima şi ultima stare nu sunt emisive. În rest, fiecare stare emite o valoare după o anumită funcție de distribuție de probabilitate (stabilită în procesul de antrenare, vezi capitolul 2.2.7.). În această lucrare nu sunt permise decât tranzițiile din starea i în stările i, i+1 și i+2. Probabilitățile de tranziție constituie matricea de tranziție A. Tranzițiilor nepermise le corespunde o probabilitate de tranziție egală cu 0.

Un HMM este complet definit prin:

- $O=\{o_1,o_2,...,o_M\}$ – Un alfabet pentru observaţiile de la ieşire, pentru cazul în care această mulțime este discretă.
- $\Omega=\{1,2,...,N\}$ – Spaţiul de stări.

- $A=\{a_{ij}\}$ – Matricea cu probabilităţiile de tranziţie, unde a_{ij} este probabilitatea de a trece din starea *i* în starea *j*.
- $B=\{b_i(k)\}$ – Matricea de ieşire, unde $b_i(k)$ este probabilitatea să se emită simbolul o_k la starea *i*. Pentru cazul în care ieşirea unui HMM ia valori continue, atunci $bi(k)$ este o funcţie de densitate de probabilitate (în particular, la această lucrare ieşirea unei stări este modelată printr-o mixtură gaussiană).
- $\Pi = \{\pi_i\}$ – Probabilitatea ca starea iniţială să fie starea *i*.

Pentru ca un HMM să fie complet, este nevoie să se specifice numărul de stări *N*, dimensiunea *M* a alfabetului de observaţii *O* (dacă este cazul), şi matricile de probabilităţi *A, B, Π* [103]. Aşadar un model îl putem nota cu:

$$\Phi=(A,B,\Pi) \tag{2.3}$$

Ca să putem aplica acest model în lumea reală trebuie rezolvate următoarele probleme:

A. Problema evaluării – Dat fiind un model Φ şi o secvenţă de observaţii $O=(O_1,O_2,...,O_T)$, care este probabilitatea $P(O|\Phi)$ ca această secvenţă *O* să fi fost generată de modelul Φ?

B. Problema decodării – Dat fiind un model Φ şi o secvenţă de observaţii $O=(O_1,O_2,...,O_T)$, care este cea mai probabilă secvenţă de stări $S=(s_0,s_1,...,s_T)$ ale modelului care a generat aceste observaţii?

C. Problema antrenării – Dat fiind un model Φ şi un set de observaţii, cum putem ajusta parametrii modelului Φ astfel încât să maximizăm probabilitatea $\prod_o P(O|\Phi)$?

În esenţă, recunoaşterea formelor sau a vorbirii constă în a decide care model (sau succesiune de modele) se potriveşte cel mai bine cu un set de observaţii date pe baza probabilităţii $P(O|\Phi)$. Ceea ce înseamnă că problema recunoaşterii în sine este rezolvată dacă avem modele antrenate. Decodarea este importantă la antrenarea HMM-urilor în cazul recunoaşterii vorbirii continue, după cum este prezentat în capitolul 2.2.8. Cea mai dificilă este problema antrenării sau a estimării. Dintr-un set de date de antrenare trebuie estimaţi parametrii HMM-ului astfel încât ei să caracterizeze cât mai bine unitatea de voce aleasă.

2.2.3. HMM şi recunoaşterea limbajului vorbit

Semnalul vocal se imparte în entităţi elementare, de exemplu: cuvinte, foneme, difoneme, trifoneme, alofone. Fiecărei entităţi i se asociază un HMM, iar stării unui HMM i se poate asocia o fereastră de timp aleasă cu parametrii de voce calculaţi pentru această fereastră. Dimensiunea ferestrei se alege astfel încât parametrii semnalului vocal să fie constanţi pe acea durată (valori uzuale sunt 10, 20, 30ms). Se poate imagina că în timpul vorbirii, tractul vocal trece printr-o serie de stări (care sunt modelate cu stările HMM) şi în fiecare stare se emite un segment din semnalul vocal cu un vector de parametrii care vor constitui ieşirea stării HMM-ului. Parametrii de voce care se folosesc la recunoaştere sunt MFCC, PLP, LPC, etc.

Cum parametrii de voce iau valori intr-un spaţiu continuu trebuie ca şi vectorul de ieşire al HMM-ului sa aibă valori continue. Din acest motiv se folosesc mixturile gaussiene

pentru modelarea ieşirii stărilor HMM-ului. Fiecare parametru al vectorului de ieşire poate fi modelat printr-o sumă ponderată de funcţii de distribuţie normală:

$$b_j(O_t) = \sum_{g=1}^{G} w_{jg} N(O_t, \mu_{jg}, \Sigma_{jg}) \tag{2.4}$$

unde:

$$N(O, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(O-\mu)'\Sigma^{-1}(O-\mu)} \tag{2.5}$$

unde $O=[o_1,o_2,..,o_n]$' este vectorul de observaţie n-dimensional, n este numărul de parametrii de voce pe care ii extragem din observaţii, $|\sum|$ este determinantul matricii diagonale de covarianţă, G este numărul de componente ale mixturi şi w_{jg} este ponderea componentei g ale stării j.

2.2.4. Problema evaluării unui HMM

Problema evaluării unui HMM constă în calcularea probabilităţii $P(O|\Phi)$, adică probabilitatea ca observaţia O să fie generată de modelul Φ. Această problemă este rezolvată prin algoritmul progresiv. Probabilitatea progresivă este definită ca fiind:

$$\alpha_j(t) = P(o_1,...,o_t, x(t) = j \mid M) \tag{2.6}$$

Adică, α este probabilitatea ca modelul M să fi generat primele t observaţii, dacă la momentul t se află în starea j. Această probabilitate se calculează recursiv:

$$\alpha_j(t) = \left[\sum_{i=2}^{N-1} \alpha_i(t-1) a_{ij} \right] b_j(o_t) \tag{2.7}$$

Trebuie observat că se sumează toate probabilităţile obţinute din toate stările care în momentul t tranzitează în starea j. Cum prima şi ultima stare a HMM sunt non-emisive, rezultă următoarele condiţii iniţiale şi finale:

$$\alpha_1(1) = 1 \tag{2.8}$$

$$\alpha_j(1) = a_{1j} b_j(o_1) \tag{2.9}$$

$$\alpha_N(T) = \sum_{i=2}^{N-1} \alpha_i(T) a_{iN} \tag{2.10}$$

Din definiţia probabilitaţii progresive rezultă că:

$$P(O \mid M) = \alpha_N(T) \tag{2.11}$$

Trebuie observat că algoritmul progresiv calculează probabilitatea ca modelul M să fi generat secvenţa de observaţii O, luând în considerare toate drumurile care pot fi parcurse prin stările modelului. Există o discrepanţă între modul în care se calculează această probabilitate la antrenare şi la recunoaştere. La recunoaştere, se ia în considerare doar drumul (succesiunea

de stări) care are cea mai mare probabilitate. Această simplificare se face pentru reducerea complexității algoritmului de decodare.

Mai trebuie observat că probabilitatea $b_j(o_t)$, fiind valoarea unei funcții de densitate de probabilitate într-un punct, este un scor (*likelihood* în limba engleză) obținut de fiecare stare pentru un vector de observație dat. De aceea, în această lucrare se va folosi, în continuare, noțiunea de scor.

2.2.5. Problema decodării unui HMM

Problema decodării coincide cu problema recunoașterii. Deși la recunoaștere ne interesează doar modelul cel mai probabil, există cazuri, de exemplu în cazul vorbiri continue, unde explicitarea (decodarea) secvenței de stări devine necesară. Prin vorbire continuă, aici se înțelege cazul în care dintr-o secvență de observații trebuie gasită cea mai probabilă secvență de modele și nu se știu a priori granițele între cuvinte. Așadar, în acest caz pentru evaluarea HMM-ului se poate folosi o formulă similară ca cea folosită în algoritmul progresiv înlocuind funcția de însumare cu funcția de maxim. Astfel, se definește probabilitatea maximă de a genera secvența de observații de la momentul 1 la momentul *t*:

$$\phi_j(t) = \max_i\{\phi_i(t-1)a_{ij}\}b_j(o_t) \tag{2.12}$$

unde:

$$\phi_1(1) = 1 \tag{2.13}$$

$$\phi_j(1) = a_{1j}b_j(o_1) \tag{2.14}$$

Pentru a nu lucra cu numere mici, se folosește probabilitatea logaritmică și atunci formula (2.7) devine:

$$\psi_j(t) = \max_i\{\psi_i(t-1) + \log(a_{ij})\} + \log(b_j(o_t)) \tag{2.15}$$

Această formulă stă la baza algoritmului Viterbi de decodare [124]. În fond, algoritmul Viterbi găsește drumul optim într-o rețea de noduri, în care cele două dimensiuni sunt ferestrele de timp și stările modelelor Markov (Fig. 2.3.). Termenul $\log(b_j(o_t))$ din formula (2.15) reprezintă scorul nodului iar termenul $\log(a_{ij})$ reprezintă scorul tranziției. Algoritmul Viterbi poate fi sumarizat cum este arătat mai jos:

Pasul 1: Inițializarea

$$\psi_i(1) = \pi_i b_i(o_1) \qquad 1 \le i \le N$$

$$B_i(1) = 0$$

Pasul 2: Inducția

$$\psi_j(t) = \max[\psi_i(t\text{-}1)a_{ij}]b_j(o_t) \qquad 2 \le t \le T;\ 1 \le j \le N$$

$$B_j(t) = arg\ \max[\psi_i(t\text{-}1)a_{ij}] \qquad 2 \le t \le T;\ 1 \le j \le N$$

Pasul 3: Terminarea

cel mai bun scor = max[$\psi_i(t)$]

$s^*_T = arg\ \max[B_i(T)]$

Pasul 4: Backtracking

$s^*_t = B_{s^*t+1}(t+1)$ $\quad t = T-1, T-2,...,1$

$S^* = (s^*_1, s^*_2,..., s^*_T)$ este cea mai bună secvență de stări

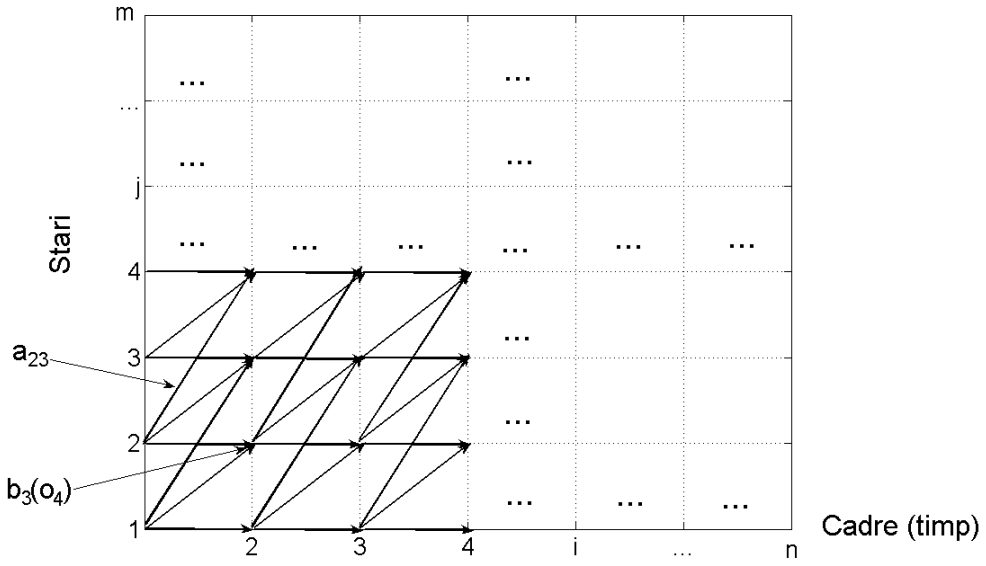

Fig. 2.3. Algoritmul de decodare Viterbi

2.2.6. Algoritmul de decodare Pasarea Jetonului (Token Passing, TP)

Algoritmul TP nu aduce nimic nou din punct de vedere al calculului costurilor sau al calculului drumului optim. În schimb, el rezolvă în mod direct problema impunerii restricțiilor de limbă prin introducerea conceptului de jeton [212]. Jetoanele se propagă prin rețea, de la stânga la dreapta (vezi Fig. 2.3.) și conțin informație despre parcurs și despre costul drumului până la nodul unde sunt evaluate. Avantajele acestei abordări constă în faptul că la recunoașterea de cuvinte sau de fraze, jetoanele din ieșirea unui HMM se propagă la intrările celorlalte modele. Astfel, decodorul poate lucra în continuu, sincron cu aparația unui nou vector de observație. Acest lucru este important pentru recunoaștere de vorbire în timp real. Ținând cont de observațiile de mai sus, algoritmul Viterbi prezentat în capitolul 2.2.5. poate fi generalizat astfel:

Inițializare:

Fiecare stare inițială are un jeton cu valoarea 0;

Restul de stări au un jeton de valoare ∞;

Algoritmul:

Pentru t=1 până la T execută

Pentru fiecare stare i execută

Mută o copie a jetonului din starea *i* către toate stările *j* incrementând scorul lui cu costul reprezentat de acea stare (nod) *j* și costul tranziției a_{ij};

Șterge toate jetoanele precedente;

Pentru fiecare stare *i* execută

Găsește jetonul din starea *i* cu cel mai mic cost (sau cel mai mare scor) și aruncă celelalte jetoane;

Finalizare:

Examinează toate stările finale, jetoanele cu cel mai mic cost (cel mai mare scor) sunt cele care indică drumul optim;

Trebuie observat că algoritmul permite mai multe stări inițiale și finale, ceea ce nu are sens pentru recunoașterea unui singur model, dar capătă importanță la recunoașterea de cuvinte și de fraze.

Algoritmul TP este folosit atât pentru HMM cât și pentru DTW. Ca și în cazul HMM-urilor, DTW impune costuri de tranziție pe orizontală și pe verticală. De fapt, DTW poate fi privit ca un caz particular al recunoașterii cu HMM. Aceste asemănări au fost evidențiate în [24], [112].

Dat fiind modul de propagare al jetoanelor, trecerea la recunoașterea de cuvinte (care conțin o succesiune de modele) sau de fraze devine ușoară. Costurile de tranziție din starea finală a unui model în starea inițială a celorlalte modele sunt calculate funcție de frecvența cu care un model este urmat de altul. Acestea sunt calculate pe baza unei gramatici construită, în mod uzual, cu ajutorul N-gramelor (vezi capitolul 2.4.). Arcele care leagă modele diferite se numesc *arce externe*. Modelele sunt legate între ele într-o buclă cum este arătat în Fig. 2.4.

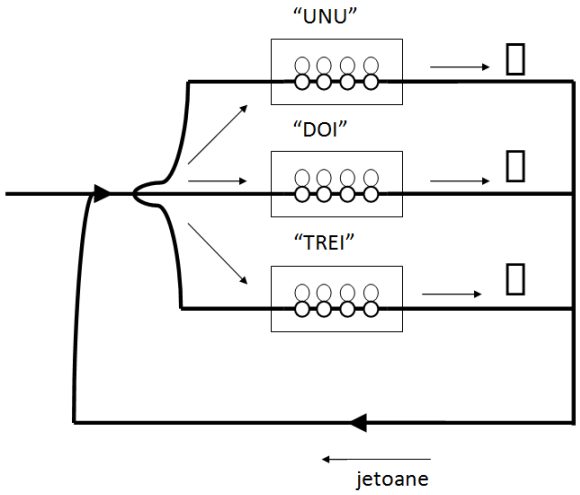

Fig. 2.4. Propagarea jetoanelor într-o rețea de cuvinte

Pentru recunoaștere de cuvinte sau de fraze, este necesar să se știe și succesiunea de modele cu costul cel mai mic. Pentru a realiza acest lucru se poate extinde algoritmul prin introducerea câmpului de *identificator de drum* în structura jetonului. Din punct de vedere al implementării, acest identificator poate fi un pointer către o înregistrare a tranziției între două modele care poartă numele WLR (din engleză, Word Link Record). Algoritmul poate fi completat cu următoarele acțiuni care se desfășoară la fiecare fereastră de timp *t*:

Pentru fiecare jeton care se propagă printr-un arc extern la fereastra de timp *t*:

Crează un nou WLR care conține:

<conținutul jetonului, *t*, identitatea modelului din care a ieșit>;

Modifică identificatorul de drum al jetonului astfel încât să pointeze către această nouă înregistrare.

Această extindere a algoritmului este ilustrată în Fig. 2.5. Granițele între modele sunt stocate în liste înlănțuite. La terminarea algoritmului, la momentul *T*, identificatorul de drum care este conținut în jeton, oferă posibilitatea de a descoperi tot drumul care are cel mai mic cost. Această concluzie este importantă pentru implementarea algoritmului în sisteme cu resurse limitate. Jetoanele de dimensiuni mari cresc complexitatea sistemului.

Astfel, modele pot fi privite ca niște unități abstracte care primes jetoane ca input și produc jetoane ca output. Această descriere ajută la implementarea algoritmilor de gramatici în care operația de a calcula costul pentru fiecare model pentru un vector dat de observații este privită ca o procedură *model_cost(t)*, ceea ce înseamnă un ciclu din bucla principală a algoritmului TP. Fiecare jeton reprezintă o aliniere parțială cu cost minim a unei secvențe de modele cu secvența de voce care a fost analizată până în acel moment. Controlul propagării jetoanelor între modele coincide cu introducerea restricțiilor gramaticale.

Fig. 2.6. arată cum poate fi definită o interfață standard între cele două componente, și anume calculul costurilor pentru modele și controlul propagării jetoanelor (adică restricții gramaticale). Nu toate jetoanele din ieșirea procedurii *model_cost(t)* coincid cu tranziții între modele, însă, cele corespunzătoare tranzițiilor sunt obiect al controlului componentei gramaticale. Modul de a privi algoritmul astfel este important pentru că permite implementarea, dezvoltarea și optimizarea celor două componente în mod independent.

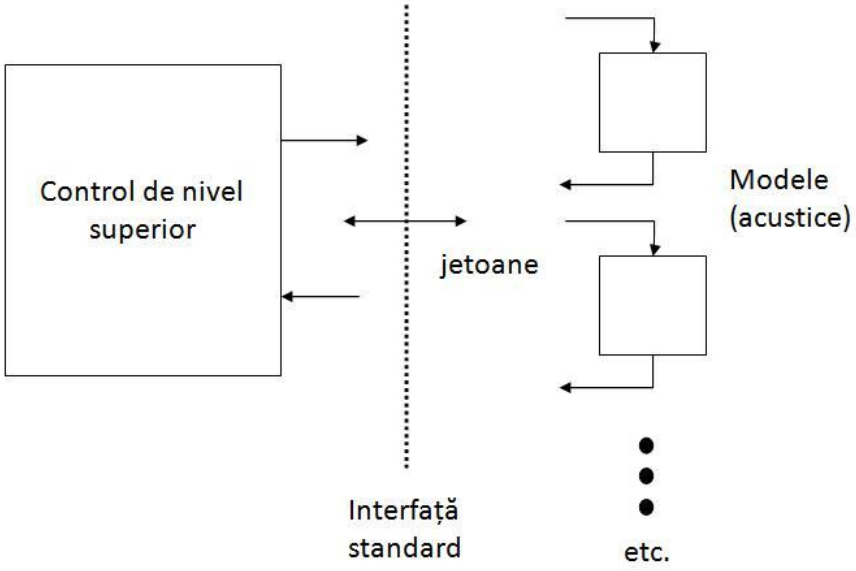

Fig. 2.5. Modificarea WLR după fiecare tranziție între cuvinte

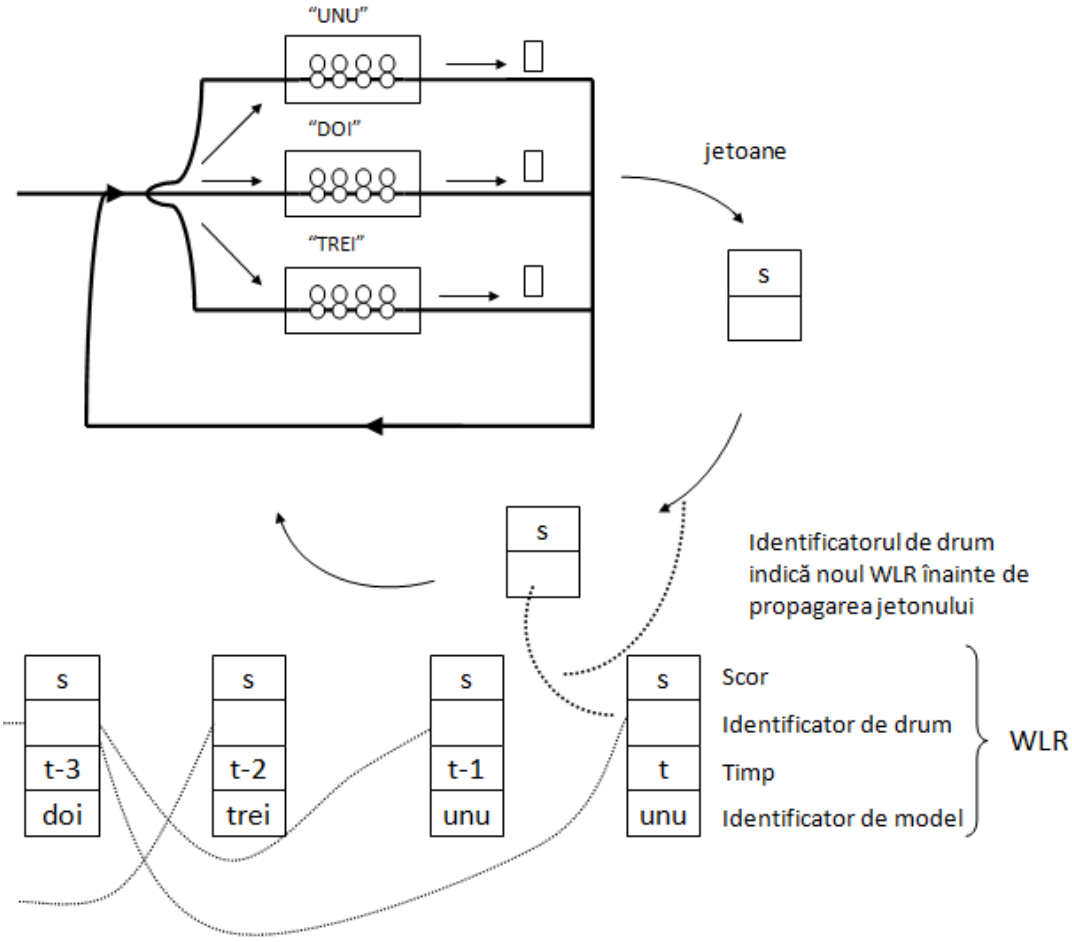

Fig. 2.6. Separarea modelelor acustice de gramatica de nivel superior

2.2.7. Problema estimării unui HMM

Se pune problema estimării parametrilor unui HMM (probabilităţile de tranziţie şi parametrii funcţiei de ieşire) de la un număr dat de secvenţe de observaţii care se presupun a fi generate de către acest HMM. Dificultatea problemei constă în faptul că nu se ştie a priori de care stare a fost generat fiecare vector de observaţie. Aşadar, raţionamentul începe cu ipoteza simplificătoare că s-ar şti care stare a generat fiecare vector de observaţie. În aceste condiţii mediile şi varianţele s-ar calcula cu formulele (2.16) şi (2.17):

$$\hat{\mu}_j = \frac{1}{T}\sum_{t=1}^{T} o_t \tag{2.16}$$

$$\hat{\Sigma}_j = \frac{1}{T}\sum_{t=1}^{T} (o_t - \mu_j)(o_t - \mu_j)' \tag{2.17}$$

Cum un vector de observaţii poate fi generat do oricare dintre stări cu o anumită probabilitate, mediile şi varianţele pot fi calculate ca sumele tuturor vectorilor de observaţie ponderate cu probabilitatea ca HMM să se afle în aceea stare în momentul corespunzător vectorului de observaţie. Deci, putem calcula media şi varianţa cu formulele (2.18) şi (2.19):

$$\hat{\mu}_j = \frac{\sum_{t=1}^{T} L_j(t) o_t}{\sum_{t=1}^{T} L_j(t)} \tag{2.18}$$

$$\hat{\Sigma}_j = \frac{\sum_{t=1}^{T} L_j(t)(o_t - \mu_j)(o_t - \mu_j)'}{\sum_{t=1}^{T} L_j(t)} \tag{2.19}$$

Formulele sunt ecuaţiile algoritmului Baum-Welch pentru calculul mediilor şi ale varianţelor. Ponderea $L_j(t)$ se poate calcula cu ajutorul algoritmului progresiv-regresiv. Probabilitatea progresivă α este cea definită şi calculată cu formula (2.7) în capitolul 2.2.4., iar probabilitatea regresivă este definită ca fiind probabilitatea ca HMM să fi generat secvenţa de observaţii de la t+1 la T dacă la momentul t HMM se află în starea j:

$$\beta_j(t) = P(o_{t+1},...,o_T \mid x(t) = j, M) \tag{2.20}$$

β se calculează prin recursia:

$$\beta_i(t) = \sum_{j=2}^{N-1} a_{ij} b_j(o_{t+1}) \beta(t+1) \tag{2.21}$$

cu condiţia iniţială:

$$\beta_i(T) = a_{iN} \tag{2.22}$$

şi cu condiţia finală:

$$\beta_1(1) = \sum_{j=2}^{N-1} a_{1j} b_j(o_1) \beta_j(1) \tag{2.23}$$

Produsul celor două probabilități, celei progresive și celei regresive, dă probabilitatea ca starea *j* a modelului *M* să fi generat observația din momentul *t*.

$$\alpha_j(t)\beta_j(t) = P(O, x(t) = j \mid M) \tag{2.24}$$

Așadar putem calcula ponderea $L_j(t)$ cu formula:

$$L_j(t) = P(x(t) = j \mid O, M) = \frac{P(O, x(t) = j \mid M)}{P(O \mid M)} = \frac{1}{P}\alpha_j(t)\beta_j(t) \tag{2.25}$$

unde $P=P(O|M)$.

Un raționament similar se poate face și pentru estimarea probabilităților de tranziție și pentru calculul ponderilor mixturilor, rezultând formulele finale pentru estimarea parametrilor:

$$\hat{\mu}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T} L^r_{jsm}(t) o^r_{st}}{\sum_{r=1}^{R}\sum_{t=1}^{T} L^r_{jsm}(t)} \tag{2.26}$$

$$\hat{\Sigma}_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T} L^r_{jsm}(t)(o^r_{st} - \hat{\mu}_{jsm})(o^r_{st} - \hat{\mu}_{jsm})'}{\sum_{r=1}^{R}\sum_{t=1}^{T} L^r_{jsm}(t)} \tag{2.27}$$

$$c_{jsm} = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T} L^r_{jsm}(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T} L^r_{j}(t)} \tag{2.28}$$

$$\hat{a}^{(q)}_{ij} = \frac{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T} \alpha^{(q)r}_i(t) a^{(q)}_{ij} b^{(q)}_j(o^r_{t+1}) \beta^{(q)r}_j(t+1)}{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T} \alpha^{(q)r}_i(t) \beta^{(q)r}_j(t)} \tag{2.29}$$

L^r_{jsm} este probabilitatea de a ocupa a *m*-a componentă gaussiană, la momentul *t*, la a *r*-a observație. Indexul *s* reprezintă sursa observațiilor, adică tipul de parametri din vectorul de observație.

Algoritmul Baum-Welch poate fi rezumat în următorii pași:

1. Se calculează probabilitățile progresive și regresive pentru fiecare stare *j* la fiecare moment *t* și ponderile *L* cu formula (2.25).

2. Se calculează parametrii HMM-ului, mediile, varianțele, matricile de tranziție și ponderile mixturilor cu formulele (2.26-2.29).
3. Se calculează probabilitatea P(*O*/*M*) și se compară cu cea obținută din iterația anterioară. Dacă valoarea obținută la iterația curentă este mai mică atunci algoritmul se oprește. În caz contrar, se reiau pașii 1 și 2 în care valorile probabilităților se calculează cu valorile estimate ale HMM-urilor din ultima iterație.

2.2.8. Antrenarea imbricată (*embedded*)

Antrenarea imbricată rezolvă problema calculării parametrilor modelelor în cazul în care lipsește informația despre granițele în timp între cuvinte. De cele mai multe ori, granițele sunt greu de stabilit, dar și atunci când se pot stabili, acest proces necesită mult timp și face ca achiziția unei baze de date de dimensiune mare sa devină costisitoare.

Problema principală este determinarea observațiilor care aparțin unui anumit model. Ca și în cazul algoritmului Baum-Welch prezentat în capitolul 2.2.7., contribuția fiecărui vector de observații la parametrii unei stări se face ponderat cu probabilitatea ca modelul să se afle în acea stare la momentul de timp dat. În acest scop, se construiește, pentru fiecare înregistrare, un HMM format din toate modelele din etichetele (transcrierile) respective pastrând ordinea (Fig. 2.7.). Apoi se calculează mediile și varianțele pentru mixturi ponderându-se cu probabilitățile care rezultă din algoritmul progresiv-regresiv. Astfel, deși nu se cunosc granițele între cuvinte, parametrii de stare se pot calcula prin ponderarea potrivită a vectorilor de observații. Formulele pentru probabilitățile de tranziție rămân nemodificate, dar formulele de calcul pentru probabilitățile de tranziție în/din stările non-emisive sunt ușor modificate după cum arată formulele (2.30) și (2.31):

$$\hat{a}_{1j}^{(q)} = \frac{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T}\alpha_1^{(q)r}(t)a_{1j}^{(q)}b_j^{(q)}(o_t^r)\beta_j^{(q)r}(t)}{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T}\alpha_1^{(q)r}(t)\beta_1^{(q)r}(t)+\alpha_1^{(q)r}(t)a_{1Nq}^{(q)}\beta_1^{(q+1)r}(t)} \tag{2.30}$$

$$\hat{a}_{iNq}^{(q)} = \frac{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T}\alpha_i^{(q)r}(t)a_{iNq}^{(q)}\beta_{Nq}^{(q)r}(t)}{\sum_{r=1}^{R}\frac{1}{P_r}\sum_{t=1}^{T}\alpha_i^{(q)r}(t)\beta_j^{(q)r}(t)} \tag{2.31}$$

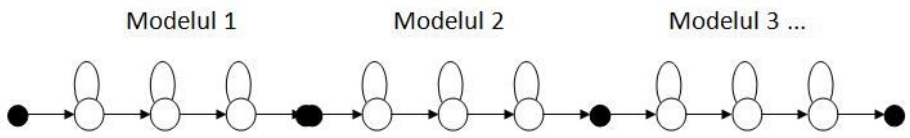

Fig. 2.7. Unirea HMM-urilor la antrenarea *embedded*. Stările non-emisive a modelelor succesive sunt suprapuse pentru a permite tranziții între modele.

2.2.9. Tehnica stărilor legate

Există cazuri în care modele diferite pot avea părți similare funcție de unitatea de limbă aleasă pentru modelare. De exemplu, când această unitate de limbă este trifonemul, toate trifonemele cu același fonem central vor avea stările centrale similare. Stările de pe margine, care reprezintă tranziția către un alt fonem vor fi diferite. În aceste condiții este avantajos ca stările centrale să aibă aceiași parametri, adică ieșirile lor să fie identice din punct de vedere statistic. Această tehnică se numește *tehnica stărilor legate*. Studiile arată că și matricile de tranziție între trifonemele cu același fonem central sunt similare și atunci pot fi legate. În timpul antrenări, având în vedere principiile antrenării imbricate, stările legate din mai multe trifoneme vor fi privite ca o singură stare, dar care este întâlnită mai des. Astfel, scade numărul global de stări, micșorând rețeaua de noduri la decodare.

2.2.10. HMM semi-continuu

Funcția de densitate de probabilitate din ieșirea unei stări a HMM-ului prezentată în capitolul 2.2.3. este continuă, dar în multe aplicații mai ales la implementări hardware pe platforme particulare este avantajoasă utilizarea unei funcții de densitate de probabilitate discretă. Aceasta din urmă oferă o complexitate mai redusă de implementare dar introduce în același timp și o eroare de cunatizare. Un compromis între cele două soluții îl oferă HMM semi-continuu. El pleacă de la premiza că toate mixturile sunt legate, formând un set de nuclee partajate și fiecare ieșire a unei stări a HMM-ului poate fi privită ca o sumă ponderată a acestor mixturi. La HMM discrete se folosește un codebook pentru mapare unui vector de observații continuu x la o_k și, astfel, se poate folosi distribuția discretă de ieșire $b_j(k)$. Codebook-ul poate fi privit ca fiind unul dintre aceste nuclee partajate. Astfel, formula (2.4) devine:

$$b_j(x) = \sum_{k=1}^{M} b_j(k) f(x \mid o_k) = \sum_{k=1}^{M} b_j(k) N(x, \mu_k, \Sigma_k) \tag{2.32}$$

unde o_k este al k-lea codeword, $b_j(k)$ este ponderea pentru funcția de densitate de probabilitate continuă, iar $N(x,\mu_k,\Sigma_k)$ sunt independente de HMM-uri individuale și sunt partajate de către toate modelele fiind multe la număr (M).

Din punct de vedere al HMM-ului discret, codebook-ul necesar constă în M funcții continue de densitate de probabilitate și fiecare funcție este caracterizată de un vector de medii și o matrice de covarianță. O cuantizare tipică produce un index de codeword care implică o distorsiune minimă față de un vector continuu de observații x. La HMM semi-continuu, operația de cuantizare produce valori pentru funcțiile continue de densitate de probabilitate $f(x|o_k)$ pentru toate codewords o_k. Deși structura modelului semi-continuu poate fi similară cu cea a modelului discret, probabilitățile de ieșire nu sunt folosite direct ca în cazul discret, adică drept valoarea cuantizată. În schimb, funcțiile de distribuție ale codebook-ului VQ, $N(x,\mu_k,\Sigma_k)$, sunt combinate cu probabilitățile de ieșire discrete ca în formula (2.32). Aceasta este o combinație de coeficienți de ponderare discreți cu funcții continue de densitate de probabilitate. O asemenea reprezentare permite reestimarea codebook-ului original împreună cu HMM.

Pe de altă parte, modelul semi-continuu seamănă cu HMM continuu cu *M* mixturi prin funcțiile continue de densitate de probabilitate care sunt partajate între toate modelele. În comparație cu HMM cu mixturi continue, HMM semi-continuu poate menține abilitatea de a modela funcții de densitate de probabilitate cu multe mixturi. În plus, numărul de parametrii care trebuie estimați și complexitatea de calcul sunt reduse, pentru că toate funcțiile de densitate sunt partajate. În concluzie, se poate obține un compromis bun între detalierea modelului acustic și posibilitatea antrenării lui.

În practică, având în vedere că *M* este foarte mare, formula (2.32) poate fi simplificată folosind cele mai semnificative *L* valori ale $f(x|o_k)$ pentru fiecare *x* fără ca performanța sistemului să fie afectată. Experiența a arătat că valorile potrivite pentru *L* sunt aproximativ 1-3% din valoarea lui *M*. Aceasta se poate obține ușor în timpul operației de cuantizare prin sortarea valorilor cuantizate (coeficienții de ponderare) și păstrarea a *L* cele mai semnificative valori. Fie $\eta(x)$ setul de *L* codeword-uri care au cel mai mare $f(x|o_k)$ pentru un *x* dat. Atunci, formula (2.32) devine:

$$b_j(x) \cong \sum_{k \in \eta(x)} p(x \mid o_k) b_j(k) \tag{2.33}$$

Deoarece numărul de mixturi $\eta(x)$ este mai mic decât *M* cu două ordini de mărime, formula (2.33) reduce semnificativ numărul de calcule. Dacă $\eta(x)$ conține doar valoarea cea mai semnificativă (adică $p(x|v_k)$), HMM semi-continuă devine un HMM discret. Pe de altă parte, dacă codebook-ul este mare, fiecare stare poate avea codeword-uri proprii ducând la cealaltă extremă adica la HMM continuu.

În cazul unui VQ codebook legat, reestimarea vectorilor de medii și a matricilor de covarianțe a diferitelor modele implică interdependențe. Dacă o observație x_t (indiferent de modelul care are cea mai mare probabilitate să-l fi generat) are o probabilitate mare a posteriori $\zeta_t(j,k)$, ea va avea o contribuție mare la reestimarea parametrilor codewordului o_k. Probabilitatea a posteriori pentru fiecare codeword se poate calcula cu formula (2.34):

$$\zeta_t(k) = p(x_t = o_k \mid X, \Phi) = \sum_j \zeta_t(j,k) \tag{2.34}$$

unde $\zeta_t(j,k)$:

$$\zeta_t(j,k) = \frac{p(X, s_t = j, k_t = k \mid \Phi)}{p(X \mid \Phi)} = \frac{\sum_i \alpha_{t-1}(i) a_{ij} c_{jk} b_{jk}(x_t) \beta_t(j)}{\sum_{i=1}^{N} \alpha_T(i)} \tag{2.35}$$

Formulele de reestimare pentru mixturile legate devin:

$$\bar{\mu}_k = \frac{\sum_{t=1}^{T} \zeta_t(k) x_t}{\sum_{t=1}^{T} \zeta_t(k)} \tag{2.36}$$

$$\hat{\Sigma}_k = \frac{\sum_{t=1}^{T} \zeta_t(k)(x_t - \hat{\mu}_k)(x_t - \hat{\mu}_k)'}{\sum_{t=1}^{T} \zeta_t(k)} \tag{2.37}$$

2.2.11. Limitări ale HMM

Modelarea duratei – este limitarea majoră al acestui model. În modul în care a fost definit modelul, probabilitatea de a rămâne în aceeaşi stare descreşte exponenţial cu timpul. Probabilitatea ca să rămână în stare *i* pentru *t* ferestre de timp este egală cu:

$$d_i = a_{ii}^t (1 - a_{ii}) \tag{2.38}$$

Modelarea staționarității – După cum se observă, staţionaritatea unui segment nu este reprezentată în mod potrivit de o singură stare al HMM. O metodă de a trece peste această problemă este introducerea lanţurilor Markov de ordin II. Astfel, probabilitatea de tranziţie între două stări în momentul de timp *t* va depinde de stările în care a fost procesul în momentele de timp *t-1* şi *t-2*. O astfel de metodă creşte în mod semnificativ complexitatea sistemului şi timpul de calcul.

2.2.12 Variabilitatea vorbirii

Variabilitatea vorbitorului – Fiecare vorbitor este diferit. Sunetul pe care el îl produce reflectă dimensiunea fizică a tractului vocal, grosimea şi lungimea gâtului, şi un număr de caracteristici fizice, vârsta, sexul, dialectul, sănătatea, educaţia, şi stilul personal [101]. Ca atare, fiecare formă de voce a unei persoane poate fi total diferită de cea a celorlalte persoane. Chiar dacă excludem aceste diferenţe între vorbitori, acelaşii vorbitor nu este întotdeauna în stare să pronunțe la fel acelaşi fonem.

Pentru recunoaştere de voce independentă de vorbitor, se folosesc în jur de 500 de vorbitori pentru a construi un model combinat. O asemenea metodă introduce variaţii mari între vorbitori pentru care avem puţine minute înregistrate. De aceea, pentru a creşte performanţa de recunoaştere putem introduce constrângeri de genul: fiecare vorbitor trebuie să aibă minim 30 de minute de vorbit.

O recunoaştere dependentă de vorbitor are performanţe mai bune cu 30%, în sensul ratei de cuvinte eronate, pentru aceaşi marime a bazei de date. Caracteristicile de vorbire ale unui vorbitor ajută la construirea unui model care să ofere o rată mare de recunoaştere. Dezavantajul recunoaşterii de voce dependentă de vorbitor este că necesită timp pentru colectarea datelor de la un singur vorbitor, ceea ce pentru anumite aplicaţii nu este cel mai practic mod. Majoritatea aplicaţiilor cer să se ofere suport şi pentru noi vorbitori, aşadar, recunoaşterea independentă de vorbitor rămâne un subiect interesant de studiu unde încă se pot aduce îmbunătăţiri considerabile.

Variabilitatea mediului – Lumea reală este plină de sunete diferite provenind din diferite surse. Când recunoaştere vorbirii este folosită în telefoane mobile de exemplu, spectrul de zgomot variază considerabil funcţie de cum se mişcă vorbitorul. Aceşti parametrii

externi, pot afecta performanţele recunoaşterii. Pe lângă zgomotul de ambianţă, trebuie luat în considerare şi zgomote produse chiar de vorbitor, de microfon, sau chiar de convertorul A/D.

Similar modului de antrenare pentru sisteme independent de vorbitor, putem construi un sistem folosind o cantitate mare de date dintr-un număr de medii diferite. Aceasta se numeşte antrenare multistil. Apoi, se pot folosi tehnici de adaptare similare antrenării adaptate vorbitorului. Deşi s-au făcut destule progrese în acest sens, variabilitatea mediului rămâne una dintre cele mai dificile probleme de rezolvat.

Variabilitatea de stil – Acelaşii vorbitor după starea emoţională poate vorbi mai lent sau mai repede, mai tare sau mai încet. Aşadar, sistemele de recunoaştere trebuie antrenate pentru a include şi aceste variaţii [12].

2.2.13 Creşterea robusteţii sistemului de recunoaştere

Un sistem de recunoaştere antrenat în laborator cu voce de calitate bună (fără zgomot) poate întâlni dificultăţi mari dacă nu se potrivesc vocea din laborator cu cea obţinută în condiţii reale. Capabilitatea de a avea performanţe bune sub condiţii adverse se numeşte robusteţe. Factorii adverşi sunt mulţi, începând cu zgomotul de ambianţă şi orice alt factor care distorsionează semnalul vocal.

Sursele de distorsiuni ale semnalului vocal le putem sumariza:

- Zgomotul aditiv (zgomot de ambianţă)
- Ecoul
- Răspunsul în frecvenţă al microfonului
- Răspunsul în frecvenţă al filtrului convertorului A/D
- Zgomotul de cuantizare

Există studii despre fiecare factor în parte şi în urma lor s-au găsit diferite soluţii care minimizează efectul lor. Aceste aspect sunt discutate în detalii în Capitolul 3.

2.3. PARAMETRII DE VOCE UTILIZAȚI ÎN RECUNOAȘTEREA LIMBAJULUI VORBIT

Parametrii de voce utilizaţi în domeniul recunoaşterii vorbirii pot fi coeficienții de predicție liniară LPC (Linear Predictive Coefficients), coeficienții cepstrali din scara Mel MFCC (Mel Frequency Cepstral Coefficients), coeficienții perceptuali de predicție liniară PLP (Perceptual Linear Prediction), etc. Pentru un semnal vocal folosit pentru antrenare sau recunoaştere aceşti parametrii se calculează la fiecare fereastră de timp aleasă şi formează vectorul de observaţii. Ieşirea unui HMM este şi ea un vector de aceşti parametri cu probabilităţile respective.

Parametrii utilizați în recunoașterea vorbirii trebuie să aibă câteva caracteristici esențiale. În cazul în care parametrii sunt folosiți pentru compresie, important este ca semnalul reconstruit să fie cât mai aproape de semnalul original. La recunoaștere procesul de decodare nu are loc și atunci criteriile de evaluare sunt altele. Același fonem pronunțat de persoane diferite sau de în contexte diferite poate fi reprezentat printr-un set de parametrii care ia valori diferite în fiecare caz. Din punct de vedere al recunoașterii de vorbire, este

apreciată acea reprezentare pentru care valorile parametrilor corespunzătoare diferitelor pronunțări ale aceluiași fonem sunt similare. Cum recunoașterea este un proces de decizie, prezintă interes parametrii, ai căror valori ocupă un volum cât mai mic în spațiul multidimensional pentru un fonem dat, iar volumurile corespunzătoare fonemelor diferite să fie disjuncte. Aceste criterii sunt indeplinite cel mai bine de către parametrii MFCC şi PLP după cum arată şi rezultatele experimentale (vezi capitolul 5.1.3.).

În această lucrare, majoritatea experimentelor au fost făcute folosind parametrii MFCC. Din acest motiv, obținerea parametrilor este descrisă în detaliu, iar obținerea parametrilor PLP este descrisă doar în principiu.

2.3.1. Parametri de voce MFCC

Parametrii MFCC diferă de cepstru real prin faptul că la calculul lor se folosește o scară neliniară de frecvență care aproximează sistemul auditiv al omului. Studiile au demonstrat că această reprezentare ajută în procesul de recunoaștere a vorbirii [50].

Pentru a determina coeficienţii cepstrali, spectrul calculat prin FFT este netezit printr-un banc de filtre triunghiulare centrate pe un set de frecvenţe aflate pe scara Mel. Pentru un semnal vocal cu lărgime de bandă de 8 kHz, un număr de 24 de bancuri de filtre sunt suficiente pentru a calcula parametrii MFCC. Cu toate acestea, în sistemele de recunoaștere numărul de bancuri de filtre este configurabil și se poate experimenta cu diferite valori ale acestuia. Pentru ca distribuţia coeficienţilor să urmeze o lege Gaussiană compresia logaritmică este aplicată la ieşirea bancurilor de filtre [103]. Ultima etapă constă în aplicarea transformatei cosinus discretă (*DCT – Discrete Cosinus Transform*) care are capacitatea de a concentra informaţia spectrală într-un număr mai mic de parametrii dar şi de a decorela aceste valori.

În prima etapă se calculează transformata Fourier discretă (DFT) a semnalului de intrare cu formula (2.39):

$$X_a[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi nk/N} \ , \ 0 \le k \le N \tag{2.39}$$

Apoi se definește un banc de M filtre (m=1,2,...,M), unde filtru m este triunghiular specificat prin formula (2.40):

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{2(k - f[m-1])}{(f[m+1] - f[m-1])(f[m] - f[m-1])} & f[m-1] \le k \le f[m] \\ \dfrac{2(f[m+1] - k)}{(f[m+1] - f[m-1])(f[m+1] - f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{2.40}$$

Aceste filtre calculează spectrul mediu în jurul frecvenței centrale a fiecărei benzi. Banda lor crește odată cu indexul *m* (Fig. 2.8.).

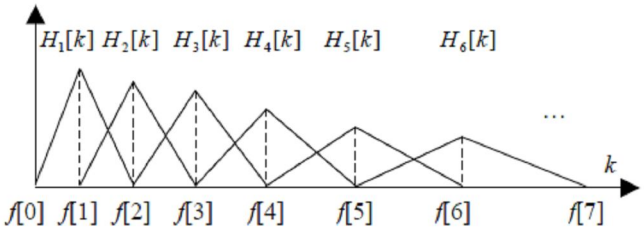

Fig. 2.8. Benzile de frecvență în scara Mel

Un alt mod de a calcula aceste filtre este dat de formulele (2.41):

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \dfrac{(k - f[m-1])}{(f[m] - f[m-1])} & f[m-1] \le k \le f[m] \\ \dfrac{(f[m+1] - k)}{(f[m+1] - f[m])} & f[m] \le k \le f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{2.41}$$

În acest caz este satisfăcută condiția $\sum_{m=0}^{M-1} H'_m[k] = 1$. Cepstrul calculat prin $H_m[k]$ și cel calculat prin $H'_m[k]$ diferă doar printr-un vector de constante pentru toate semnalele de intrare, așa că alegerea unei dintre variante devine neimportantă în cazul recunoașterii vorbirii atâta timp când se folosește același filtru peste tot.

Fie f_l și f_h frecvența cea mai joasă și respectiv cea mai înaltă a bancului de filtre în Hz, F_s frecvența de eșantionare în Hz, M numărul de filtre și N dimensiunea ferestrei FFT. Punctele de graniță $f[m]$ sunt așezate uniform, în scara Mel:

$$f[m] = \left(\frac{N}{F_s}\right) B^{-1}\left(B(f_1) + m\frac{B(f_h) - B(f_l)}{M+1}\right) \tag{2.42}$$

unde scara Mel B este:

$$B(f) = 1125\ln(1 + f/700) \tag{2.43}$$

iar inversul ei B^{-1} este:

$$B(f) = 700(\exp(b/1125) - 1) \tag{2.44}$$

Apoi, se calculează energia logaritmică la ieșirea fiecărui filtru:

$$S[m] = \ln\left[\sum_{k=0}^{N-1} |X_a[k]|^2 H_m[k]\right],\ 0 \le m \le M \tag{2.45}$$

Coeficienții cepstrali ai scării Mel sunt calculați prin transformata cosinus discretă (DCT), pentru cele *M* valori ale energiei logaritmice:

$$c[n] = \sum_{m=0}^{M-1} S[m]\cos(\pi n(m+1/2)/M),\ 0 \le n \le M \tag{2.46}$$

unde *M* variază funcție de implementare de la 24 până la 40. Pentru recunoașterea de vorbire, se aleg, de obicei, 13 coeficienți cepstrali. Trebuie observat că, calculul parametrilor MFCC nu este o transformare homomorfă. Ar fi fost o transformare homomorfă dacă s-ar inversa ordinea sumării și a logaritmării din formula (2.45):

$$S[m] = \sum_{k=0}^{N-1} \ln\left(|X_a[k]|^2 H_m[k]\right),\ 0 \le m \le M \tag{2.47}$$

În practică, calculul parametrilor MFCC este aproximativ homomorfic pentru filtre care au o funcție de transfer netedă. Avantajul reprezentării MFCC calculate cu formula (2.47) în locul formulei (2.45) este că energiile filtrelor respective sunt mai robuste față de erorile de zgomot și cele de estimare a spectrului.

2.3.2. Parametri de voce PLP

În tehnica PLP, proprietățile sistemului auditiv sunt simulate prin diferite aproximări. Spectrul semnalului vocal care rezultă din aceste aproximări este modelat cu un model "numai poli" autoregresiv. Această metodă a fost propusă pentru prima oară în [93]. În continuare, metoda a fost îmbunătățită cu diferite optimizări. Schema bloc pentru obținerea parametrilor PLP este prezentată în Fig. 2.9.

Analiza benzilor critice se face similar analizei cepstrale, dar în acest caz curbele de mascare a benzilor critice sunt asimetrice. Această conversie particulară între scările Hertz și Bark este descrisă în detaliu în [180]. Pre-accentuarea ține cont de neuniformitatea sensibilității sistemului auditiv uman la diferite frecvențe și simulează sensibilitatea acestuia la nivelul 40 dB. În cazul analizei PLP această aproximare este adoptată din [135]. Operația de conversie a intensitatății este făcută după legea de putere pentru sistemul auditiv uman [189]. Împreună cu pre-accentuarea psico-fizică a intensității, această operație contribuie și la reducerea variației amplitudinii spectral. Acest aspect ajută la modelarea "numai poli", care poate fi de ordin mic. Detalii despre modelarea spectrală "numai poli" se găsesc în [134]. Primele *M* + 1 valori ale autocorelației sunt folosite pentru a rezolva equațiile Yule-Walker pentru găsirea coeficienților modelului "numai poli" de ordin *M*.

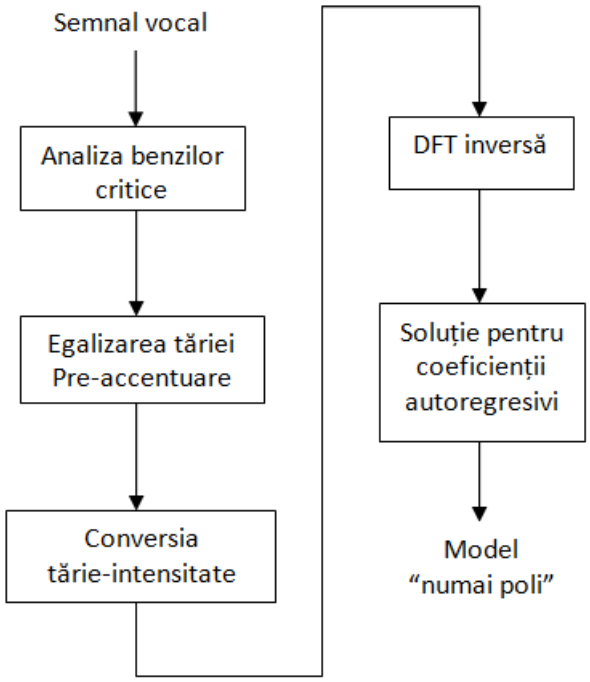

Fig. 2.9. Schema bloc pentru analiza PLP

2.3.3. Importanța variațiilor temporale ale parametrilor de voce

Variaţiile temporale ale caracteristicilor spectrale joacă un rol extrem de important în percepţia umană. Pentru diferite foneme aceste caracteristici variază foarte mult. Astfel, ele pot ajuta la discriminarea diferitor unităţi de limbă. O modalitate de a îngloba această informaţie este utilizarea coeficienţilor delta, ca o măsură a modificărilor parametrilor de bază în timp. Un sistem tipic de recunoaştere se bazează pe următorii parametrii:

- N (tipic 12) coeficienţi de bază (c_k care pot fi LPC, MFCC, PLP sau combinaţii ale acestora)
- N coeficienţi de variaţie de ordinul 1, Δc_k, calculaţi cu formula (2.48):

$$\Delta c_k = c_{k-1} - c_{k+1} \tag{2.48}$$

- N coeficienţi de variaţie de ordinul 2, $\Delta\Delta c_k$, calculaţi cu formula (2.49):

$$\Delta\Delta c_k = \Delta c_{k-1} - \Delta c_{k+1} \tag{2.49}$$

În capitolul 5.1.3., s-a tratat efectul acestor parametrii în acurateţea recunoaşterii de vorbire.

2.4. MODELAREA LINGVISTICĂ

Modelul acustic discutat la capitolul 2.2. şi cunoştinţele de limbă sunt la fel de importante în recunoaşterea limbajului vorbit. Cunoştiinţele lexicale, sintaxa şi semantica unei limbi sunt necesare pentru a determina dacă o secvenţă de cuvinte care a rezultat din recunoaştere este corectă şi are sens. *Gramatica* este o specificaţie a structurilor permisibile

pentru o anumită limbă. Tehnica *parcurgerii* este metoda prin care se analizează propoziţia pentru a vedea dacă structura ei îndeplineşte contstrângerile impuse de gramatică. Folosind cantităţi mari de texte din diferite domenii este posibil să generalizăm gramatica formală şi să calculăm probabilităţi precise în legătură cu ordinea cuvintelor în propoziţie. Relaţia probabilistică între secvenţele de cuvinte poate fi dedusă din texte prin mijlocul aşa numitelor modele lingvistice stohastice, ca de exemplu *n-gramele.*

Un n-gram este o secvenţă de *n* simboluri (de exemplu cuvinte) şi un model lingvistic bazat pe n-grame îşi propune să prezică fiecare simbol din secvenţă dacă se ştiu cele *n*-1 simboluri precedente. Se pleacă de la presupunerea că probabilitatea ca un specific n-gram să apară într-o secvenţă de text necunoscută se poate calcula din aparițiile sale în nişte texte folosite pentru antrenare. Aşadar, construcţia n-gramelor, aşa cum este arătat şi în Fig. 2.10. de mai jos este un proces cu trei etape. Pentru început, textul folosit pentru antrenare se analizează, se numără n-gramele şi se stochează în nişte structuri numite *gram* [45]. În etapa a doua, câteva cuvinte se pot încadra în categoria "cuvinte ce nu aparţin dicţionarului" şi apoi în ultima etapă, numărătorile din fişierele gram sunt folosite pentru calculul probabilităţilor n-gramelor şi care sunt stocate într-un fişier de model lingvistic. În sfârşit, pentru a testa cât de bine acest model lingvistic reflectă realitatea unei limbi, se calculează o metrică numită *perplexitate.* Perplexitatea măsoară câte cuvinte echiprobabile pot urma după un cuvânt dat. Această metrică măsoară doar cât de bine un model lingvistic reprezintă o limbă şi nu este foarte corelat cu performanța RLV, pentru că nu ține cont de similitudinile de pronunție între cuvinte. Perplexitatea se măsoară pe un nou text folosit ca test. În general, un model lingvistic este cu atât mai bun cu cât perplexitatea lui este mai mică [41].

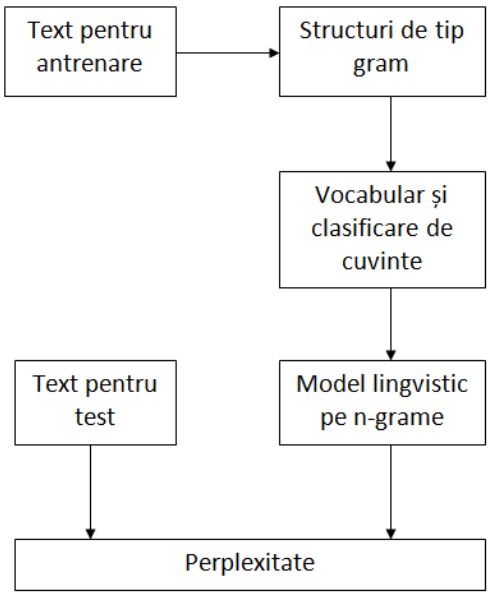

Fig. 2.10. Construcţia n-gramelor

Deşi principiul de bază al unui model lingvistic bazat pe n-grame este simplu, în practică exista multe posibile n-grame care nu pot fi niciodată culese în texte de antrenare. De

asemenea, utilizând un text de antrenare finit, un singur model lingvistic nu se poate potrivi în toate materialele de test care variază mult. De exemplu, un model lingvistic antrenat cu date din ziare nu poate fi un model bun pentru poezii. O altă dificultate o prezintă dimensiunea finită a dicţionarului care a fost fixat la momentul construirii unui model lingvistic bazat pe n-grame. În aceste condiţii, se pot prezice doar secvenţe cu cuvinte din dicţionar, iar dacă se vor introduse noi cuvinte, modelul lingvistic trebuie reconstruit.

O încercare de a aplica modele de limbă pentru limba română s-a făcut în [47], unde s-a plecat de la un corpus achiziționat de pe Internet. Pentru a obține un model de limbă pentru un domeniu mai restrâns s-a folosit un corpus în limba franceză tradus automat cu Google Translate.

CAPITOLUL 3

RECUNOAȘTERA LIMBAJULUI VORBIT ÎN REȚELE DE TELECOMUNICAȚII MOBILE

3.1. NECESITĂŢI, TENDINŢE ŞI STANDARDE

Numarul de abonaţi la retele de comunicaţii mobile creşe în fiecare zi cu paşi uriaşi. După un studiu facut de CBSNews [99] numărul de abonaţi a atins în 2010 cifra de 4 miliarde. În timp ce telefonia rămâne principala funcţie a telefoanelor mobile, un număr mare de aplicaţii sofisticate au început să se implementeze pe acestea. Acum un telefon mobil este echipat cu camera, web browser, jocuri, video şi muzică. Cum noi tehnologii de reţele 3G şi 4G sunt implementate din ce în ce mai mult pe gamă largă, transferul unei cantităţi mari de informaţie este posibil acum la viteze mari.

Interfaţa cu utilizatorul, în schimb, nu a fost dezvoltată în mod semnificativ. Pentru a interacţiona cu utilizatorul toate echipamentele au nevoie de o tastatură ceea ce impune şi anumite dimensiuni fizice. În acest sens comunicaţia prin voce prezintă multe avantaje. Pe lângă confortul oferit, comunicaţia prin voce poate fi şi mai rapidă decât tastarea comenzilor pe tastatură. Gradul de utilizare al acestei soluții depinde de un număr mare de factori: calitatea sistemului de recunoaştere de voce, arhitectura aleasă, calitatea reţelei de telecomunicaţii, etc.

Recunoaşterea limbajului vorbit (RLV) este un proces consumator de resurse. După cum se ştie, telefoanele mobile sunt caracterizate ca echipamente cu memorie limitată, putere de procesare limitată, şi viaţă de baterie limitată. Din aceste motive se consideră posibilitatea ca o parte din acest proces să fie desfaşurat pe un calculator la distanță, echipat cu procesor şi memorie suficientă pentru a realiza o recunoaştere de voce în timp real. Un alt avantaj al utilizării unei entităţii centrale de recunoaştere de voce este uşurinţa cu care se poate interveni în aplicaţie în situaţii de trecere de la o versiune a software-ului la alta. Astfel orice implementare de aplicaţie care implică recunoaştere de voce va avea un cost scăzut.

În aceste condiţii, devine importantă analizarea distorsiunilor introduse în reţea şi impactul lor asupra recunoaşterii. Pe lângă pierderilor de pachete în reţea, un alt inconvenient este întârzierea, care are un rol critic în aplicaţii de timp real. În majoritatea cazurilor comunicaţia între telefon mobil şi server care realizează recunoaşterea se face prin canale digitale de bandă limitată, ceea ce implică utilizarea unor codoare pentru compresie. Parametrii de voce folosiţi de codor pentru compresie trebuie aleşi astfel încât să coincidă sau să fie uşor deductibili din parametrii de voce folosiţi de sistemul de recunoaştere. Astfel se limitează zgomotul introdus în procesele de codare şi de decodare a vocii. Un alt aspect important este zgomotul de ambianţă sau cel introdus de rețeaua de comunicație. Studiul principalelor surse de zgomot devine necesar pentru înlăturarea acestuia.

Pentru a ajuta dezvoltarea acestor sisteme pe o scară largă s-au implementat diferite standarde de catre diferite instituții/organizații. Una dintre cele mai mari încercări în acest sens a fost făcută de către ETSI (European Telecomunications Standards Institute) cu programul său Aurora, pentru dezvoltarea unui SRLV distribuitiv [100]. ETSI a venit cu o suită de standarde şi definiţii care privesc diferite funcţii în procesul de recunoaştere. Sistemul de recunoaştere privit ca unul modular face posibil ca anumite funcţii să fie implementate pe entităţi fizice distincte, permiţând astfel o flexibilitate sporită în implementarea unui SRLV pe o reţea de comunicaţii [61], [62], [63], [64].

3.2. ARHITECTURA SISTEMELOR NSR, DSR ŞI ESR

În procesul de recunoaştere al limbajului vorbit, există o serie de funcţii care pot fi implementate ca unele componente disjuncte. Aceste funcţii sunt prezentate şi în Fig. 3.1. După ce este primit un semnal vocal, din el se extrag parametrii de voce necesar recunoaşterii (de exemplu parametrii MFCC prezentat în capitolul 2.3.1.). Apoi din aceşti parametrii se generează secvenţa de text (transcrierea) corespunzătoare semnalului vocal pe baza unui model acustic, unui dicţionar fonetic (model fonetic) şi al unui model lingvistic (model de limbă).

În funcţie de locul unde sunt implementate aceste funcţii, pe client (telefon mobil) sau pe server (calculator puternic în distanţă), sistemul se numeşte NSR (Network-based Speech Recognition), DSR (Distribuitive Speech Recognition) sau ESR (Embedded Speech Recognition).

Deşi modelul acustic este şi el strâns legat de aplicaţie, în sensul că se poate adapta cerinţelor aplicaţiilor, cel mai mult legat de aplicaţie este modelul lingvistic, mai ales modelul lingvistic bazat pe reguli. Cu toate acestea, modelele lingvistice stohastice, care sunt create pe baza de n-grame, sunt independente de aplicaţie şi se pot crea offline.

Dacă aruncăm o privire pe componentele SRLV, vom observa că modulul de calcul al parametrilor de voce nu este foarte consumator de resurse. Din acest motiv el se poate plasa şi pe client, adică pe telefonul mobil. Pe de altă parte, decodorul Viterbi (vezi capitolul 2.2.5) bazat pe HMM-uri necesită multă memorie şi putere de procesare sporită, pentru că numărul de parametri în modelul acustic este mare. Apoi, dicţionarul şi modelul lingvistic au nevoie de spaţiu mare în memorie. De aceea, această soluție nu este foarte răspândită în dispozitivele mobile de astăzi.

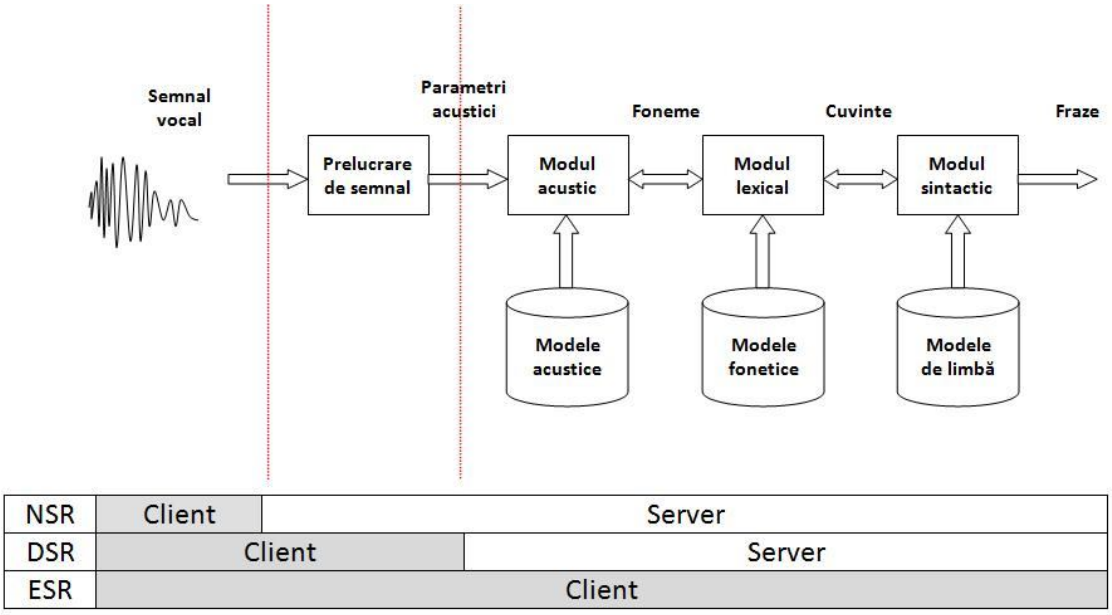

Fig. 3.1. Arhitectura sistemelor NSR, DSR, ESR

3.3. PROBLEME SPECIFICE IMPLEMENTĂRII PE DISPOZITIVE MOBILE

Problemele principale care apar pe telefoanele mobile (sau orice alt dispozitiv care oferă mobilitate) sunt legate de puterea de procesare redusă, aritmetica în virgulă fixă, capacitatea de memorie redusă, viteza mică de acces al memoriei şi consumul de putere (timpul de viaţă al bateriei). Aceste echipamente se pot împărţi în două categorii: dispozitive cu resurse largi (*high-end*) şi dispozitive cu resurse reduse. Din prima categorie fac parte PDAs (din engleză Personal Digital Assistant), telefoanele inteligente (în engleză Smart Phones) sau diferite dispozitive utilizate în maşină (*Car Kits*). Când un client cumpără un telefon inteligent se aşteaptă ca el să aibă aplicaţii avansate cum ar fi telefonie cu video sau TV mobil. Pentru a implementa aceste aplicaţii sunt necesare o putere de calcul ridicată şi o capacitate mare de memorie. În acest context, pot să profite şi aplicaţiile bazate pe recunoaştere de voce cum ar fi dictarea, voice dialing, etc. Pe de altă parte telefoanele simple care au ca scop principal telefonia, nu dispun de aceste resurse. Aşadar, în acest caz devine obligatorie folosirea arhitecturilor DSR şi NSR.

Timpul de viaţă al bateriei (aproximativ 5-7h timp de vorbire pe un telefon mobil) este o constrângere majoră pentru că algoritmii de procesare a semnalului vocal, stocarea unei cantităţi mari de date cu acces rapid implică un consum ridicat de putere. Pe lângă acestea, aplicaţiile video consumă şi mai multă putere. O soluţie ar fi ca producătorii sa mărească timpul de viaţă a bateriei nedepăşind totuşi anumite dimensiuni fizice ale telefonului. O altă soluţie ar fi să se implementeze algoritmi optimizaţi pentru consum redus de putere. În această direcţie s-au şi făcut anumite studii care au adus anumite optimizări.

Memoria scăzută în aceste dispozitive este o problemă care merită atenţie şi care trebuie avută în vedere la implementarea recunoaşterii vorbirii. Telefoanele obişnuite sunt echipate cu memorie RAM de acces lent 4-16MB şi cu o memorie cache de 32kB. Deci, numărul de algoritmi de procesare a semnalului vocal care pot rula simultan este limitat. La fel de limitate sunt dimensiunea modelelor lingvistice şi acustice. Rezultatul va fi o performanţă de compromis.

Puterea de calcul fiind şi ea limitată implică utilizarea metodelor suboptimale în recunoaşterea de vorbire şi în consecinţă şi performanţe scazute. În plus, proiectantul care implementează algoritmii nu are acces la nivelul de jos în sistemul de operare pentru optimizare, iar programarea de nivel înalt este într-adevăr mai confortabilă, dar mai puţin eficientă. De asemenea, procesorul lucrează cu aritmetică în virgulă fixă care implică implementarea algoritmilor în virgulă fixă sau utiliziarea unei aritmetici în virgulă mobilă emulată pe hardware-ul de virgulă fixă al procesorului.

Lipsa resurselor devine şi mai sensibilă atunci când se lucrează în condiţii dificile ale mediului. Zgomotul ambiental şi ecoul implică algoritmi specializaţi pentru eliminarea acestora sau al efectului acestora, iar în cazul telefoanelor mobile aceşti factori adverşi sunt de multe ori prezenţi. Aceşti algoritmi necesită resurse suplimentare iar implementarea lor nu este întotdeauna posibilă din cauza constrângerilor menţionate mai sus.

Ultima preocupare este legată de aspecte ce privesc caractiristicile telefoanelor mobile. Vocea folosită pentru diferite comenzi trebuie sa concureze interfaţa de utilizator existentă. Metodele actuale de navigare prin tipărire şi *touch screen* au devenit populare iar introducerea comenzilor prin voce trebuie să fie cel puţin la acelaşi nivel de intuitivitate şi timp de răspuns.

3.4. ROBUSTEŢEA FAŢĂ DE ZGOMOT

În capitolul 2.2.12, au fost prezentate câțiva factori de variabilitatea vorbirii, una dintre ele fiind variația mediului. Una dintre formele de schimbare ale mediului este zgomotul ambiental pentru eliminarea căruia se folosesc algoritmii de compensare a zgomotului, mai ales dacă avem un număr mic de date de adaptare. Altfel, se folosesc metodele de adaptare prezentate în capitolul 2.5.

Algoritmii de compensare a zgomotului pleacă de obicei de la premisa că un semnal curat de voce x_t în domeniul timp este alterat de un zgomot aditiv n_t şi de un răspuns la impuls convolutiv (al canalului de telecomunicaţii) h_t, rezultând, astfel, semnalul zgomotos y_t [1]:

$$y_t = x_t \otimes h_t + n_t \tag{3.1}$$

unde cu ($\otimes$) este notată operaţia de convoluţie.

În practică, un SRLV operează cu parametrii de voce extraşi direct din spectru, sau alte transformări ale lui. Dacă se folosesc parametri de voce din domeniul cepstral (cum sunt de exemplu parametrii MFCC), formula (3.1) duce la următoarea relaţie între vocea curată, zgomot şi semnal alterat:

$$y_t^s = x_t^s + h + C\log(1+\exp(C^{-1}(n_t^s - x_t^s - h))) = x_t^s + f(x_t^s, n_t^s, h) \tag{3.2}$$

unde C este matricea DCT (transfromata cosinus discretă). Pentru un set dat de condiţii de zgomot, vectorul de observaţii y^s_t este o funcţie neliniară de semnalul curat x^s_t. Cu alte cuvinte, cu scăderea raportului semnal-zgomot (RSZ) al doilea termen din formula (3.2) devine dominant, mediile vectorilor tind către mediile zgomotului, iar varianţele devin mai mici. La un RSZ foarte mic, zgomotul domină şi nu rămâne decât puţină informaţie în semnal. Efectul este că distribuţiile de probabilitate antrenate cu semnale curate de voce nu mai reprezintă distribuţiile vectorilor de observaţie din medii zgomotoase şi din acest motiv rata de recunoaştere descreşte.

Problema care se ridică este construirea unor modele care să aibă imunitate faţă de zgomot. De exemplu, parametrii PLP prezintă o robusteţe mai mare faţă de ceilalţi parametri [94]. O îmbunătăţire ulterioară se poate face printr-o filtrare trece-bandă a parametrilor în timp într-un proces numit filtrare spectrală relativa care duce la aşa-numiţii coeficienţi RASTA-PLP [94]. Cu toate acestea, capacitatea acestor tehnici pentru reducerea efectului zgomotului este limitată.

Există două metode principale pentru obţinerea robusteţii faţă de zgomot, compensarea parametrilor şi compensarea modelelor. Aceste metode sunt prezentate în Fig. 3.2. care ia în considerare cele două medii, cel curat de antrenare şi cel zgomotos în care se face recunoaşterea. În metoda compensării parametrilor, parametrii de voce sunt corectaţi sau "îmbunătăţiţi" pentru eliminarea efectului zgomotului. Recunoaşterea şi antrenarea în acest caz se fac într-un mediu curat (mediu rămâne în continuare zgomotos dar corectarea parametrilor de voce ne pune în condiţiile unui semnal curat vezi Fig. 3.2.). În metoda compensării modelului, modelele acustice obţinute din antrenarea cu semnale curate sunt corectate cu scopul de a reprezenta semnalul zgomotos. În acest caz, antrenarea se face într-un mediu curat, dar modelele sunt modificate astfel încât să poată lucra într-un mediu zgomotos. În general, compensarea parametrilor este mai simplă şi uşor de implementat, dar compensarea modelelor poate oferi o robusteţe mai mare pentru că se poate folosi direct de cunoştiinţele detaliate despre semnalul original care este modelat de HMM. Mai departe, atât metoda compensării parametrilor cât şi cea a compensării modelelor pot fi folosite pentru stabili o măsură de încredere sau de incertitudine pentru fiecare parametru. Metodele bazate pe incertitudine sunt metoda „parametrilor lipsă" şi decodarea sub incertitudine.

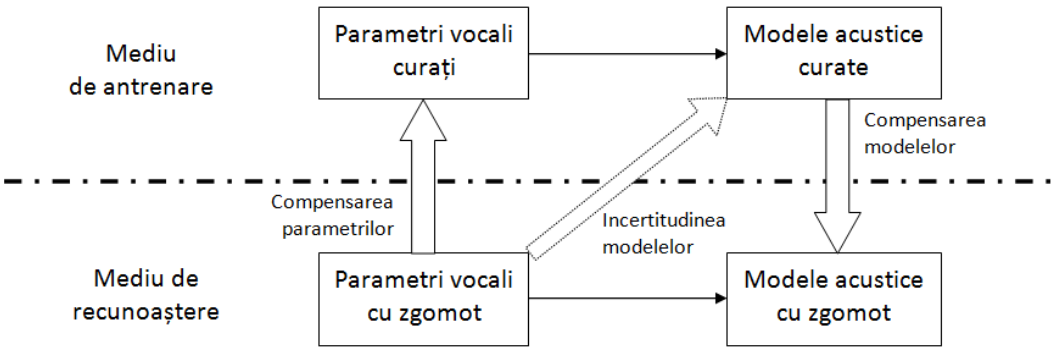

Fig. 3.2. Metode pentru eliminarea efectului zgomotului

3.4.1. Metoda compensării parametrilor

Scopul principal al metodei compensării parametrilor este eliminarea efectelor zgomotului din parametrii de voce măsuraţi. Una dintre cele mai simple soluţii la această problemă este scăderea spectrală (SS) [20]. Dat fiind un estimat al spectrului zgomotului în domeniul frecvenţă n^f, spectrul semnalului curat este calculat prin simpla scădere a n^f din spectrul observaţiei. Celelalte reprezentări (ca de ex. MFCC), se pot calcula din spectrul corectat astfel. Deşi SS este folosit des, totuşi această metodă prezintă câteva probleme. În primul rând, este necesară o detecție a vocii pentru izolarea segmentelor de spectru care conțin zgomotul care trebuie estimat. Această procedură este expusă diferitelor erori. În al doilea rând, netezirea poate duce la un subestimat al zgomotului, iar erorile ocazionale pot duce la un supraestimat al zgomotului. Aşadar, formula pentru obținerea semnalui curat estimat prin metoda SS poate fi:

$$\hat{x}_i^f = \begin{cases} y_i^f - \alpha n_i^f, & dacă \quad \hat{x}_i^f > \beta n_i^f \\ \beta n_i^f, & în \quad rest \end{cases} \tag{3.3}$$

unde f indică domeniul frecvență, α este un factor de supra-scădere determinat empiric şi β este un prag de jos care evită posibilitatea ca spectrul curat estimat să devină negativ [133].

O metodă mai robustă este cea prin calcularea minimului erorii pătratice medie (MMSE, din engleză Minimum Mean Square Error):

$$\hat{x}_t = \varepsilon\{x_t \mid y_t\} \tag{3.4}$$

Distribuția semnalului zgomotos observat este reprezentat ca o mixtură de gaussiene cu N componente:

$$p(y_t) = \sum_{n=1}^{N} c_n N(y_t; \mu_y^{(n)}, \Sigma_y^{(n)}) \tag{3.5}$$

Presupunând că componentele mixturii sunt independente atunci pentru fiecare componentă a mixturii avem:

$$\varepsilon\{x_t \mid y_t, n\} = \mu_x^{(n)} + \Sigma_{xy}^{(n)} (\Sigma_y^{(n)})^{-1} (y_t - \mu_y^{(n)}) = A^{(n)} y_t + b^{(n)} \tag{3.6}$$

Deci, estimatul MMSE devine o combinație liniară de predicții:

$$\hat{x}_t = \sum_{n=1}^{N} P(n \mid y_t)(A^{(n)} y_t + b^{(n)}) \tag{3.7}$$

unde probabilitatea a posteriori pentru componenta n este dată de formula:

$$P(n \mid y_t) = \frac{c_n N(y_t; \mu_y^{(n)}, \Sigma_y^{(n)})}{p(y_t)} \tag{3.8}$$

În diferite studii de cercetare s-au propus diferiți algoritmi pentru calculul parametrilor $A^{(n)}$, $b^{(n)}$, $\mu^{(n)}$ și $\Sigma^{(n)}$ [1], [52], [58], [148].

3.4.2. Metoda compensării modelelor

Metoda compensării parametrilor de voce este limitată de necesitatea de a folosi un model simplu pentru semnalul vocal, de ex. GMM. Fiindcă modelele acustice din decodor oferă o reprezentare mai detaliată a semnalului vocal, atunci se pot obține rezultate mai bune prin transformarea acestor modele pentru a le adapta noilor condiții de zgomot. De obicei, compensarea modelelor se bazează pe combinarea modelului obținut din antrenarea cu semnal curat cu un model de zgomot și în majoritatea cazurilor o singură gausiană este de ajuns pentru modelarea zgomotului. Cu toate acestea, pentru anumite situații în care zgomotul este variabil, este necesar un model cu mai multe stări și mai multe componente ale mixturii gaussiene. Acest lucru afectează complexitatea de calcul atât pentru compensare cât și pentru decodare. Dacă modelul are *M* componente gaussiene și zgomotul *N* componente gaussiene, atunci modelul combinat va avea *MN* componente. Similar, se mărește și numărul de stări al modelului combinat dacă modelul pentru zgomot are mai multe stări. În general, însă, o singură gaussiană este de ajuns pentru modelarea zgomotului și decodorul poate rămâne nemodificat [74], [201].

Principiul compensării modelului este obținerea distribuțiilor pentru parametrii semnalului afectat de zgomot din modelul curat și modelul zgomotului. Se presupune că dacă distribuțiile modelelor semnalului vocal și ale zgomotului sunt gaussiene atunci și distribuția modelului combinat va fi tot gaussiană (distribuțiile presupunându-se a fi independente).

Inițial, se consideră doar mediile și variațele ale parametrilor statici (adică nu și accelerațiile parametrilor vezi capitolul 2.3.3.). Parametrii distribuției semnalului afectat de zgomot pot fi calculate cu formulele (3.9) și (3.10):

$$\mu_y = \varepsilon\{y^s\} \tag{3.9}$$

$$\Sigma_y = \varepsilon\{y^s y^{sT}\} - \mu_y \mu_y^T \tag{3.10}$$

unde y^s sunt observațiile afectate de zgomot obținute dintr-o anumită componentă a modelului acustic al semnalului curat combinat cu „observația” de zgomot din modelul zgomotului. Nu există o soluție directă pentru calculul mediei și al variației zgomotului, de aceea se folosesc diferite metode de aproximare.

Una dintre metodele de compensare a modelelor este combinația modelelor paralele (PMC, din engleză Parallel Model Combination) [72]. PMC standard presupune că parametrii de voce folosiți sunt transformări liniare ale spectrului logaritmic, de ex. MFCC, și că zgomotul este aditiv. Există și alte metode care consideră cazul zgomotului convolutiv ca în [73].

Principiul de bază este conversia mediilor şi ale variaţelor gaussienelor din domeniul cepstral în cel liniar unde zgomotul este aditiv, calcularea noilor medii şi variaţe corespunzătoare semnalului afectat de zgomot şi readucerea parametrilor în domeniul cepstral.

Construcţia unei distribuţii gaussiene $N(\mu,\Sigma)$ în domeniul log-spectral (notat cu l) ştiind parametrii distribuţiei în domeniul cepstral, se poate face prin DCT inversă:

$$\mu^l = C^{-1}\mu \tag{3.11}$$

$$\Sigma^l = C^{-1}\Sigma(C^{-1})^T \tag{3.12}$$

unce *C* reprezintă DCT. Trecerea din domeniul log-spectral în domeniul liniar (notat cu *f*) se face cu formulele (3.13) şi (3.14):

$$\mu_i^f = \exp\{\mu_i^l + \sigma_i^{l2}/2\} \tag{3.13}$$

$$\sigma_{ij}^f = \mu_i^f \mu_j^f (\exp\{\sigma_{ij}^l\} - 1) \tag{3.14}$$

Deoarece vocea şi zgomotul sunt independente, atunci parametrii semnalului afectat de zgomot se obţin prin însumarea parametrilor distribuţiei semnalului de voce şi ai parametrilor distribuţiei zgomotului.

$$\mu_y^f = \mu_x^f + \mu_n^f \tag{3.15}$$

$$\Sigma_y^f = \Sigma_x^f + \Sigma_n^f \tag{3.16}$$

Distribuţiile în domeniul liniar sunt normale logaritmate, dar suma a două distribuţii normale logaritmate nu este şi ea log-normală. PMC standard ignoră acest aspect şi aproximează suma distribuţiilor log-normale cu o altă distribuţie normală logaritmată. În acest caz, se fac transformările inverse pentru μ^f_y şi Σ^f_y pentru a trece înapoi în domeniul cepstral. Aproximaţia normală logaritmică de mai sus nu este foarte precisă, mai ales la RSZ mici. O metodă mai precisă s-ar obţine folosind tehnica Monte Carlo pentru eşantionarea distribuţiilor de voce şi de zgomot, prin combinarea eşantioanelor cu formula (3.2) şi apoi estimând noua distribuţie [74]. Această metodă, însă, nu se poate aplica sistemelor mari. O aproximaţie mai rapidă şi mai simplă este neglijarea varianţei zgomotului. Această este aproximaţia *log-add* şi este definită în formulele (3.17) şi (3.18):

$$\mu_y = \mu_x + f(\mu_x, \mu_n, \mu_h) \tag{3.17}$$

$$\Sigma_y = \Sigma_x \tag{3.18}$$

unde *f* este dată de formula (3.2). Această aproximaţie este mai rapidă decât PMC standard pentru că nu este necesară trecerea matricii de covarianţă din domeniul cepstral în cel spectral liniar şi operaţia inversă.

O metodă alternativă pentru compensarea modelelor ar fi utilizarea aproximării Taylor pentru semnalul de observaţie, descrisă în [2]. Se face o dezvoltare în serie Taylor şi punctele în care se face dezvoltarea sunt mediile distribuţiilor semnalului vocal, ale zgomotului şi ale zgomotului convolutiv. Astfel, rezultă media pentru semnalul afectat de zgomot:

$$\mu_y = \varepsilon\{\mu_x + f(\mu_x, \mu_n, \mu_h,) + (x^s - \mu_x)\frac{\partial f}{\partial x^s} + (n^s - \mu_n)\frac{\partial f}{\partial n^s} + (h^s - \mu_h)\frac{\partial f}{\partial h^s}\} \tag{3.19}$$

Derivatele parţiale pot fi exprimate funcţie de derivatele parţiale ale lui y^s în raport cu x^s, n^s şi h în punctul μ_x, μ_n, μ_h. Acestea sunt de forma:

$$\frac{\partial y^s}{\partial x^s} = \frac{\partial y^s}{\partial h} = A \tag{3.20}$$

$$\frac{\partial y^s}{\partial n^s} = I - A \tag{3.21}$$

unde $A=CFC^{-1}$ şi F este o matrice diagonală a cărei matrice inversă are elementele diagonale date de $(1+\exp(C^{-1}(\mu_n - \mu_x - \mu_h)))$. Rezultă că mediile şi varianţele ale semnalului afectat de zgomot sunt date de formulele (3.22) şi (3.23):

$$\mu_y = \mu_x + f(\mu_x, \mu_n, \mu_h) \tag{3.22}$$

$$\Sigma_y = A\Sigma_y A^T + A\Sigma_h A^T + (I-A)\Sigma_n (I-A)^T \tag{3.23}$$

Se poate arăta empiric că aproximaţia în serie Taylor este în general mai acurată decât cea PMC log-normală. Trebuie observat că media este compensată la fel ca în aproximaţia PMC *log-add*.

O altă problemă care apare la metodele de compensare a modelelor, este compensarea parametrilor delta, adică derivatele de ordin I şi de ordin II [71]. În acest caz se foloseşte des aproximaţia continuă în timp [66]. La această tehnică, parametrii dinamici, care sunt un estimat al gradientului, sunt aproximaţi cu derivata în timp:

$$\Delta y_t^s = \frac{\sum_{i=1}^{n} w_i (y_{t+1}^s + y_{t-1}^s)}{2\sum_{i=1}^{n} w_i^2} \approx \frac{\partial y_t^s}{\partial t} \tag{3.24}$$

De exemplu, se poate arăta că media parametrilor delta $\mu_{\Delta y}$ poate fi exprimată sub forma:

$$\mu_{\Delta y} = A\mu_{\Delta x} \tag{3.25}$$

Varianţele şi derivatele de ordin doi pot fi şi ele compensate astfel.

Metodele descrise de compensare a modelelor presupun că parametrii modelului de zgomot μ_n, Σ_n, μ_h sunt ştiuți.

3.4.3. Metode bazate pe incertitudine

Metodele de compensare a parametrilor de voce sunt eficiente din punct de vedere al complexităţii de calcul, dar ele sunt limitate la nişte modele acustice simple. Pe de altă parte, metodele de compensare a modelelor pot profita de utilizarea unor modele detaliate dar aduc dezavantaje din punct de vedere al complexităţii de calcul. Metodele bazate pe incertitudine oferă un compromis între aceste două neajunsuri.

Efectele zgomotului aditiv pot fi reprezentate printr-o rețea Bayes dinamică (DBN, din engleză Dynamic Bayes Network) cum este arătat în Fig. 3.3. În acest caz, probabilitatea de ieşire a unei stări pentru un vector de parametri de voce afectat de zgomot poate fi scris:

$$p(y_t \mid \theta_t; \lambda, \breve{\lambda}) = \int_{x_t} p(y_t \mid x_t; \breve{\lambda}) p(x_t \mid \theta_t; \lambda) dx_t \tag{3.26}$$

unde

$$p(y_t \mid x_t; \breve{\lambda}) = \int_{n_t} p(y_t \mid x_t, n_t)\, p(n_t; \breve{\lambda}) dn_t \tag{3.27}$$

iar λ este modelul acustic curat şi λ' este modelul pentru zgomot.

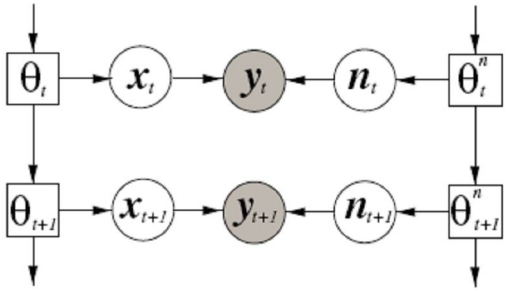

Fig. 3.3. DBN al unui model combinat voce-zgomot.

Probabilitatea condiţionată din formula (3.27) poate fi privită ca o măsură a incertitudinii care rezidă în estimarea lui x_t din y_t. La limită, această probabilitate poate fi privită ca un comutator care dacă este unu atunci x_t este cunoscut cu certitudine, altfel acest parametru poate fi considerat necunoscut sau *lipsă*. Pe această consideraţie se bazează *metoda parametrilor lipsă*. O metodă alternativă ar fi utilizarea probabilităţii $p(y_t|x_t,\lambda')$ ca parametru. Ea poate să intre direct în decodor ca o măsură a incertitudinii a elementelor vectorului de parametrii de voce. Această metodă se numeşte *decodarea sub incertitudine.*

3.4.3.1. *Metoda „parametrilor lipsă"*

Metoda parametrilor lipsă a fost dezvoltată din analizele făcute sistemului auditiv şi plecând de la premisa că oamenii recunosc vocea înnecată în zgomot prin identificarea unor „zone" puternice în planul timp-frecvenţă şi ignorând restul de semnal. În domeniul frecvenţă, o secvenţă de vectori de observaţie $Y_{1:T}$ poate fi privită ca o spectogramă indexată în dimensiunea timp i=1,...,T iar în dimensiunea frecvenţă k=1,...,K. Fiecare element y_{tk} are asociat un bit care determină dacă acel element este de încredere sau nu. În aceeaşi idee, aceşti biţi formează o mască care determină care zone ale spectogramei sunt afectate de zgomot şi acele zone vor fi clasificate ca fiind lipsă. În aceste condiţii, recunoaşterea de voce implică două probleme: găsirea măştii de biţi şi calculul probabilităţii folosind acele măşti [131], [171].

Problema alcătuirii măştii este cel mai dificil aspect al metodei „parametrilor lipsă". O soluţie simplă ar fi estimarea RSZ şi aplicarea unor praguri. Această soluţie poate fi îmbunătăţită prin îmbinarea ei cu criterii bazate pe percepţie cum ar fi informaţia armonicii de vârf (în engleză Harmonic Peak Information), rapoartele de energie, etc. [10]. Performanţele cele mai bune le oferă antrenarea unui clasificator Bayesian cu semnal curat căruia i s-a adăugat zgomot artificial [183].

Odată stabilită masca, se calculează apoi probabilităţile, fie prin înlocuirea, fie prin eliminarea parametrilor de voce de neîncredere. În primul caz, determinarea parametrilor

„lipsă” se face printr-un estimat MAP (din engleză Maximum A posteriori Probability) presupunând că vectorul de parametri a fost generat de un GMM de semnal curat [171].

Rezultatele experimentale arată că eliminarea parametrilor dă rezultate mai bune decât înlocuirea. Cu toate acestea, avantajul înlocuirii este posibilitatea de a trece în domeniul cepstral după reconstruirea vectorilor de parametrii de voce.

3.4.3.2. *Decodare sub incertitudine*

Scopul decodării sub incertitudine este exprimarea probabilităţii condiţionate din formula (3.27) sub forma unui parametru care poate fi pasat decodorului pentru a calcula probabilitatea de ieşire a unei stări (formula (3.26)) în mod eficient. O decizie importantă constă în forma care i se dă probabilităţii condiţionate. În formularea originală, distribuţia condiţionată depinde de regiunea din spaţiul acustic. O metodă alternativă ar fi ca distribuţia condiţionată să depindă de componenta gaussiană, similar cu regresiile liniare de adaptare (vezi capitolul 2.5.).

O soluţie pentru aproximarea probabilităţii condiţionate este utilizarea unei GMM cu N componente pentru modelarea acustică. Funcţie de forma incertitudinii utilizate, acest GMM poate fi folosit în mai multe moduri pentru a modela distribuţia condiţionată, $p(y_t|x_t;\lambda')$ [59], [126]. De exemplu în [126]:

$$p(y_t \mid x_t;\breve{\lambda}) \approx \sum_{n=1}^{N} P(n \mid x_t;\breve{\lambda})\, p(y_t \mid x_t, n;\breve{\lambda}) \tag{3.28}$$

Dacă componenta este aproximată a posteriori:

$$P(n \mid x_t;\breve{\lambda}) \approx P(n \mid y_t;\breve{\lambda}) \tag{3.29}$$

atunci este posibil să construim un GMM bazată pe vectorul de observaţie alterat, y_t, şi nu pe vectorul curat x_t pe care nu-l ştim. Dacă distribuţia condiţionată este aproximată prin alegerea celor mai probabile componente n^*, atunci:

$$p(y_t \mid x_t, n;\breve{\lambda}) \approx p(y_t \mid x_t, \breve{c}_{n^*};\breve{\lambda}) \tag{3.30}$$

Probabilitatea condiţionată din formula (3.27) se reduce la o singură gaussiană şi, fiindcă componentele modelului acustic sunt şi ele gaussiene, probabilitatea de ieşire a stării din formula (3.26) devine convoluţia a două gaussiene care este şi ea o gaussiană. Se poate arăta că în acest caz formula devine:

$$p(y_t \mid \theta_t;\lambda,\breve{\lambda}) \approx \sum_{m\in\theta_t} c_m N(A^{(n^*)} y_t + b^{(n^*)};\mu^{(m)},\Sigma^{(m)} + \Sigma_b^{(n^*)}) \tag{3.31}$$

Aşadar, operaţia de transformare liniară a vectorului de observaţie este realizată în dispozitivul mobil în timpul rulării. Costul adiţional este introducerea unui offset care i se pasează decodorului şi care se adaugă la varianţele modelelor. Aceeaşi formă a compensaţiei se obţine folosind metoda SPLICE cu incertitudine [59], deşi modul în care sunt calculaţi parametrii de compensare $A^{(n)}$, $b^{(n)}$ şi $\Sigma^{(n)}$ diferă. Un avantaj al acestei scheme este faptul că costul calculării parametrilor de compensare depinde de numărul de componente folosite pentru modelarea spaţiului de parametri de voce. Comparativ cu tehnicile de compensare a

modelelor (PMC şi VTS), în acest caz costul de compensaţie depinde de numpărul de componente în decodor. Această decuplare permite un nivel ridicat de flexibilitate în etapa proiectării.

Decodarea sub incertitudine bazată pe GMM aduce rezultate bune. Cu toate acestea, în condiţii de RSZ mic poate face ca toate modelele compensate să fie asemănătoare, reducând, astfel, capacitatea de discriminare a decodoroului [127]. Pentru a preveni această problemă, distribuţia condiţionată poate fi legată de componentele modelelor decodorului. Aceasta seamănă cu clasele de regresie din schemele de adaptare (de ex. MLLR). În acest caz:

$$p(y_t \mid \theta_t; \lambda, \breve{\lambda}) \approx \sum_{m \in \theta_t} \mid A^{(r_m)} \mid c_m N(A^{(r_m)} y_t + b^{(r_m)}; \mu^{(m)}, \Sigma^{(m)} + \Sigma_b^{(r_m)}) \tag{3.32}$$

unde r_m este clasa de regresie căreia îi aparţine m. Trebuie observat că în acest caz, transformarea jacobiană $|A^{(rm)}|$ poate varia de la o componentă la alta, faţă de de valoare fixă folosită în formula (3.32). Un aspect interesant al acestei metode este faptul că în timp ce numărul de clase de regresie tinde către numărul de componente ale decodorului, performanţa tinde către cea a metodelor bazate pe compensaţia modelelor.

3.4.4. Tipuri de zgomot

Există multe studii despre efectul zgomotului asupra calităţii RLV. Aceasta este încă o problemă deschisă şi motivul principal este variaţia zgomotului. Există multe tipuri de zgomot şi nu toate au acelaşi impact asupra acurateţei RLV. Soluţiile oferite pentru creşterea robusteţii SRLV faţă de zgomot au şi ele eficienţă diferită de la un tip de zgomot la altul. În acest scop, există multe baze de date standard cu înregistrări afectate de tipuri diferite de zgomot. Una dintre ele este baza de date Aurora definită de către ETSI [96]. Astfel impactul optimizărilor diferite este uşor măsurabil. Tipurile de zgomot definite în acest standard sunt cele de fond. Astfel avem zgomot în condiţii de metrou, maşină, restaurant, aeroport, gară, stradă, etc. Printre cele mai grele situaţii, este cea în care vorbesc mai mulţi în acelaşi timp şi sistemul trebuie să se focalizeze doar pe unul dintre ei (în engleză aceasta se numeşte Cocktail Party Effect sau Babble Noise). Un exemplu de cum afeactează fiecare tip de zgomot acurateţea RLV este dat în Tabelul 3.1. [49]:

Tabelul 3.1. Variaţia ratei de recunoaştere cu funcţie de tipul zgomotului [49]

Rata de recunoaştere [%]										
	Metro	*Babble*	Maşină	Expoziţie	Media	Restaurant	Stradă	Aeroport	Gară	Media
RSZ	98.86	98.61	99.11	99.07	98.91	98.86	98.61	99.11	99.07	98.91
40dB	98.37	98.28	98.42	98.15	98.31	97.61	98.04	98.36	98.27	98.07
35dB	97.94	97.25	98.15	97.35	97.67	96.41	97.04	97.14	97.47	97.02
30dB	95.36	94.8	96.33	94.23	95.18	93.18	94.68	94.66	95.37	94.47
25dB	90.64	87.64	90.75	86.92	88.99	84.22	86.94	87.86	88.34	86.84
20dB	74.58	66.2	73.31	69.7	70.95	62.24	65.51	72.68	70.87	67.83
15dB	39.18	31.32	30.69	35.59	34.2	29.72	32.2	36.81	33.88	33.15
Media	91.38	88.83	91.39	89.27	90.22	86.73	88.44	90.14	90.06	88.84

Din aceste date se observă că rezultatele cele mai slabe sunt obținute în cazul în care apare efectul *Babble* (*Babble*, Expoziție, Restaurant). Se mai observă că pentru un raport semnal zgomot (RSZ) mai mic decât 25 dB rata de recunoanștere începe să scadă brusc.

3.5. ADAPTAREA MODELELOR

Modelele acustice sunt antrenate pentru a acoperi o gama largă de variație a parametrilor vocali. Cu toate acestea, există tot timpul nepotriviri între condițiile acustice la antrenare și cele la recunoaștere. Aceste nepotriviri derivă din variația vorbitorului și mai în general de contextul în care au fost pronunțate cuvintele, dar în această categorie intră și nepotrivirile cauzate de microfonul diferit folosit, canalul de telecomunicație folosit și zgomotul de ambianță. Este imposibil de a antrena sistemul de fiecare dată când apar condiții noi de aceea trebuie găsită o metodă care să adapteze modelele existente la condițiile noi folosind o cantitate mică de date, ceea ce constituie și scopul adaptării. Una dintre situațiile tipice este variația vorbitorului. Un sistem dependent de vorbitor oferă o rată de recunoaștere mai mare față de un sistem independent de vorbitor [102]. Acest rezultat este confirmat și în această lucrare (vezi capitolul 5.1.2.). Însă, adaptarea vorbitorului poate da același rezultat cu o cantitate mult mai mică pentru un vorbitor particular. În aceste condiții, strategia de antrenare devine clară. Se antrenează niște modele în anumite condiții (cât mai variate). Urmând apoi adaptarea lor la câteva condiții particulare folosind o bază de date mică din care se extrag caracteristicile noilor condiții.

Adaptarea se poate face prin diferite metode. Una dintre ele este adaptarea continuă a modelelor în timpul recunoașterii, ceea ce înseamnă ca fiecare frază decodată de către sistem să fie folosită pentru adaptarea modelelor. Această metodă se numește adaptare nesupraveghată. Avantajul principal al acestei metode este capabilitatea ei de a prinde nepotriviri nestaționare. În schimb, dacă rezultatele recunoașterii sunt incorecte sistemul se îndrumă greșit. O alternativă a metodei nesupraveghată este cea supraveghată în care transcrierea unei porțiuni ale semnalului vocal se știe dinainte. Astfel, utilizatorului i se poate cere să pronunțe o frază anume, eventual, aleasă strategic ca să cuprindă acele foneme cu care se întâlnesc cel mai des probleme.

3.5.1. Regresia liniară de probabilitate maximă.

La un SRLV bazat pe HMM continue, probabilitatea de ieșire a unei stări este modelată printr-o mixtură gaussiană. Parametrii acestei distribuții (adică vectorul de medii și matricea de covarianță a fiecărei componente ale mixturii) sunt cei care trebuie adaptați la noile condiții. Metoda regresiei liniare de probabilitate maximă (MLLR, din engleză Maximum Likelihood Linear Regression) calculează un set de transformări care reduce nepotrivirea între modelele inițiale și datele de adaptare.

Transformările se consideră a fi liniare cu forma generală:

$$\tilde{\mu}_m = A_m \mu_m + b_m \tag{3.33}$$

unde $\tilde{\mu}_m$ reprezintă noul vector de medii al gaussienei *m*. El este privit ca fiind o rotaţie şi o translaţie a vectorului original de medii μ de dimensiune *n*. A_m este matricea de rotaţie de dimensiune *n* x *n* şi b_m este vectorul de translaţie. De obicei transfromarea se scrie într-o formă mai compactă:

$$\tilde{\xi} = W_m \xi_m \tag{3.34}$$

unde ξ este vectorul de medii extins:

$$\xi_m = [1 \quad \mu_{m1} \quad \mu_{m2} \quad \ldots \quad \mu_{mn}]^T \tag{3.35}$$

şi W_m este o matrice de transformare de dimensiune *n* x (*n* + 1) care poate fi scrisă sub forma:

$$W_m = [b_m \quad A_m] \tag{3.36}$$

Elementele matricii de transformare Wm sunt estimaţi prin maximizarea probabilităţii logaritmice pentru datele de adaptare.

Numărul de transformării se alege ca un compromis între numărul de componente gaussiene şi cantitatea de date disponibile pentru adaptare. Dacă există suficiente date de adaptare, se poate construi o transformare pentru fiecare componentă gaussiană din setul de modele. Trebuie amintit că fiecare stare a unui model conţine câteva gaussiene, de aceea numărul total de componente gaussiene este foarte mare (de ordinul miilor) şi cantitatea de date de adaptare necesară este foarte mare. Din acest motiv, gaussienele se pot grupa în clase mai mari de foneme (şi anume vocale, fricative, etc.) şi pentru fiecare clasă se foloseşte o singură transformare, obţinând o estimare robustă a parametrilor. În loc să folosim câteva clase predefinite, se obţine estimări robuste prin *arborii claselor de regresie* [123].

Arborele claselor de regresie sunt construite prin gruparea gaussienelor care sunt apropiate în spaţiul acustic. Astfel, distribuţiile similare suferă aceeaşi transformare. Fig. 3.4. arată un arbore de regresie tipic cu patru frunze. Un arbore de regresie binară poate fi construit prin utilizarea unui algoritm *top-down* cu distanţă euclidiană [75]. Pentru început, toate gaussienele din setul de modele sunt asignate nodului rădăcină al arborelui. La un nivel dat al arborelui, fiecare nod este împărţit în două noduri fii până când se obţine un număr dorit de noduri frunză. Gaussienele sunt împărţite între noduri astfel încât suma distanţelor euclidiene din centroidul nodului sa fie minimă în fiecare nod. Nodurile frunză reprezintă clasele de regresie şi fiecare gaussiană din setul de modele aparţine uneia dintre aceste clase.

În timpul adaptării, datele (vectorii de observaţie) sunt aliniate cu setul de modele şi se numără corespondenţele (adică numărul de vectori de observaţie care aparţin unei gaussiene date) pentru fiecare clasă. Dacă clasa de regresie are date suficiente se estimează matricea de transformare. Dacă nu există date suficiente pentru un nod dat, atunci observaţiile nodurilor fii sunt grupate la nodul părinte. Acest proces este repetat până când există date suficiente pentru estimarea matricii de transformare. În exemplul din figura, nodurile 6 şi 7 au date insuficiente, de aceea observaţiile lor sunt grupate la nodul părinte 3 şi o matrice de transformare comună este estimată pentru aceste două noduri. În acest caz, se calculează doar 3 matrici de transformare pentru nodurile 4, 5 şi 3. Apoi, toate gaussienele din aceeaşi clasă sunt transformate folosind aceeaşi matrice.

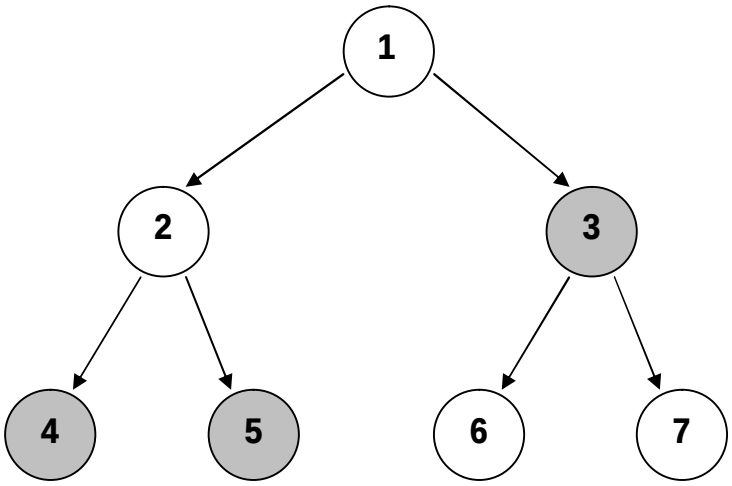

Fig. 3.4. Arborele claselor de regresie

3.6. LIMITĂRI CARE APAR ÎN REŢELE DE TELECOMUNICAŢII DE DATE

Problemele care apar în reţelele de telecomunicaţii diferă în primul rand de tipul de serviciu care i se oferă clientului. Putem distinge între reţele cu comutare de circuite şi reţele cu comutare de pachete. În primul caz se stabileşte un canal de comunicaţie între două terminale pe toată durate de comunicaţie şi aceasta va introduce o întârziere constantă şi o rată de date constantă. La comutarea de pachete, pachetele împărtăşesc aceeaşi bandă de transmisie şi cu alt trafic de date şi funcţie de încărcarea reţelei vom avea efecte diferite ale perturbaţiilor (întârziere, pierdere de pachete, etc.). Astăzi pe aceste reţele s-au implementat şi aplicaţii de timp real (de exemplu telefonie pe Internet, VoIP). Datorită flexibilităţii pe care o oferă, dar şi a îmbunătăţirilor în cadrul calităţii serviciului (QoS), reţelele cu comutare de pachet şi în mod deosebit reţelele IP domină în acest domeniu. În aceste condiţii studiul factorilor adverşi în aceste reţele în vederea eliminării lor şi ale efectelor acestora devine important. Numeroase studii s-au facut în acest sens iar ariile atinse sunt codec-urile utilizate, pierderea de pachete, zgomotul, etc.

3.6.1. Pierderea de pachete

În aplicații de timp real, pierderea de pachete nu se poate remedia prin retransmisie, pentru că această soluție ar crește întârzierea pachetelor prin rețea. De aceea este nevoie să se analizeze efectul pierderii pachetelor asupra calității semanlului vocal și de consecință asupra ratei de recunoaștere. Apoi se pot încerca metode de remediere cum ar fi controlul ratei (dacă codecul folosit permite așa ceva), corecția de erori (FEC, din engleză Forward Error Correction), întrețeserea (în engleză Interleaving) și mascarea erorilor (în engleză Error Concealment).

Fiecare canal de comunicație are caracteristicile sale și distribuția pierderii de pachete diferă de la un canal la altul. Această distribuție este importantă pentru că experimentele au arătat că distribuții diferite duc la rate diferite de recunoaștere. După cum se va vedea mai jos pentru același procent de pierdere de pachete, pierderea pachetelor în rafală aduce o scădere mai mare a ratei de recunoaștere decât pierderea aleatoare a pachetelor.

Pentru sistemele GSM s-au făcut numeroase studii pentru modelarea pierderii de pachete și unul dintre ele este GSM EP (modele de erori, din engleză Error Patterns, [66]). În Tabelul 3.2. sunt prezentate rezultatele obținute folosind standardul ETSI DSR FE [61], folosind un sistem NSR care folosește codecul GSM EFR, iar transmisia datelor este simulată prin modelul de erori EP [165]. Ca și bază de date pentru recunoaștere s-a folosit AURORA-2 [96]. EFR obține rezultate mai slabe atât în condiții fără pierderi de pachete (datorită codării) cât și în condiții cu pierderi de pachete (datorită nivelului scăzut de protecție pe care îl oferă în acest caz).

Tabelul 3.2. Comportarea codecului folosit în ETSI DSR și EFR pentru EP diferite [165]

Sistemul de recunoaștere	Modele de pierderi de pachete			
	Fără pierderi	EP1	EP2	EP3
DSR	99.04	99.04	98.95	93.41
EFR	98.70	98.44	96.91	84.48

Pierderea de pachete în rețelele IP a fost analizată de către Bolot [21], [22], și el a ajuns în următoarele concluzii:

a) Pachetele tind să vină în rafală (fenomenul de compresie a fluxului)

b) Întârzierile datorită așteptării în cozi fluctuează rapid (cozile de asteptare pot fi atât în stația sursă cât și pe parcurs în rutere, etc.).

c) Pierderea de pachete are o natură aleatoare cu excepția cazului în care fluxul particular asupra căruia se fac măsurătorile, ocupă o parte importantă a capacității de transmisie a canalului.

d) Pachetele pierdute tind să fie în rafală. Dacă un pachet s-a pierdut din cauza congestiei care a avut loc în una dintre cozile pe parcursul rutei, atunci probabilitatea ca următorul pachet să fie pierdut este mare.

Pentru a simula acest comportament, și anume pierderea de pachete în condiții de laborator, se folosesc diferite modele. Modelele folosite în acest scop sunt de obicei niște lanțuri Markov cu două sau mai multe stări. Cel mai simplu dintre ele este modelul Gilbert-Elliot [89], [60], ilustrat în Fig 3.5. Starea 0 reprezintă cazul în care nu se pierd pachete iar starea 1 cazul în care se pierd. Acest model acoperă atât pierderea aleatoare a pachetelor cât și pierderea lor în rafală. Ținând cont că:

$$\begin{cases} P(0\,|\,0) + P(1\,|\,0) = 1 \\ P(0\,|\,1) + P(1\,|\,1) = 1 \end{cases} \tag{3.37}$$

și notând P(1|0) cu *p* și P(0|1) cu *q*, modelul este parametrizat și depinde doar de doi parametrii. Rata de pierderi de pachete este mai mare cu cât *p* este mai mare, iar dimensiunea rafalei este mai mare cu cât *q* este mai mic.

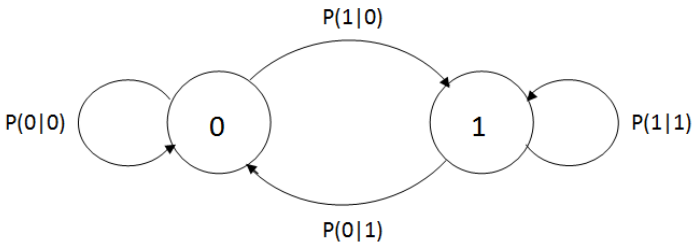

Fig. 3.5. Modelul Gilbert-Elliot

Cu acest model este ușor de simulat un canal cu anumite caracteristici. Mai întâi se fac măsurători pe canalul de comunicație care urmează a fi modelat, apoi se estimează parametrii *p* și *q*. Studiile arată că pentru același procent de pierdere de pachete, pierderea de pachete în rafală (pentru un *q* mic) duce la o creștere mai mare a ratei de erori la recunoaștere decât pierderea aleatoare a pachetelor (pentru un *p* mare). Un asemenea studiu e facut de către [165], iar rezultatele sunt prezentate în Tabelul 3.3.

Tabelul 3.3. Rata de recunoaștere funcție de tipul de pierderi de pachete

Rata de pierderi [%]	Numarul de pachete pierdute conzecutive				
	1	2	4	8	16
10	98.98	98.61	96.50	93.42	90.77
20	98.96	98.08	94.02	87.28	83.03
30	98.88	97.56	90.92	81.43	74.87
40	98.92	96.81	88.18	76.61	66.89
50	98.90	96.21	84.56	70.36	59.63

Acest model caracterizează bine canalul de comunicație pentru pierderi în rafală scurte (până în 3 pachete) și nu modelează bine rafale mai lungi [110], [146]. Lipsa de acuratețe în acest caz se datorează naturii simple ale modelului (cu doar două stări) și care are doar doi parametri. Pentru a obține rezultate mai bune este necesar un model mai complex.

O variantă extinsă a modelului Gilbert-Elliot este propusă de către Sanneck și Carle [178]. Modelul este prezentat în Fig. 3.6. Modelul are n+1 stări și doar starea s_0 este considerată bună (fără pierderi), iar orice altă stare s_k (pentru k diferit de 0) reprezintă cele k pachete conzecutive care s-au pierdut. Dacă numărul de pachete pierdute consecutive este mai mare decât n atunci există o tranziție în aceeași stare s_n cu probabilitatea a_{nn}.

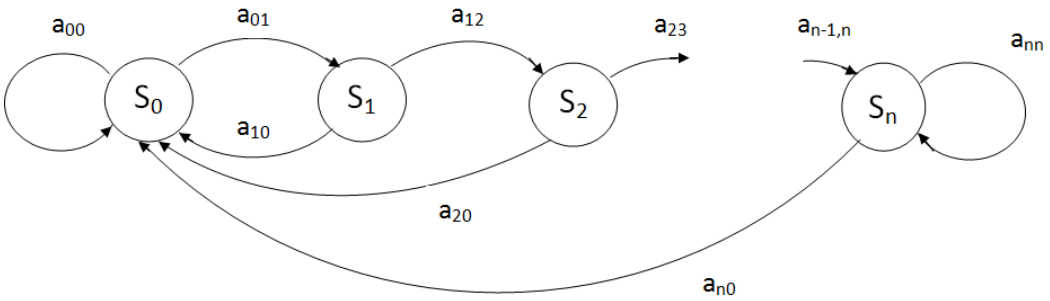

Fig. 3.6. Modelul Gilbert-Elliot extins

Acest model are însă un număr mare de parametrii care crește odată cu *n*. Un model mai simplu este elaborat de către ETSI [65] și este prezentat în Fig. 3.7. Practic, el conectează două modele cu câte două stări fiecare pentru perioadele fără pierderi și respectiv pentru perioadele cu pierderi. Stările din Fig. 3.7. au următoarea semnificație:

Starea 0 – Pachet recepționat corect în perioada fară pierderi

Starea 1 – Pachet recepționat corect în perioada cu pierderi

Starea 2 – Pachet pierdut în perioada cu pierderi

Starea 3 – Pachet pierdut în perioada fără pierderi

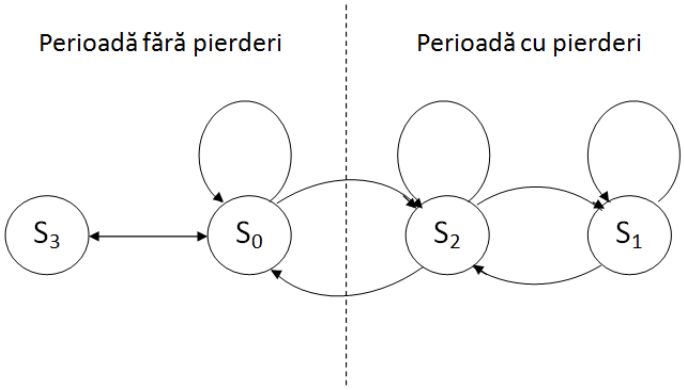

Fig. 3.7. Modelul Gilbert-Elliot îmbunătățit de către ETSI

Un model mai complex care poate modela mai multe tipare de pierdere de pachete, este modelul Markov de ordin *n* (mai mare ca 1). În acest caz, variabila aleatoare X_t depinde nu numai de cea precedentă, ci de cele *n* variabile precedente. Astfel, se va lucra cu probabilitatea de tranziție sub forma $P(X_t \mid X_{t-1}, X_{t-2},.., X_{t-n})$. Yajnik [211] a arătat că un model Markov de ordin n=6 este de ajuns pentru modela toate secvențele de pierderi pe care le foloseau.

3.6.2. Efectul codări semnalului vocal în recunoașterea limbajului vorbit

Numărul de codecuri folosite azi în rețelele de telecomunicații este mare. Analiza efectului codării asupra calității recunoașterii de vorbire este fundamentală pentru a alegerea codecului. Insă, pentru a construi un SRLV de aplicabilitate largă nu putem impune restricția utilizării unui codec anume. Astfel, se pune problema recunoașterii de voce plecând de la un semnal vocal recepționat pe un canal de comunicație (NSR). În cazul DSR, problema este mai simplă pentru că se folosește un codor cu aceiași parametri de voce care sunt folosiți și la recunoaștere. Totuși, existența unui semnal vocal de calitate acceptabilă la recepție este o caracteristică importantă a NSR, pe care DSR nu o are. Din acest motiv, NSR devine o soluție interesantă [116].

La evaluarea unui codor se pot lua în considerare câteva aspecte cum ar fi:

- Rata de biţi
- Calitatea semnalului de voce decodat
- Costul de calcul
- Întârzierea introdusă
- Robusteţea faţă de degradările de canal

În Tabelul 3.4. [204], sunt prezentate câteva codecuri uzuale împreună cu ratele lor, timpul de pachetizare și MOS.

Tabelul 3.4. Caracteristicile unor codecuri uzuale [204]

Codor	Rata [kb/s]	Pachetizări tipice [ms]	Întârziere de pachetizare [ms]	MOS maxim obținut
G.711	64	20	1	4.41
G.726	32	20	1	4.22
G.723.1	5.3	30	37.5	3.69
G.723.1	6.3	30	37.5	3.87
G.729	8.0	20	25	4.07

Este de așteptat ca un codor, care este bun în sens MOS, să fie bun și pentru recunoaștere, dar această afirmație nu este întotdeauna adevărată pentru că codoarele sunt proiectate folosind criterii diferite față de cele folosite la recunoaștere. De aceea, are sens să studiem cum se comportă fiecare codor la recunoaștere. La o lucrare mai veche, Euler și Zinke [67] arată că există două motive pentru care un codec degredează performanțele recunoașterii. Prima și cea mai importantă cauză este degradarea care provine din compresie care duce la degradarea vocii și de consecință și a performanței SRLV. Testele au fost făcute pe o bază de date de cuvinte izolate, iar HMM-urile au fost antrenate cu semnal vocal codat cu legea A și cu o rată de 64kb/s. Parametrii modelați sunt LPC și energia împreună cu derivatele lor. În Tabelul 3.5., sunt trecute codecurile folosite și rezultatele de recunoaștere pentru fiecare codec.

Tabelul 3.5. Variația ratei de recunoaștere funcție de codec [67]

Codec	Rata [kb/s]	Rata de recunoaștere [%]
G.711 legea A	64	98.52
G.728 LD-CELP	16	97.57
GSM-FR	13	96.96
TETRA CELP	4.8	96.04

Al doilea motiv pentru reducerea performanței este faptul că sistemul acceptă voce din diferite codecuri, și atunci apare o nepotrivire între condițiile de antrenare și cele de testare (asemănătoare cu problema adaptării, vezi capitolul 2.5.). Pentru a rezolva această problemă, Euler a propus un clasificator gaussian care permite identificarea codecului aplicat, și utilizarea HMM-urilor antrenate cu acel codec [67].

În [128], s-a obținut o descreștere similară a performanței odată cu rata de transmisie caracteristică codecului. S-au ales șase codecuri care acoperă plaja 40-4.8 kb/s: 40 kb/s G.726 ADPCM, 32 kb/s G.726 ADPCM, 24 kb/s G.726 ADPCM, 16 kbps G.728 LD-CELP, 13 kb/s GSM-FR și 4.8 kb/s CELP-1016. O concluzie importantă a acestei lucrări este că rata de recunoaștere nu descrește neapărat monoton cu rata de transmisie, deoarece codoarele de voce sunt proiectate pe un criteriu perceptual și performanțele lor în recunoaștер nu pot fi predictibile. La aceeași concluzie s-a ajuns în [15], însă se mai precizează că la codoare din aceeași familie, rata de recunoaștere descrește monoton cu rata de transmisie. Rezultatele sunt prezentate în Tabelul 3.6. O altă concluzie importantă este că utilizarea parametrilor MFCC oferă rezultate mai bune decât utilizarea parametrilor LPC. La această concluzie s-a ajuns și din experimentele făcute în limba română și prezentate în capitolul 5.1.3.

Tabelul 3.6. Variația ratei de recunoaștere funție de rata de transmisie a codecului [15]

Codor	Rata de cuvinte eronate [%]
Necodat (Fs=16kHz)	7.7
MPEG Lay3 64 kb/s	7.8
MPEG Lay3 32 kb/s	7.9
MPEG Lay3 24 kb/s	8.4
MPEG Lay3 16 kb/s	14.6
MPEG Lay3 8 kb/s	66.2
MPEG Lay2 64 kb/s	7.7
MPEG Lay2 32 kb/s	7.8
MPEG Lay2 24 kb/s	29.4
MPEG Lay2 16 kb/s	41.7
MPEG Lay2 8 kb/s	93.8
MPEG Lay1 32 kb/s	27
G.711 (Fs=8kHz, 64kb/s)	8.1
G.723.1 (6.3 kb/s)	8.8

Un studiu complet despre efectul codecurilor FR (codecul GSM de rată mare din engleză Full Rate), HR (codecul GSM de rată mare din engleză Half Rate), EFR (codecul GSM îmbunătățit din engleză Enhanced Full Rate) și AMR (Adaptive Multi-Rate codec) folosite în GSM/UMTS (sistem universal de telecomunicații mobile, din engleză Universal

Mobile Telecommunication System) asupra recunoașterii de voce a fost făcut în [97]. Experimentele au fost făcute cu baza de date Aurora-2 [96], atât cu semnal curat cât și cu semnal degradat, folosind standardul ETSI FE și AFE FE (fără compresie) [61], [63]. În Tabelul 3.7., sunt prezentate rezultatele experimentelor pentru cazul în care antrenarea și testarea au fost făcute cu același codec. Bazele de date de antrenare și testare conțin atât semnal curat cât și degradat de zgomot. Valorile ratei de recunoaștere din coloana etichetată "cvasi-curată" corespund mediilor a șase experimente care folosesc semnal vocal cu RSZ între 0 și 20 dB. Trei dintre aceste experimente folosesc numai semnal curat pentru antrenare. Mediile celorlalte trei experimente (cu bază de antrenare cu zgomot) sunt trecute în coloanele etichetate "condiții multiple". Rezultatele experimentelor cu FE "cvasi-curată" nu spun nimic în legătură cu rata de transmisie, pentru că rata de recunoaștere nu scade odată cu rata de transmisie. Aceasta se întâmplă din cauza nepotrivirii a bazei de date de antrenare cu cea de testare. Nu se poate spune același lucru în cazul FE "condiții multiple". În schimb, rezultatele pentru AFE FE sunt coerente atât în cazul "cvasi-curată" cât și în cazul "condiții multiple". În alte experimente, se studiază efectul nepotrivirii codării între baza de date de antrenare și cea de testare. Concluzia în acest caz este că performanța cea mai bună se obține, în general, folosind PCM pentru antrenare. Acest rezultat, obținut și în [116], [117], contrazice ceea ce a observat Euler și Zinke [67] sau Pelaez [166]. În [116] și [117]. Comportamentul variabil este explicat ca o consecință a numărului variabil de componente de mixturi gaussiene folosite în HMM.

Tabelul 3.7. Rezultatele recunoașterii în condițiile în care se folosește același codec la antrenare și la recunoaștere [97]

Codec și rata [kb/s]	**Rata de recunoaștere [%]**			
	FE cvasi-curată	FE condiții multiple	AFE cvasi-curată	AFE condiții multiple
PCM 64	73.23	86.39	89.30	91.55
A-Law 64	70.15	85.76	88.88	91.53
FR 13	68.31	85.28	87.31	90.28
HR 5.6	66.44	82.45	81.77	87.18
EFR 12.2	71.44	86.22	88.16	90.97
AMR 4.75	70.16	84.89	84.76	88.99
AMR 5.15	71.17	84.49	84.23	89.09
AMR 5.9	69.46	85.05	85.02	89.66
AMR 7.4	67.58	85.74	85.23	90.01
AMR 10.2	68.38	85.63	86.36	90.50

Un alt studiu despre performanța recunoașterii folosind diferite codecuri GSM, G.723.1 și G.729, în condiții de zgomot se găsește în [203]. Rezultatele cele mai bune au fost obținute pentru G.729, FR și AMR pentru rate mari, deși FR nu este așa robust în condiții de zgomot. HR și G.723.1 au performanța cea mai slabă.

3.6.3 Detecția activității vocale

Detectecţia activităţii vocale (VAD, din engleză Voice Activity Detection) este o componentă importantă, folositoare în multe aplicaţii de procesare a semnalului vocal cum ar fi recunoaşterea vorbirii, transmisia discontinuă, cancelarea ecoului, etc.

În recunoaşterea vorbirii, VAD joacă un rol important în două aplicaţii principale. În primul rând, realizează estimarea parametrilor statistici ai zgomotului de fond, necesară pentru algoritmi de reducere a zgomotului. Deşi câteva tehnici sugerează estimarea continuă a zgomotului ambiental, în majoritatea cazurilor el este calculat în perioada de linişte.

În al doilea rând, VAD este folosit pentru a elimina porţiunea de linişte înainte de recunoaşterea de voce propriu-zisă. Eliminarea cadrelor de linişte din sistemul de recunoaştere reduce în mod semnificativ rata de eroare.

În cazurile în care VAD este folosit pentru eliminarea zgomotului din semnalul vocal, el devine critic pentru performanţele sistemului. Aceste tehnici calculează spectrul zgomotului în perioadele de linişte astfel încât să elimine efectul lui perturbător. Aşadar VAD are un rol sensibil în cazul în care avem medii non-staţionare, pentru că în acest caz este necesară calculara permanentă a statisticilor de voce, iar orice eroare poate afecta performanţele sistemului.

CAPITOLUL 4

BAZA DE DATE, STRATEGIA DE ANTRENARE ŞI INFRASTRUCTURA SISTEMULUI DE RECUNOAŞTERE A LIMBAJULUI VORBIT

Baza de date şi infrastructura SRLV sunt două aspecte importante care determină performanţele sistemului atât din punct de vedere al ratelor de recunoaştere cât şi al vitezei de rulare. Pentru a obţine rezultate bune în aceste direcţii este nevoie de o infrastructură care sa permită rularea corectă şi rapidă a proceselor de pregătire a datelor, de antrenare a modelelor şi de testarea lor. Acest capitol descrie implementarea unei asemenea infrastructuri şi organizarea bazei de date.

4.1. BAZA DE DATE

O bază de date tipică SRLV constă într-o mulţime de fişiere de vorbire care în majoritatea cazurilor sunt stocate în format wav. Pentru fiecare fişier de vorbire se construieşte un fişier de etichete care conţine transcrierea vorbirii şi marcajele de timp corespunzătoare. Etichetarea se poate face la nivel de fonem, de cuvânt, de frază sau de fişier (atunci când fişierul de etichete nu conţine marcajele de timp). Dictionarul fonetic este cel care completează baza de date. El conţine transcriera cuvintelor din forma literară folosită în fişierele de etichetare, în forma fonetică (aşa cum se pronunţă).

Caracteristicile bazei de date folosită într-un SRLV depind semnificativ de tipul aplicaţiei. De exemplu, pentru un sistem care recunoaşte comenzi este de ajuns o bază de date de cuvinte izolate. Dimensiunea bazei de date depinde de numărul de cuvinte care se vor a fi recunoscute şi de dependenţa de vorbitor. Bineînţeles, baza de date are caracteristicile specifice limbii din care face parte.

Fiindcă această lucrare a fost facută în cadrul unui proiect mai mare care are ca scop recunoaşterea vorbirii continue în limba română, atunci şi baza de date a fost aleasă cu acest

scop. Este important de prezentat atât caracteristicile şi criteriile alegerii bazei de date, cât şi câteva aspecte care privesc achiziţionarea ei.

4.1.1. Achiziţionarea bazei de date

O bază de date poate fi achiziţionată în urmatoarele moduri şi fiecare din ele ridică nişte probleme [174]:

1. Prin înregistrare care implică:
   - Alegerea locului de înregistrare (studio, laborator, etc.).
   - Alegerea microfonului (direcţional/omnidirecţional, funcţia de transfer pe care prezintă, zgomotul introdus, etc.).
   - Alegerea staţiei de înregistrare (placa audio folosită pentru achiziţie, amplitudinea semnalului, etc.).
2. Prin etichetarea „cărţilor audio" sau a altor materiale de vorbire care implică:
   - Aducerea materialelor la aceeaşi frecvenţă de eşantionare.
   - Fragmentarea fişierelor audio şi a fişierelor de etichete în entităţi mai mici: foneme, cuvinte sau grup de cuvinte.
   - Detectarea şi corectarea erorilor de etichetare şi de fragmentare.
3. Prin etichetarea unor fişiere audio de pe Internet sau din spectacole din radio sau TV. Acesta implică [28]:
   - Aducerea materialelor la aceeaşi frecvenţă de eşantionare.
   - Omogenizarea condiţiilor de vorbire şi catalogarea particularităţilor (prin omogenizare); aici trebuie înţeleasă alegerea unei baze de date echilibrate unde modul de vorbire nu este predominant un anume tip (de exemplu în şoaptă), sau majoritatea înregistrărilor să fie pronunţate de acelaşi vorbitor.
   - Detectarea şi corectarea erorilor de etichetare şi de fragmentare.

Primul tip de bază de date este util pentru experimentele care folosesc la găsirea configuraţiei optime sau pentru dezvoltarea noilor metode. Această bază de date este departe de realitate din punct de vedere fonetic. Vorbitorul căruia i s-a cerut să citească un text este concentrat şi mobilizat înainte de a începe să pronunţe. În lumea reală, chiar şi acelaşi vorbitor nu pronunţă la fel cuvintele. Modul cum pronunţă cuvintele depinde mult de context (de ex. dacă un dispozitiv nu raspunde la prima comandă, utilizatorul tinde să ridice tonul), de starea emoţională, etc. Pe de altă parte, prin modul în care este creată această bază de date este mai imună la erori şi controlul asupra ei este mai mare, ceea ce face ca investigarea diferitelor probleme în faza de dezvoltare şi de implementare să fie mai uşoară. Un alt avantaj pe care îl prezintă este faptul că ea se poate obţine destul de uşor. Aşadar, o practică bună ar fi de a începe dezvoltarea unui sistem cu acest tip de bază de date şi atunci când sistemul devine mai robust să se folosească şi celelalte două.

Al doilea tip de baze de date este potrivit pentru vorbirea continuă. Vorbitorul pronunţă corect şi intonaţia se schimbă odată cu evoluţia materialului scris din care citeşte. Ca şi în primul caz, aici vorbitorul este mobilizat şi atent la ceea ce o să spună. Achiziţionarea ei prin simplă etichetare este destul de dificilă şi cere mult timp. Aplicarea unei metode, care

după analizarea prozodiei semnalului, poate fragmenta fişierul în grup de cuvinte sau fraze ar reduce semnificativ timpul de achiziţie [28].

Al treilea tip de baze de date este potrivit pentru recunoaşterea de vorbire spontană sau recunoaşterea de emoţii. Gradul de variabilitate este ridicat pentru că vorbitorii se află în situaţii diferite iar în unele cazuri nici nu ştiu că sunt înregistraţi, ceea ce este mai aproape de lumea reală. Achiziţionarea ei este difcilă, pentru că în unele cazuri vorbitori nu vorbesc corect gramatical ceea ce necesită o atenţie sporită din partea celui care face etichetarea [167], [92].

4.1.2. Descrierea bazei de date

Indiferent de tipul de achiziţie, baza de date trebuie să satisfacă câteva cerinţe care în cazul recunoaşterii de voce devin importante. Acelea sunt:

- Să asigure o acoperire bună şi cât se poate de uniformă a vocabularului limbii şi a unităţilor acustice importante (foneme, alofone, silabe, etc.).
- Să asigure o separare bună între foneme, adică să fie pronunţate corect cu excepţia cazului în care chiar se vrea detecția unor erori de pronunțare.
- Să nu conţină erori de transcriere.
- Înregistrările să nu conţină zgomot. În cazul în care se folosesc înregistrări cu zgomot, el trebuie să fie controlat sau ușor măsurabil.

În Tabelele 4.1., 4.2., 4.3., 4.4. şi 4.5., sunt prezentate caracteristicile bazelor de date achiziţionate pentru acest proiect. Ele sunt numerotate în ordinea achiziţionării.

Tabelul 4.1. Baza de date 1 – Vorbire spontană

Metoda de achiziție	Emisiuni radio sau TV difuzate pe Internet; Etichetarea conținutului			
Data de achiziție	Februarie 2008			
Autori	Andi Buzo, Cristina Petrea, Diana Hanes			
Caracteristici ale înregistrărilor	Limbă		Română vorbită	
	Tip		Vorbire continuă, spontană	
	Durata totală		Aproximativ 4 ore	
	Mediul de înregistrare		Studio radio/TV	
	Frecvența de eșantionare		16 kHz	
	Dimensiunea eșantionului		16b	
	Etichetare		La grup de cuvinte (o etichetă la ~60s)	
Vorbitori	Numărul de vorbitori	12	Femei	8
			Bărbați	4
	Sesiuni per vorbitor		3-20	
	Timpul între două sesiuni		Între o zi și două săptămâni	
Cuvinte	Numărul total de cuvinte		37604	
	Cuvinte distincte		8068	

Tabelul 4.2. Baza de date 2 – Vorbire continuă

Metoda de achiziție	Etichetare de cărți audio			
Data de achiziție	August 2009			
Autori	Adina Popa, Diana Uzum, Mihai Iordache, Horia Cucu, Dan Oneata, Tudor Mihailescu, Ioana Rolea			
Caracteristici ale înregistrărilor	Limbă		Română literară	
	Tip		Citire, continuă	
	Durata totală		Aproximativ 11 ore	
	Mediul de înregistrare		Studio de înregistrare	
	Frecvența de eșantionare		16 kHz	
	Dimensiunea eșantionului		16 bits	
	Etichetare	3% la nivel de cuvânt 12% la grup de cuvinte (o etichetă la ~3s) 85% la grup de cuvinte (o etichetă la ~60s)		
Vorbitori	Numărul de vorbitori	7	Femei	3
			Bărbați	4
	Sesiuni per vorbitor		-	
	Timpul între două sesiuni		-	
Cuvinte	Numărul total de cuvinte		40016	
	Cuvinte distincte		13770	

Tabelul 4.3. Baza de date 3 – Foneme izolate

Simbol	Apariții		Simbol	Apariții		Simbol	Apariții		Simbol	Apariții		Simbol	Apariții		Simbol	Apariții
@	30		f	26		i2	26		k	26		o	13		t	23
a	561		g1	38		i3	22		l	38		p	7		u	21
b	31		g2	0		i	27		m	35		r	38		v	32
d	32		g	25		j	22		n	48		s1	31		w	31
e1	26		h	26		k2	0		o1	22		s	38		y	0
e	52		i1	0		k1	33		o2	0		t1	40		z	26

Tabelul 4.4. Baza de date 4 – Cuvinte izolate

Metoda de achiziție	Înregistrare directă	
Data de achiziție	Martie 2010	
Autori	Adina Popa, Diana Uzum, Tudor Mihailescu, Ioana Rolea, Florin Baltescu	
Caracteristici ale înregistrărilor	Limbă	Română literară
	Tip	Citire de cuvinte izolate
	Durata totală	-
	Mediul de înregistrare	Laborator
	Frecvența de eșantionare	16 kHz
	Dimensiunea eșantionului	16 bits
	Etichetare	La nivel de cuvânt

Vorbitori	Numărul de vorbitori	5	Femei	3
			Bărbați	2
	Sesiuni per vorbitor		10 - 20	
	Timpul între două sesiuni		O zi	
Words	Numărul total de cuvinte		50000	
	Cuvinte distincte		10000	

Tabelul 4.5. Baza de date 5 – Cuvinte izolate (comenzi)

Metoda de achiziție	Înregistrare directă			
Data de achiziție	Martie 2010			
Vorbitori	Andreia Vlad, Florin Teodoru, Daria Ion, Daniela Milea			
Caracteristici ale înregistrărilor	Limbă		Română literară	
	Tip		Citire de cuvinte izolate	
	Durata totală		-	
	Mediul de înregistrare		Laborator	
	Frecvența de eșantionare		16 kHz	
	Dimensiunea eșantionului		16 bits	
	Etichetare		La nivel de cuvânt	
Vorbitori	Numărul de vorbitori	4	Femei	3
			Bărbați	1
	Sesiuni per vorbitor		3 -10	
	Timpul între două sesiuni		O zi	
Cuvinte	Numărul total de cuvinte		5600	
	Cuvinte distincte		4	

Pentru fiecare bază de date se ţine o evidenţă cât mai completă care conţine informaţii despre tipul de înregistrare, vorbitori, numărul de cuvinte, dimensiunea bazei de date etc. Această evidenţă este importantă pentru a da rapid o imagine de ansamblu. Astfel este mai uşor de caracterizat baza de date şi în acelaşi timp este uşor de văzut unde este nevoie de îmbunătăţire. De exemplu, dacă se observă că numărul de vorbitori bărbaţi a crescut, se poate interveni prin a adăuga nişte clipuri cu vorbitori femei. De asemenea, se încearcă să se menţină aceleaşi condiţii de înregistrare.

Bazele de date 1 şi 2 au fost construite prin etichetare manuală. Se ascultă un clip şi se scriu într-un fişier de tip text cuvintele aşa cum au fost pronunţate. Pentru o prelucrare mai uşoară diacriticile au fost înlocuite cu alte caractere (vezi capitolul 4.1.3.). Marcajele de timp lipsesc şi fişierele de etichete conţin doar succesiunea de foneme pronunţate. Clipurile au fost alese astfel incât să cuprindă subiecte din domenii diferite: sport, politică, religie, literatură, etc. Aceste baze de date conţin vorbire continuă şi spontană. Dimensiunile lor sunt relativ mici pentru că sunt greu de construit. De asemenea, numărul de erori de etichetare este mai ridicat decât la celelalte baze de date. Din acest motiv, sunt necesare iteraţii suplimentare de verificare care cresc semnificativ timpul global de obţinere a bazei de date.

Baza de date 3 conţine clipuri etichetate la nivel de fonem. Fiecare fonem apare în medie 30-40 de ori. Ea a fost construită în scopul de a obţine modele de iniţializare pentru foneme. O problemă majoră la antrenarea embedded (vezi capitolul 2.2.8.) este alinierea. Aşadar, mai ales la vorbire continuă, unde dimensiunea clipurilor este mai mare (câteva cuvinte faţă de un cuvânt la baza de date de cuvinte izolate), este mai bine de a începe cu nişte modele deja antrenate decât cu modele neiniţializate. Clipurile pentru această bază de date au fost luate în parte din [197], iar restul din clipuri din baza de date 2. Trebuie menţionat că o bază de date etichetată la nivel de fonem, teoretic, ar trebui să modeleze cel mai bine fonemele pentru că elimină problema alinierii. Însă, procesul de etichetare la nivel de fonem nu este deloc uşor. În multe cazuri, graniţa între foneme este greu de stabilit.

Baza de date 4 este obţinută prin înregistrare directă. Fiecare fişier conţine un singur cuvânt astfel fişierele de etichete au structura „sil cuvânt sil", unde „sil" este eticheta pentru modelul corespunzător liniştei. Cuvintele care constituie dicţionarul fonetic pentru această bază de date au fost alese astfel încât să acopere toate silabele limbii române. Cinci vorbitori (2 barbati şi 3 femei) au făcut înregistrările în condiţii de laborator. Această bază de date este potrivită pentru recunoaştere de cuvinte izolate. Cuvintele au fost rostite cu aceeaşi intonaţie. Experimentele pentru determinarea configuraţiei optime şi pentru optimizarea algoritmilor de antrenare şi de recunoaştere au fost făcute mai întâi pe această bază de date pentru că ea oferă control mai bun decât celelalte uşurând astfel procesul de investigaţie. În acelaşi timp, dicţionarul având o dimensiune relativ mare (10000 de cuvinte) semnificaţia testelor făcute pe această bază de date este importantă.

Baza de date 5 a fost folosită pentru un sistem de recunoaştere de comenzi puţine la număr (doar 4). Ea este menţionată cu scopul de a pune în evidenţă cum variază caracteristicile bazei de date funcţie aplicaţie.

4.1.3. Analiza fonetică a bazei de date

La modelarea unităţilor de vorbire (foneme, alofone, silabe, etc.) contează mult numărul (distribuţia) apariţiilor unităţilor de vorbire. O bază de date folosită în scopul recunoaşterii vorbirii este mai bună dacă:

- Are cât mai multe apariţii pentru fiecare fonem.
- Numărul de apariţii să fie aproximativ acelaşi pentru toate fonemele. Cum acest lucru este greu de obţinut şi ineficient din punct de vedere a obţinerii bazei de date atunci se vrea ca numărul de apariţii să fie proporţional cu probabilitatea de apariţie a respectivului fonem în limba aleasă.
- Fonemele apar în contexte cât mai variate. Studiile arată că pronunţia şi parametrii vocali ai unui fonem diferă funcţie de fonemele vecine.

Fonemele folosite în acest proiect sunt prezentate în Tabelul 4.6. Prima coloană conţine simbolurile din standardul IPA (alfabetul fonetic internaţional, în engleză International Phonetic Alphabet), iar a doua coloană conţine simbolurile pentru foneme folosite în acest proiect. Nu s-au folosit direct simbolurile IPA din cauză că multe dintre ele nu fac parte din codul ASCII, iar toate utilitarele folosite lucrează cu codul ASCII (vezi capitolul 4.3. cu descrierea infrastructurii). Simbolurile SAMPA nu au fost folosite pentru că ele conţin caractere majuscule, iar la prelucrarea datelor este mai uşor de a lucra cu caractere

minuscule. S-a făcut o distincție pentru vocalele care fac parte din diftong și vocalele care apar singure în silabe. De exemplu, "o" din diftong se notează cu "o1". Aceasta înseamnă că vom antrena două modele diferite pentru „o" și „o1". Aceste cazuri pot fi tratate diferit, de exemplu, se poate construi un model pentru întregul diftong (de ex. "oa"), în loc de câte un model pentru fiecare fonem în parte, însă, trebuie testat care din cele două abordări dă rezultate mai bune.

Tabelul 4.6. Simbolurile utilizate pentru fonemele limbii române

Simbol IPA	Simbol folosit		Simbol IPA	Simbol folosit		Simbol IPA	Simbol folosit
A	a		o̯	o1		f	f
ə	@		W	w		v	v
E	e		C	k2		h	h
I	i		b	b		ʒ	j
ɨ	i1		p	p		ʃ	s1
ɨ̯	i2		k	k		l	l
O	o		ʧ	k1		m	m
U	u		g	g		n	n
Y	y		ʤ	g1		s	s
Ø	o2		ɟ	g2		z	z
e̯	e1		d	d		r	r
J	i3		t	t		ʦ	t1

O analiză cantitativă a aparițiilor fonemelor s-a făcut prin comparația bazelor de date 1 și 2. Ele au aproximativ aceeași dimensiune de aceea dacă considerăm baza de date 1 ca fiind baza de date inițială, iar uniunea celor două baze de date ca fiind baza de date extinsă, atunci putem analiza cum se modifică distribuțiile fonemelor prin dublarea bazei de date. Tabelul 4.7. arată cum se modifică numărul de apariții a alofonelor la extinderea bazei de date [169], [168].

Tabelul 4.7. Aparițiile pentru alofone

	Ore de voribire	Nr. total de alofone	Nr. de alofone dinstincte
Baza de date inițială	4	183,007	5,191
Baza de date extinsă	9	352,460	6,525

După cum se observă, la creşterea bazei de date apar cuvinte noi, în consecinţă şi alofone noi. La dublarea bazei de date apar aproximativ 1500 de alofone noi, adică numărul de alofone creşte cu 21%. Pentru a vedea dacă raportul alofonelor s-a modificat după mărirea bazei de date am făcut următorul studiu: am definit o măsură numită *pondere a alofonului* care se măsoară ca raportul între numărul de apariţii ale unui alofon şi numărul total de apariţii ale alofonelor şi se exprimă în procent. S-au luat 100 de alofone cu cele mai multe

apariţii din ambele baze de date care se măsoară. Pentru fiecare dintre aceste alofone s-a calculat *ponderea*. În medie ponderea s-a modifcat cu 23%, iar deviaţia standard a acestei abateri este de 28%. Aceste cifre sunt foarte mari; dacă avem o bază de date mare, la creşterea ei, *ponderile* nu se modifică cu atât de mult. Se poate spune că o bază de date reprezintă proporţiile unei limbi atunci când distribuţia cuvintelor/fonemelor/ alofonelor nu se modifică la mărirea ei [174]. Putem spune că baza de date pe care o avem nu *reprezintă* în mod fidel distribuţia alofonelor ale limbii române.

Mărirea bazei de date este un proces continuu. Oricât de puternic ar fi un algoritm de antrenare, o bază de date mai mare o să aducă rezultate mai bune. O dată ce rata de recunoaştere creşte la o valoare rezonabilă, atunci putem face etichetări de înregistrări mai lungi (ceea ce este mai uşor de făcut).

Tabelele cu histograme (Tabelele 4.8. şi 4.9.) arată că numărul de apariţii este diferit pentru diferite foneme/alofone. Cum era de aşteptat, numărul de apariţii pentru foneme este în medie mult mai mare decât numărul de apariţii pentru alofone. Aceste două observaţii, impun anumite metode de antrenare cum ar fi tehnica starilor legate.

Tabelul 4.8. Histograma cu aparițiile fonemelor pentru bazele de date 1 și 2

Intervale de apariții	Nr. de foneme
> 20,000	6
[10,000 : 20,000]	9
[1,000 : 10,000]	15
< 1000	4

Tabelul 4.9. Histograma cu aparițiile alofonelor pentru bazele de date 1 și 2

Intervale de apariții	Nr. de alofone
>2,000	10
[1,000 : 2,000]	36
[500 : 1,000]	66
[100 : 500]	650
[20 : 100]	1571
< 20	4194

Tabele cu histograme s-au construit şi pentru baza de date 4 (Tabelele 4.10. şi 4.11). Observațiile privind comparația între distribuțiilor fonemelor şi ale alofonelor rămân aceleaşi. În acest caz, se vede o distribuție mai bună a alofonelor pentru că avem un număr mai mare de alofone cu cel puțin 100 de apariții. Cu toate acestea, şi în acest caz trebuie folosită tehnica stărilor legate pentru că avem multe alofone cu puține apariții (sub 100).

Tabelul 4.10. Histograma cu aparițiile fonemelor pentru baza de date 1

Intervale de apariții	Nr. de foneme
>30,000	7
[20,001:30,000]	2
[10,001:20,000]	7
[900:10,000]	18

Tabelul 4.11. Histograma cu aparițiile alofonelor pentru baza de date 1

Intervale de apariţii	Nr. de alofone
>5,000	4
[3,001:5000]	6
[1,001:3,000]	24
[501:1,000]	67
[100:500]	1043
<100	6223

4.1.4. Dicţionarul fonetic

Dicționarul fonetic are funcția de transcriere fonetică a cuvintelor. Fiecare intrare în dicționar conține cuvântul şi forma lui fonetică care indică cum este pronunțat acel cuvânt. Mai jos sunt date câteva intrări ale dicționarului folosind notațiile pentru foneme prezentate în Tabelul 4.6. În transcrierea fonetică, între foneme se lasă spațiu. De exemplu:

casă k a s @

...

geantă g1 e1 a n t @

...

4.2. STRATEGIA DE ANTRENARE

Antrenarea se va face la nivel de alofon ceea ce inseamnă că la sfârşit sistemul va recunoaşte o succesiune de alofone. Prin alofon se inţelege un fonem pentru care se precizează fonemul care îl precede şi fonemul care urmează.

Am decis să folosim HMM pentru avantejele pe care ele le prezintă şi care au fost descrise în capitolul 2.2. Antrenarea s-a făcut în trei etape:

a) Antrenarea fonemelor izolate
b) Antrenarea la nivel de fonem embedded
c) Antrenarea la nivel de alofon embedded

4.2.1. Antrenarea fonemelor izolate

Scopul acestei antrenări este obţinerea unui HMM pentru fiecare fonem în parte. După cum se va vedea mai jos, antrenarea *embedded* se face fără separatori între foneme/alofone în interiorul unui cuvânt. Aceasta este posibilă datorită algoritmului Baum-Welch. Acest algoritm însă are nevoie de o aliniere iniţială destul de bună între semnalul de voce şi transcrierea de voce corespunzătoare. Această aliniere iniţială este realizată prin antrenarea fonemelor izolate care este realizată în două etape. În prima etapă se foloseşte algoritmul de aliniere a stărilor, Viterbi (vezi capitolul 2.2.5.), care găseşte secvenţa stărilor HMM ce a

generat semnalul de voce folosit pentru antrenarea modelului. Utilitarul HTK [68] folosit este HInit iar funcţionarea acestuia este prezentat în diagrama din Fig. 4.1.:

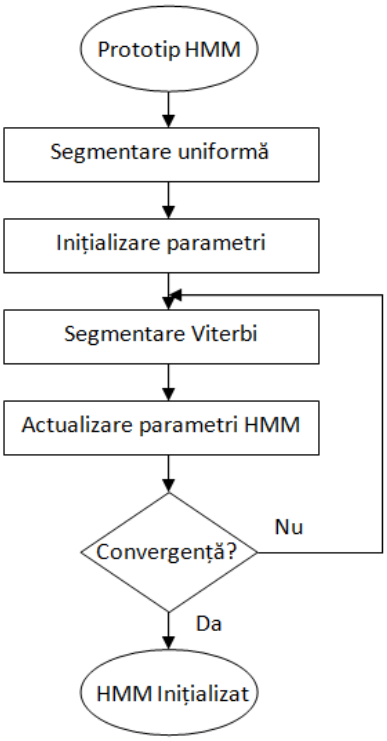

Fig. 4.1. Antrenarea fonemelor izolate pentru iniţializare

La început se face o segmentare uniformă a observaţiilor (semnalui vocal) iar parametrii HMM sunt iniţializaţi cu aceeaşi valoare şi anume 0 pentru medii iar 1 pentru varianţe. Convergenţă înseamna că de la o iteraţie la alta, probabilitatea logaritmică ca o anumită secvenţă aleasă să genereze observaţiile cu care antrenăm, să nu crească. Algoritmul Viterbi face o decizie preliminară asupra secvenţei de stări ceea ce nu reflectă tocmai realitatea. Pentru a lua o decizie mai nuanţată în a doua etapă se foloseşte algoritmul Baum-Welch (vezi capitolul 2.2.7.). Acesta este util în cazul nostru, la antrenarea bazată pe foneme/alofone, pentru că nu există o graniţă bine stabilită între fonemele din interiorul unui cuvânt. Utilitarul HTK [68] folosit este HRest iar funcţionarea acestuia (algoritmul Baum-Welch) este prezentată în capitolul 2.2.7.

Unele dintre problemele care apar la această etapă sunt:

- Etichetarea fonemelor este un proces dificil şi consumator de timp. Este greu de a stabili graniţa între două foneme într-un cuvânt.
- Pentru unele foneme avem puţine apariţii, ceea ce duce la obţinerea unor varianţe mici. Efectul varianţelor mici în recunoaştere se manifestă printr-o rată de recunoaştere bună pentru foneme rostite de acelaşi vorbitor în aceleaşi condiţii ca cele folosite pentru antrenare (ceea ce se reflectă bine în Tabelul 4.12.) şi o rată de recunoaştere scăzuta pentru foneme rostite de alţi vorbitori în alte condiţii (accent, stres, dialect, etc.). Cu alte cuvinte generalitatea modelului este scazută [28], [29].

- Pentru câteva foneme (”j” sau ”i_” de exemplu) numărul de cadre este mai mic decât numarul de stări emitente în HMM (în cazul nostru acest număr este egal cu 4). Acest număr este şi un indicator pentru a reduce numărul de stări pentru fonemul respectiv.

Tabelul 4.12. Rezultatele recunoaşterii fonemelor izolate

fonem	Apariţii	Numărul aparițiilor recunoscute corect	Rata de recunoaştere [%]
@	30	19	63
a	561	445	79
b	31	24	77
d	32	24	75
dz	38	32	84
e	52	42	80
ex	26	24	92
f	26	22	84
g	26	22	84
h	22	18	81
i	30	27	90
j	12	12	100
k	29	27	93
l	38	32	84
m	28	24	85
n	48	35	72
o	33	33	100
p	8	8	100
r	38	33	86
s	38	34	89
sh	31	26	83
t	36	33	91
ts	49	44	89
tsh	33	29	87
u	36	33	91
v	26	20	76
w	31	30	96
y	14	14	1
z	30	23	76
zh	30	25	83
total	1465	1214	82

Testele pentru recunoaştere au fost făcute folosind aceleaşi apariţii folosite la antrenare. Excluzând anumite excepţii (de exemplu: ”@”, ”j”, etc.), în general s-a obţinut o rată acceptabilă de recunoaştere.

4.2.2. Antrenarea la nivel de fonem *embedded*

Pentru recunoaştere de vorbire continuă este nevoie de o bază de date foarte mare. În aceste caz etichetarea la nivel de fonem pentru toată baza de date nu se poate aplica deoarece necesită mult timp. Aşadar, antrenarea fonemelor izolate nu este scalabilă. Pentru a trece peste această problemă se foloseşte antrenarea *embedded*. Acest mod de antrenare nu cere ca fonemele să fie separate prin marcaje de timp şi nici măcar graniţele între cuvinte nu trebuie marcate. Un fişier cu etichetări conţine doar înşiruirea fonemelor pronunţate în fişierul de voce corespunzător. Acesta este un avantaj major permiţându-ne să construim baze de date mari cu un efort redus. Antrenarea embedded funcţionează după cum urmează:

- Se consideră fiecare fişier de etichetare în parte.
- HMM-urile fonemelor din acelaşi fişier de etichetare se înlănţuiesc formând un singur HMM lung.
- Folosind algoritmul progresiv-regresiv (vezi capitolul 2.2.7.) se calculează sumele pentru mediile ponderate şi se acumulează în variabile
- După ce au fost parcurse toate fişierele se caclulează noii parametrii pentru modelele HMM pentru toate fonemele.
- Se repetă această procedură până când probabilitatea ca succesiunea de HMM-uri indicată în fişierele de etichetare să fi generat semnalul vocal să nu crească de la o iteraţie la alta.

Utilitarul HTK [68] utilizat pentru antrenarea *embedded* este HERest.

Totuşi există câteva dezavantaje ale acestei metode şi câteva probleme care pot fi îmbunătăţite:

- Acest mecanism este vulnerabil la greşeli de etichetare. Dacă la etichetarea fonemelor izolate se face o greşeala, se generează o eroare care afectează doar acel fonem. La antrenarea *embedded*, eroarea generată afectează toate fonemele din jurul acelui fonem.
- Se pune problema numărului de cuvinte incluse într-un fişier de etichetare. Folosind un număr mare de cuvinte, s-ar putea ca alinierea să nu se facă corect şi de aici să nu avem foneme bine antrenate. Pe de altă parte, existenţa fişierelor cu puţine cuvinte implică realizarea unei etichetări destul de fine ceea ce consumă timp la achiziţia bazei de date. Trebuie facut un compromis în acest sens şi, în consecinţă, am stabilit fişiere de semnal vocal de maxim un minut.
- Antrenarea embedded pentru o bază de date mare este consumătoare de timp. Algoritmul trebuie să se oprească atunci când probabilitatea de generare a secvenţei de voce nu mai creşte de la o iteraţie la alta; există însă cazuri în care probabilitatea creşte foarte încet şi atunci poate fi nevoie de multe iteraţii. Pentru a preveni acesta putem stabili un număr maxim arbitrar de iteraţii *N*. Se alege o suită de fişiere folosite pentru testare. După *N* iteraţii se testează recunoaşterea. Se mai face o iteraţie şi se testează recunoaşterea din nou. Dacă rata de recnuoaştere nu a crescut sau a crescut nesemnificativ ne oprim. Din experimente, numarul *N* nu este mai mare decât 10. Nu trebuie exagerat cu numarul de iteraţii pentru că riscăm ca varianţele să devină mici şi sistemul să piardă din generalitate. O consecinţă a acesteia este o rată de recunoaştere bună pentru clipuri folosite în antrenare şi o recunoaştere proastă pentru noi secvenţe de semnal vocal.

- Se mai poate interveni la două aspecte care ţin de modul în care funcţionează utilitarul HERest. HERest oferă posibilitatea de a lucra pe mai multe calculatoare dintr-o reţea, ceea ce va reduce mult timpul de calcul. HERest poate modifica algoritmul progresiv-regresiv astfel încât să trunchieze plaja de calcul pentru parametrii $\beta_j(t)$ şi $\alpha_j(t)$ menţionaţi la algoritmul Baum-Welch.

Într-o bază de date numărul de foneme este mai mare decât numărul de alofone ca număr mediu de apariţii pentru fiecare unitate de vorbire în parte. Dacă se începe direct cu antrenarea de alofone, alinierea şi antrenarea pentru anumite alofone care au un număr foarte mic de apariţii (în jur de 10) va fi difcilă. Mai mult, antrenarea fiind embeded acest neajuns ar fi afectat şi antrenarea alofonelor vecine. Scopul este acela de a utiliza la alofone unii parametri obţinuţi în urma antrenării de foneme. Aceşti parametri constituie matricea de tranziţie şi funcţia de ieşire pentru stările centrale ale HMM-ului. Este de aşteptat ca alofonele care au în centru acelaşi fonem să aibă pentru stările centrale aceleaşi valori de ieşire [31], [30].

4.2.3. Antrenarea la nivel de alofon *embedded*

Se pleacă de la fonemele antrenate în etapa anterioară. În prima fază, HMM-ul unui alofon se copiază de la HMM-ul fonemului central din alofon. HMM-ul alofonului avand aceeaşi matrice de tranziţie cu HMM-ul fonemului central din alofon, rezultă o aliniere bună între semnalul vocal şi fişierul de transcriere cu etichete.

După cum s-a menţionat în capitolul 4.1.3., pentru unele alofone avem destule apariţii şi pentru a trece peste aceasta problemă se foloseşte tehnica ”stărilor legate” ale alofonelor (tied-state triphones). Trebuie considerate atat avantajele cât şi dezavantajele folosirii tehnicii ”stărilor legate”; pe de o parte aceasta tehnică rezolva problema insuficienţei bazei de date, pe de altă parte reduce discriminarea ce exista între alofone cu acelaşi fonem central. De aceea strategia noastră iniţială este de a lega pentru început matricea de tranziţie şi anumite stări pentru alofonele care încep cu foneme din acelaşi grup (de exemplu nazale ”m”, ”n”, etc.). Apoi urmează a fi făcut un studiu sau să ne folosim de studii deja efectuate pentru a clasifica fonemele şi efectul pe care il au asupra fonemului pe care il precede sau il urmează astfel încât să legăm stări ale fonemelor cu acelaşi efect.

În ultima fază, se face antrenarea embedded pentru alofone în acelaşi mod ca cea realizată pentru foneme. În acest caz, vom avea mai multe modele de antrenat (aproximativ 6500 la număr) dar unele din ele sunt ”legate”. Utilitarul HTK folosit este acelaşi şi anume HERest.

4.2.4. Recunoaşterea şi testarea

Pentru recunoaştere se foloseşte algoritmul Viterbi prezentat în capitolul 2.2.5. şi implementat în utilitarul HTK HVite [68]. Cum din recunoaştere se obţine un şir de alofone, este nevoie de un dicţionar fonetic care să facă conversia în cuvinte. Acest dicţionar este acelaşi folosit şi pentru transcrierea fonetică la antrenare.

Un aspect important la testare este alegerea secvenţelor de test. Pentru recunoaştere independent de vorbitor trebuie ca înregistrările de test să nu fie făcute de aceiaşi vorbitori utilizaţi la antrenare. De obicei, din baza de date etichetată se păstrează 10% pentru testare.

Utilitarul HVite permite ca semnalul vocal să fie preluat atât de la un fişier cât şi achiziţionat în timp real. Din semnalul vocal se calculează apoi coeficienţii MFCC ca şi în cazul antrenării.

4.2.5. Soluţii la câteva probleme practice ce au aparut in timpul antrenării

4.2.5.1. *Problema unei baze mari de date*

Antrenarea pentru recunoaşterea de vorbire spontană trebuie făcută considerând problemele specifice pe care le ridică acest proces. În comparaţie cu recunoaşterea cuvintelor izolate, în vorbirea spontană cuvintele sunt rostite şi accentuate în moduri diferite, aşa că este necesară o bază de date mai mare care să cuprindă şi aceste cazuri. Dacă se vrea o recunoaştere independentă de vorbitor atunci este necesară o bază de date şi mai mare. În această situaţie, o etichetare la nivel de fonem sau chiar de cuvânt nu este scalabilă. Pentru a obţine o bază de date destul de mare într-un timp rezonabil, etichetarea se face la nivel de propoziţie şi fişierele de etichetare (fişierele .lab) vor conţine doar ordinea fonemelor. Aceste fişiere sunt antrenate cu metoda *embedded* prezentată în capitolul 2.2.7.

4.2.5.2. *Problema alinierii*

Alinierea prezintă cea mai mare dificultate în procesul de antrenare, din cauza a două probleme. În primul rând, algoritmul Baum-Welch (vezi capitolul 2.2.7.) se opreşte dacă găseşte un maximum local al probabilităţii locale. În cel de al doilea rând, HMM are dificultăţi în a reprezenta durate diferite pentru acelaşi fonem; în modul în care a fost definită probabilitatea de a rămâne în aceeşi stare pentru câteva cadre succesive scade exponenţial. Pentru a evita această problemă, am decis să introducem 5 modele pentru linişte, fiecare având respectiv 50ms, 200ms, 500ms, 1s, 2s. Astfel ne aşteptăm să delimităm cuvintele. O altă măsură pe care o luăm ca să ajutăm alinierea este să iniţializăm HMM-urile alofonelor cu estimate obţinute din foneme antrenate cu date etichetate la nivel de fonem. Câteva înregistrări sunt etichetate la nivel de fonem pentru a obţine HMM-uri antrenate bine. Toate alofonele vor *moşteni* parametrii de model din HMM-ul fonemului central.

4.2.5.3. *Sensibilitatea la erori în timpul etichetării*

Antrenarea *embedded* este sensibilă la erori de etichetare. O eroare de etichetare poate afecta una sau mai multe modele din jurul modelului etichetat eronat. În mod particular, va afecta calculul mediilor şi al varianţelor componentelor mixturii gaussiene din ieşirea HMM-ului. Pentru a depăşi această problemă se construieşte o bază de date de cuvinte izolate. Mai întâi, se antrenează nişte HMM-uri robuste din cuvinte izolate şi apoi se antrenează modelele cu baza de date de vorbire continuă. Baza de date cu vorbire continuă introdusă se testează, apoi, pentru recunoaştere. Fişierele cu rată mare de erori vor fi verificate pentru eventualele erori pe care le conţin.

4.2.5.4. *Numărul mic de apariţii*

Un dezavantaj semnificativ în folosirea alofonelor îl constituie numărul mic de apariţii pentru multe din ele şi din această cauză antrenarea lor devine dificilă. Pentru acesta se foloseşte tehnica stărilor legate. Alofonele care au acelaşii fonem central, vor avea aceiaşi parametrii HMM pentru stările centrale. Aceste stări se pot lega în timpul antrenării astfel ele vor avea aceiaşi parametri dar beneficiind de un număr mai mare de apariţii. Această tehnică este motivată prin faptul că indiferent de fonemele vecine, partea centrală a unui fonem este pronunţată la fel.

4.2.5.5. *Contextul în recunoaşterea vorbirii continue*

În vorbirea continuă, fonemele şi cuvintele sunt pronunţate într-un anumit context; într-o propoziţie nu orice combinaţie de cuvinte este posibilă. Această observaţie ne dă posibilitatea să considerăm unele secvenţe ca fiind mai probabile decât celelalte. Alofonele introduc deja un context faţă de utilizarea fonemelor [137], iar pentru cuvinte putem folosi tehnica n-gramelor. Această tehnică constă în analizarea unui număr foarte mare de texte şi în a calcula probabilitatea ca *n* cuvinte date să fie succesive într-o propoziţie.

4.3. INFRASTRUCTURA SRLV

Experimentele în domeniul recunoaşterii vorbirii includ un număr variat de procese care au în general o complexitate de calcul ridicată. De aceea, pentru studii aprofundate în acest domeniu este nevoie de o infrastructură solidă şi optimizată care să permită rularea a cât mai multor procese în timp cât mai scurt. Două sunt dificultăţiile care implică un efort mai mare în această direcţie:

- Numărul mare al variabilelor de intrare şi particularitatea de rulare pe care o necesită fiecare.
- Numărul mare de fişiere care trebuie prelucrate.

Toată infrastructura a fost organizată în jurul utilitarelor HTK [68]. Aceste utilitare au deja implementaţi algoritmii principali de antrenare a modelelor Markov (Baum-Welch) şi de recunoaştere (Viterbi). În plus, aceste utilitare au implementate şi niste funcţii auxiliare care ajută prelucrarea şi organizarea datelor. Pe de altă parte, interfaţa cu utilizatorul pe care o oferă aceste utilitare presupune organizarea datelor într-un anume fel, de aceea s-au scris, în acest scop, o serie de scripturi în Matlab sau în Java şi s-a construit o procedură de rulare cu paşi simpli şi clari. Procedura include organizarea datelor în directoare într-un mod eficient iar la nivel logic rulările particulare sunt organizate în proiecte. În unele cazuri, de exemplu la implementarea ponderilor rezultate din calculul confuziilor (vezi capitolul 6.1.3.), a fost nevoie de modificarea codurilor sursă a utilitarelor HTK care sunt scrise în limbajul C. Infrastructura solidă împreună cu procedura simplă au făcut posibil ca multe teste să fie rulate de studenţi începători în domeniul recunoaşterii vorbirii. Infrastructura este organizată ca în Fig 4.2. Procesele sunt organizate în 4 etape [27]:

- Pregătirea datelor
- Antrenarea Modelelor
- Recunoașterea
- Verificarea

Fiecare etapă este descrisă în detaliu în paragrafele care urmează.

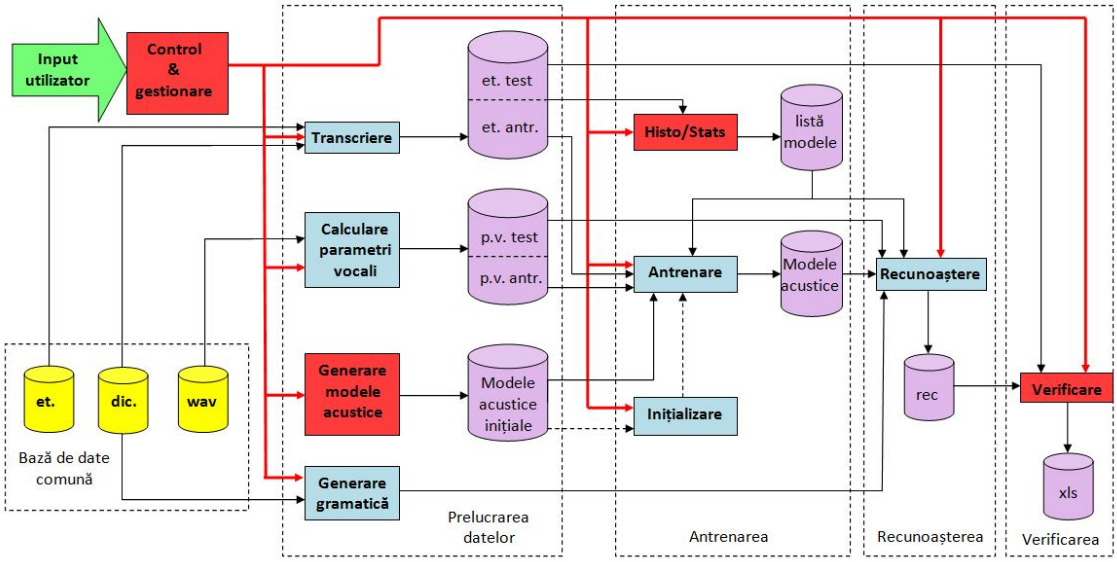

Fig. 4.2. Infrastructura SRLV.

Procesele cu albastru sunt implementate în pachetul de utilitare HTK. Procesele cu roșu a fost implementat de autorul acestei lucrări. Bazele de date în galben sunt disponibile tuturor proiectelor. Ele sunt selectate prin modulul ”control și gestionare”. ”et.” sunt etichetele, ”wav” sunt înregistrările iar ”dic.” este dicționarul fonetic. Datele de culoare violet sunt ieșirile fiecărui proces. ”et. antr.” și ”et. test” sunt etichetele la nivel de unitate de limbă folosite pentru antrenare, respectiv pentru testare. ”p.v. antr.” și ”p.v. test” sunt parametrii de voce calculați din semnalul vocal și sunt folosiți pentru antrenare, respectiv testare. ”rec” conține outputul recunoașterii (în format text). În urma verificării, rezultatele sunt stocate într-un fișier xls.

4.3.1. Pregătirea datelor

Materia primă este baza de date descrisă în capitolul 4.1.2. Ea conține fișierele de semnal vocal (.wav), fișierele cu etichete (.lab, din limba engleză *label*) și dicționarul fonetic (.dic). Fiindcă aceste date sunt folosite în comun de către toate experimentele, ele sunt stocate într-o singură locație. Pe de altă parte, fiindcă fiecare experiment produce date diferite funcție de variabilele folosite, fiecare proiect este organizat în câte un director separat. Organizarea datelor în directoare este arătată în Fig 4.3., iar această sctructură va afecta toate scripturile auxiliare de pregătire a datelor. Etichetele se află în directorul *lab* (din engleză *label*), iar înregistrările în directorul *wav*. În fiecare din cele două subdirectoare datele sunt împărțite în fraze (*phra*), cuvinte (*word*) și silabe (*syl*) funcție de nivelul de etichetare și modul în care au fost pronunțate. Mai departe datele sunt impărțite astfel încât într-un director să se afle numai datele care aparțin unui singur vorbitor. Fișierele de etichetare și de înregistrare conțin și

identificatorul de frază/cuvânt/silabă. Modulul ”control şi gestionare” permite alegerea a orice combinaţie de date ca intrare primind ca input de la utilizator tipul etichetării şi identificatorii de vorbitori şi de fişiere.

Fig. 4.4. arată cum sunt organizate datele în interiorul unui proiect. Organizarea este aceeaşi indiferent dacă unităţile elementare de vorbire sunt foneme, alofone sau silabe. Datele de intrare sunt împărţite în date de antrenare şi date de test. De obicei, pentru test se păstrează 10% din date. Directorul de antrenare conţine mai multe subdirectoare care sunt folosite funcţie de strategia de antrenare aleasă.

Scopul principal a pregătirii datelor este cel de a converti informaţia iniţială în parametrii necesari modelului Markov în procesele de antrenare şi de recunoaştere, ceea ce înseamnă rezolvarea următoarelor probleme:

- Calcularea parametrilor de voce. Aceştia sunt calculaţi din eşantioanele semnalului vocal. Cei mai utilizaţi sunt LPC, MFCC, PLP.
- Transcrierea fonetică a etichetelor. Pentru început datele sunt etichetate după regulile limbii literare dar fără semne de punctuaţie. Cu ajutorul dicţionarului fonetic aceste etichete sunt convertite în succesiunea de fonemele pronunţate pentru cuvântul respectiv.
- Crearea modelelor acustice iniţiale. Un model Markov este caracterizat de un număr mare de parametri, cum ar fi numărul de stări, numărul de componente ale mixturii gaussiene, numărul de parametrii, etc. Dacă la experimente se doreşte variaţia acestor parametrii, modelul prototip utilizat pentru iniţializare trebuie editat de fiecare dată.
- Generarea gramaticii. Prin gramatică, înţelegem aici restricţiile introduse asupra succesiunii de cuvinte rezultate în urma recunoaşterii.

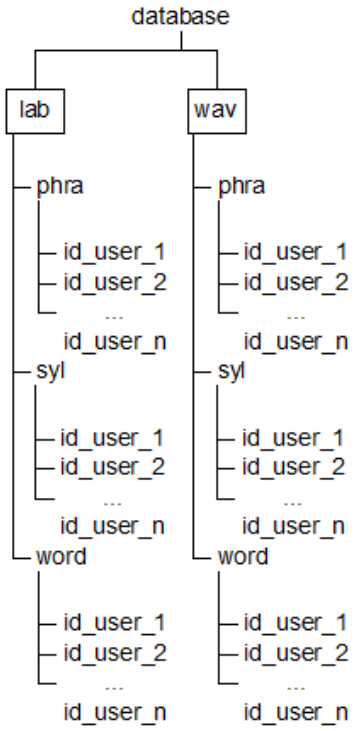

Fig. 4.3. Organizarea datelor de intrare comune

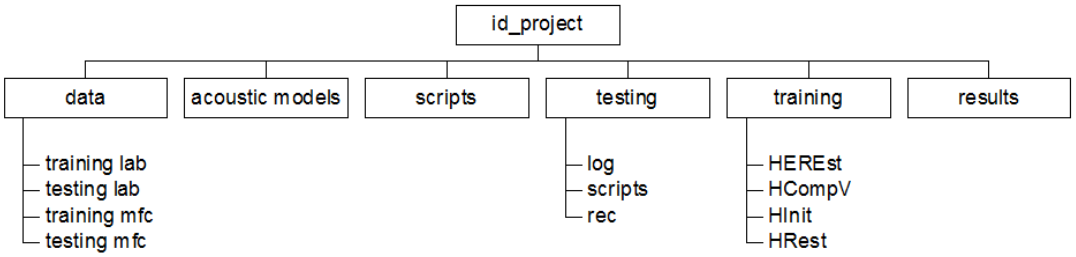

Fig. 4.4. Organizarea datelor de intrare în interiorul unui proiect

4.3.2. Antrenarea modelelor

Antrenarea modelelor presupune estimarea parametrilor HMM pornind de la înregistrări şi etichetele respective cum se prezintă în capitolul 2.2.7. Această infrastructură permite selectarea uşoară a datelor pentru antrenare prin modulul ”control şi gestionare”. Se pot utiliza diferite strategii de antrenare (iniţializare folosind media şi varianţa globală a parametrilor de voce, iniţializare prin antrenare de foneme, etc.). Utilitarul HTK HEREst realizează, la fiecare rulare, o singură iteraţie a algoritmului Baum-Welch. Pentru a alege cea mai bună iteraţie se testează fiecare cu baza de date de test. Infrastructura permite şi antrenarea altor unităţi de vorbire, cum ar fi trifonemele sau silabele. În fine, se poate implementa cu uşurinţă şi configura şi tehnica stărilor legate.

4.3.3. Recunoaştera

Recunoaşterea primeşte ca intrare modelele acustice obţinute la etapa de antrenare, înregistrările care trebuie recunoscute şi gramatica care a fost generată în etapa de pregătirea datelor. Procesul de recunoaştere este construit în jurul utilitarului HVite din pachetul HTK. Toate log-urile generate de acest tool sunt stochate în folderul *log* (Fig. 4.4.). Aceste log-uri conţin erori (dacă s-au produs) care ajută la procesul de depanare. În plus, în ele se scriu parametrii vocali pentru înregistrările recunoscute. Aceştia sunt utili la analiza comportamentului unui nou algoritm de optimizare. De exemplu, ele sunt folosite pentru calculul confuziilor şi a incertitudinii tratate în capitolul 6. Ieşirile acestui proces sunt nişte fişiere text care conţin recunoaşterea. Ele mai pot conţine şi graniţele în timp a cuvintelor, fonemelor sau chiar ale stărilor decodate al HMM-urilor.

4.3.4. Verificarea

După recunoaştere, trebuie verificat dacă textul generat este corect. Acesta este comparat cu informaţia din etichetele folosite pentru test. Pentru comparaţia cuvintelor s-a implementat un script de la zero, iar pentru comparaţia frazelor se poate folosi utilitarul HResults din pachetul HTK, care face o măsurare cantitativă a erorilor. În primul caz, rezultatele sunt scrise într-un fişier *xls* pentru o prelucrare mai uşoară. În afară de rata de recunoaştere, în fişier se scriu şi cuvintele recunoscute corect şi cele recunoscute greşit (în acest caz se specifică şi cu ce cuvânt s-a confundat). O asemenea prezentare a rezultatelor

ajută la detecția rapidă a problemelor. De exemplu, se poate observa ușor care vorbitor are cea mai mică rată de recunoaștere sau cu ce se confundă cel mai des un anumit fonem.

CAPITOLUL 5

OPTIMIZĂRI PENTRU SISTEMUL DE RECUNOAŞTERE AL LIMBAJULUI VORBIT

În timpul anilor, s-au evidenţiat în linii mari algoritmii care dau rezultatele cele mai bune, atât din punct de vedere a ratei de recunoaştere, cât şi din punct de vedere al vitezei de execuţie. Cu toate acestea, particularităţile sistemelor sunt aşa de multe şi parametrii de intrare aşa de variaţi încât găsirea configuraţiei optime nu este deloc o sarcină uşoară. Această configuraţie depinde mult de limbă, de dimensiunea bazei de date, tipul aplicaţiei, etc. În schimb, se pot observa nişte tendinţe care dau indicaţii importante despre configuraţia optimă. În lucrarea de faţă, pentru început, se variază parametrii ce caracterizează un HMM (numărul de stări, numărul de parametri vocali, numărul de mixturi, etc.) şi apoi parametrii algoritmului de recunoaştere cum ar fi dimensiunea fascicolului, penalizarea de inserţie, dimensiunea dicţionarului, etc. O dată găsită configuraţia optimă, s-au încercat diferite modificări ale algoritmului de recunoaştere cât şi ale procedurilor astfel încât să se obţină rate mai bune de recunoaştere şi viteze mai mari de execuţie.

Mai trebuie menţionat că în acest capitol testele au fost făcute pe o bază de date ”curată”. După ce se evidenţiază tendinţele pe care le are rata de recunoaştere la modificarea parametrilor sistemului, se încearcă optimizări ulterioare şi în condiţii adverse care sunt tratate în Capitolul 6.

5.1. CĂUTAREA CONFIGURAŢIEI OPTIME DIN PUNCT DE VEDERE AL RATEI DE RECUNOAŞTERE

La construirea unui sistem complex care urmează a fi optimizat, este importantă definirea unor configuraţii de referinţă. După aceea, celelalte optimizări se pot introduce pe rând şi noile rezultate se pot compara cu rezultatele de referinţă. Dificultatea acestei sarcini stă în caracteristicile HMM şi în natura algoritmilor de antrenare şi de testare [170]:

- O modificare în topologia HMM (de exemplu numărul de stări) necesită o nouă serie de antrenare/testare ceea ce implică organizarea datelor într-un proiect nou cum este explicat în capitolul 4.3.
- Algoritmul de antrenare Baum-Welch este consumator de resurse si de timp. Nu se ştie a priori numărul de iteraţii de antrenare care duce la cele mai bune rezultate.
- Cea mai mare rată de recunoaştere nu se obţine cu modelele obţinute în urma unui număr anume de iteraţii de antrenare. Din acest motiv, pentru a găsi rata de recunoaştere cea mai bună trebuie rulat procesul de recunoaştere (testare) după fiecare iteraţie.
- Algoritmul de recunoaştere Viterbi este consumator de resurse si de timp, iar optimizările pentru îmbunătăţirea vitezei nu se pot introduce în această etapă dat fiind că unele din ele pot afecta rata de recunoaştere (este vorba despre algoritmi la care se tolerează o mică scădere a ratei de recunoaştere cu avantajul obţinerii unei reduceri semnificative de complexitate a algoritmului).

Ţinând cont de aceste aspecte, este evident că nu se pot încerca toate combinaţiile de variaţie a parametrilor sistemului (există mii de combinaţii). În schimb, se poate ţine cont de o tendinţă generală: cu cât baza de date este mai variată cu atât modelul trebuie să fie „mai detaliat”. Prin model „mai detaliat” se înţelege un model cu număr mai mare de stări şi număr mai mare de parametri.

Valorile de referinţă pentru parametri de model şi rezultatele recunoaşterii obţinute cu acestea sunt prezentate în Tabelul 5.1. Fiecare încercare constituie un SRLV independent, de aceea fiecărui proiect îi este dat un identificator. Testele au fost realizate cu baza de date 4, care conţine înregistrări cu 5 vorbitori (vezi capitolul 4.1.2.). Pentru test s-a folosit 10% din baza de date, adică un acelaşi set de 1000 de cuvinte (din cele 10000) pentru fiecare vorbitor. S-a pornit cu un model cu 6 stări, 2 mixturi gaussiene şi 26 de parametri vocali pentru fiecare vector de observaţii (12 parametri MFCC + energia împreună cu derivatele sale de ordin I, de aici notaţia MFCC_E_D). Din rezultatele obţinute se observă că, deşi bazele de date pentru vorbitori diferiţi au aceaşi dimensiune (aceleaşi 10000 de cuvinte), ratele de recunoaştere diferă semnificativ. Rata de recunoaştere pentru vorbire independentă de vorbitor este mult mai mică datorită numărului nepotrivit de stări (vezi capitolul 5.1.2.) şi variabilităţii mai mari a bazei de date, ceea ce era de aşteptat.

Tabelul 5.1. Valorile de referinţă pentru recunoaşterea dependentă de vorbitor şi independentă de vorbitor

Identificator SRLV	Vorbitor	HMM	Restricţii gramaticale	Rata de recunoaştere [%]
011	toţi vorbitorii	• modelare de foneme • 6 stări • 2 componente de mixturi gaussiene • MFCC_E_D	• fără restricţii gramaticale • toate cuvintele din dicţionar	48.93
019	1			77.67
020	2			78.07
021	3			69.42
022	4			74.45
065	5			71.43

5.1.1. Variaţia ratei de recunoaştere cu iteraţia de antrenare

Algoritmul de recunoaştere Baum–Welch se opreşte atunci când probabilitatea logaritmică nu mai creşte de la iteraţia precedentă la cea curentă (vezi capitolul 2.2.7.). Această condiţie însă nu garantează că modelele obţinute la această iteraţie asigură şi cea mai bună rată de recunoaştere. Baza de date fiind mare, probabilitatea de recunoaştere pentru o iteraţie poate creşte pentru unele modele dar scade pentru altele, iar însumarea scorurilor obținute de modele nu constituie tocmai cea mai bună soluţie. De aceea, o soluţie alternativă şi mai eficientă (din punct de vedere al calității) este ca după fiecare iteraţie să se ruleze procesul de recunoaştere pe baza de date de test şi să se aleagă iteraţia cu cele mai bune rezultate. În Fig. 5.1., Fig. 5.2. și Fig. 5.3., sunt prezentate câteva evoluţii ale ratelor de recunoaştere cu numărul de iteraţii de antrenare obţinute din diferite teste.

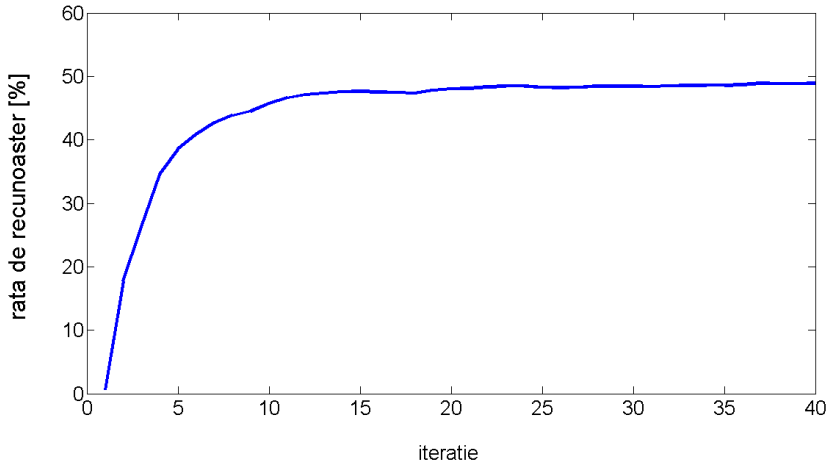

Fig. 5.1. Exemplu în care rata de recunoaștere crește monoton cu iterația de antrenare

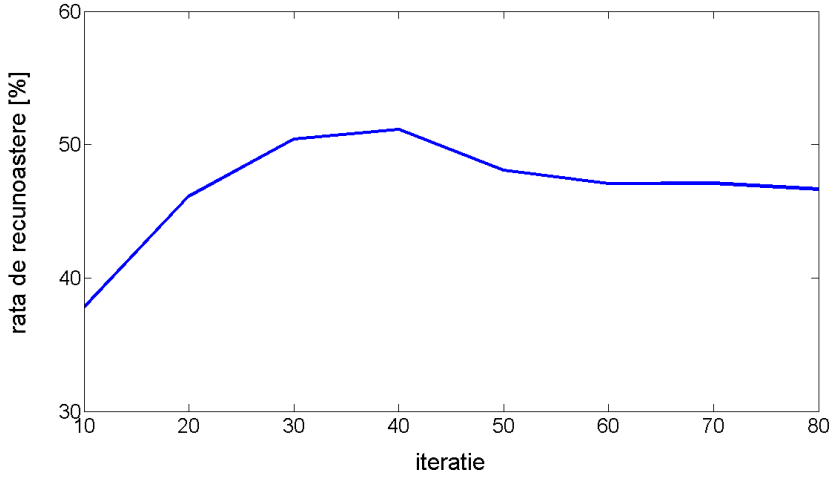

Fig. 5.2. Exemplu în care rata de recunoaștere are un maxim funcție de iterația de antrenare

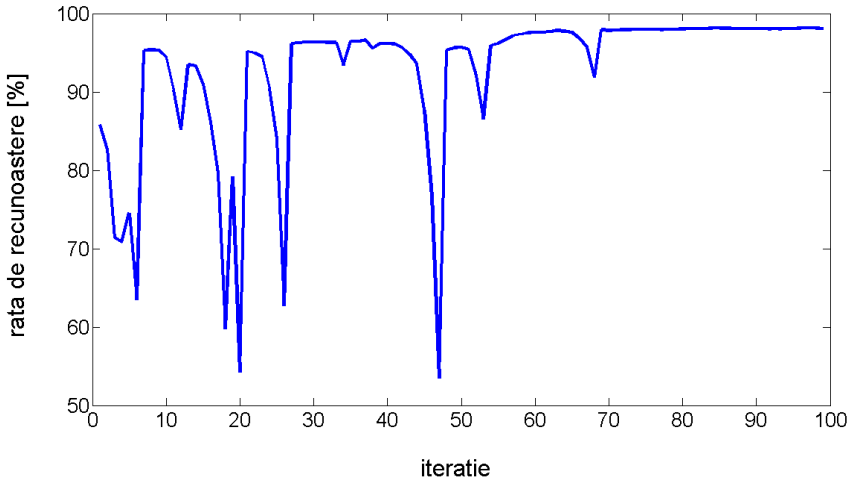

Fig. 5.3. Exemplu în care rata de recunoaștere crește monoton cu iterația de antrenare, dar are multe minime locale

Rezultatele acestor teste indică o saturaţie a ratei de recunoaştere când numărul de iteraţii devine foarte mare. Excepţie fac unele cazuri ca în Fig. 5.2., unde se obţine un maxim la iteraţia 40. Totuşi, în celelalte figuri, tendința este aceea de creştere asimptotică. Aceste rezultate sunt explicabile prin natura algoritmului de antrenare Baum-Welch. De la o iteraţie la alta, algoritmul încearcă să alinieze cât mai bine semnalul vocal cu etichetele care conţin transcrierea fonetică. Aceasta înseamnă că modelele reprezintă din ce în ce mai bine semnalul vocal din baza de date de antrenare. Cum baza de date pentru testare este diferită de cea de antrenare există riscul ca, după mai multe iteraţii, modelele să reprezinte bine doar eşantioanele din baza de date de antrenare şi să piardă din generalitate, dând rezultate mai slabe cu baza de date de testare. Acest fenomen, care este prezentat în şi Fig. 5.3., este greu de prevăzut. De aceea, pentru o precizie mare, se pot testa modelele obţinute la fiecare iteraţie pentru un număr foarte mare de iteraţii. Această procedură, însă, este consumătoare de timp (zeci de ore) cum este arătat şi în capitolul 5.2.2. După câteva încercări am stabilit empiric că pentru fiecare configuraţie să se testeze modelele obţinute din 10 în 10 iteraţii până la 100.

5.1.2. Variaţia numărului de stări

Alegerea numărului de stări este legată de următoarele aspecte: durata fonemelor, variabilitatea în timp a parametrilor şi variabilitatea bazei de date. Aşa cum e conceput un HMM, o stare modelează o distribuţie de valori pe care o are fonemul (sau o altă unitate de voce) într-un segment al evoluţiei sale în timp. Dacă variabilitatea fonemului este mare, atunci este necesar un număr mai mare de stări care să reprezinte aceste variaţii. Cu acelaşi raţionament, foneme diferite au durate şi variaţii diferite, atunci şi modelele diferite pot avea număr diferit de stări. La prelucrarea datelor pentru antrenare, este mai uşor de lucrat cu un număr fix de stări pentru toate modelele.

Primul experiment constă într-o serie de teste unde s-a modificat numărul de stări (același pentru toate modelele), şi rezultatele sunt prezentate în Tabelul 5.2. Experimentul s-a realizat pe o bază de date independentă de vorbitor, folosind două mixturi gaussiene. Se observă că rata de recunoaştere creşte cu numărul de stări şi apoi se saturează. Creşterea semnificativă a ratei de recunoaştere cu mărirea numărului de stări arată că valoarea iniţială de 6 stări era insuficientă pentru reprezentarea evoluției parametrilor de voce ai fonemului.

Tabelul 5.2. Variația ratei de recunoaștere independentă de vorbitor cu numărul de stări

Identificator SRLV	Vorbitor	HMM		Restricții gramaticale	Rata de recunoaștere [%]
011	toți vorbitorii	• modelare de foneme • 2 componente de mixturi gaussiene • MFCC_E_D	6 stări	• fără restricții gramaticale • toate cuvintele din dicționar	48.93
013			7 stări		47.52
014			8 stări		56.40
051			9 stări		63.77
055			10 stări		67.79
057			11 stări		69.67

Același experiment s-a făcut şi pentru bazele de date dependente de vorbitor (vezi Tabelul 5.3., unde sunt prezentate valorile obținute cu vorbitorul 02). Se observă că numărul optim de stări este mai mic, ceea ce confirmă că o bază de date mai variată necesită un model HMM cu mai multe stări.

Tabelul 5.3. Variația ratei de recunoaștere dependentă de vorbitor cu numărul de stări

Identificator SRLV	Vorbitor	HMM		Restricții gramaticale	Rata de recunoaștere [%]
201	02	• modelare de foneme • 2 componente de mixturi gaussiene • MFCC_E_D	3 stări	• fără restricții gramaticale • toate cuvintele din dicționar	60.16
202			4 stări		64.23
203			5 stări		71.40
204			6 stări		78.07
205			7 stări		73.72
206			8 stări		72.11

În realitate, fonemele au durate diferite şi de aceea este firesc ca şi modelele să aibă număr diferit de stări. Algoritmi de antrenare (Baum-Welch) şi de recunoaştere (Viterbi) permit ca foneme diferite să fie modelate cu număr diferit de stări. Complexitatea algoritmului Viterbi depinde de numărul de stări pe care le au HMM-urile, de aceea este preferat ca pentru aceeași performanță din punct de vedere al ratei de recunoaștere să se aleagă soluția cu numărul total de stări cel mai mic. Este de așteptat ca fonemele de durată mai mare să fie modelate cu mai multe stări. În acest scop, ar fi utilă o informație despre duratele fonemelor. Această informație se poate obține automat (fără să fie nevoie de stabilire manuală a granițelor între foneme) fie plecând de la foneme deja antrenate, fie din decodarea

Viterbi care realizează implicit o aliniere la nivel de fonem între semnalul vocal şi transcrierea fonetică.

Pentru a calcula numărul de stări pentru modelul unui fonem se pleacă de la modelele deja antrenate cu număr fix de stări. Variația numărului de stări s-a făcut folosind modelele cu 12 stări. Din aceste modele se calculează distribuția duratelor cu formula (5.1):

$$p(t) = \sum_{j=1}^{S} \alpha_j(t) a_{jS} \tag{5.1}$$

unde $p(t)$ este probabilitatea ca durata fonemului să fie t cadre (ferestre), $\alpha_j(t)$ este probabilitatea progresivă prezentată la capitolul 2.2.7., a_{jS} este probabilitatea de tranziție de la stare j la starea S, iar S este numărul total de stări.

Distribuția duratelor fonemelor se mai poate calcula din rezultatele recunoașterii. Algoritmul de recunoaștere Viterbi realizează implicit și alinierea semnalului vocal cu modelele recunoscute. Cu alte cuvinte, după procesul de recunoaștere avem și granițele care indică segmentele ocupate de fiecare model în semnalul vocal.

In Fig. 5.4., este prezentată distribuția duratelor pentru fonemul ”ă” obținută atât prin calcul din modelele antrenate cât și din rezultatele recunoașterii. Diferențele între cele două distribuții nu sunt mari și acest lucru se respectă și pentru celelalte foneme.

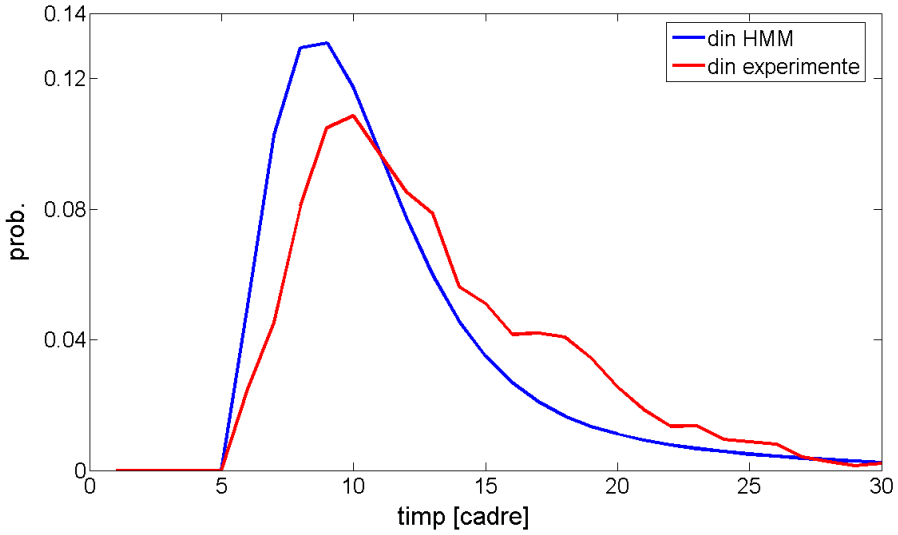

Fig. 5.4. Distribuțiile pentru durata fonemului ”ă” obținute din HMM și respectiv din rezultatele recunoașterii.

Se poate construi un nou sistem de recunoaștere unde fiecare model are un număr de stări egal cu media distribuției duratei sale.

Metoda cu număr variabil de stări este avantajoasă în cazul în care la aceleaşi performanţe numărul total de stări (pentru toate modelele) este mai mic. Astfel, dimensiunea reţelei devine mai mică iar numărul de calcule se reduce (vezi capitolul 5.2.1.).

5.1.3. Variaţia parametrilor si a numărului de mixturi

Alegerea parametrilor de voce este importantă, pentru că pe baza acestora se iau deciziile la recunoaştere. Dacă pentru un codec cel mai bun set de parametri de voce este cel care reproduce cât mai fidel vocea, la recunoaştere (care este un proces de decizie) setul de parametrii trebuie să aibă capacitatea cât mai mare de discriminare între foneme (cu alte cuvinte, parametrii de voce să ia valori cât mai diferite pentru foneme diferite). Trebuie aleşi parametri care reprezintă cât mai fidel semnalul vocal şi care discrimină cel mai bine între modele (foneme). O altă proprietate de apreciat în setul de parametrii este ca parametrii să fie la fel de relevanţi, căci contribuţia dată de fiecare nu este ponderată. De exemplu, în cazul transformării cosinus discrete (DCT), energia este concentrată în primii coeficienţi, ceea ce înseamnă că un număr mare de astfel de parametrii devine inutil. Studiile arată că parametri care dau cele mai bune rezultate la recunoaştere sunt MFCC şi PLP [50]. Tabelul 5.4. arată rezultatele obţinute pentru fiecare tip de parametri în cazul în care s-au folosit 12 stări şi o mixtură gaussiană cu două componente pentru modelarea ieşirii ale unei stări. Şi aici, se observă că cele mai bune rezultate sunt obţinute pentru MFCC şi PLP. Deşi ratele recunoştere pentru aceste două tipuri de parametri sunt aproximativ egale, mulţimea cuvintelor recunoscute corect nu este aceeaşi în ambele cazuri (vezi Tabelul 5.5.). Numărul de cuvinte recunoscute corect în ambele cazuri constituie aproximativ 80% din numărul total de cuvinte. 11% din cuvinte nu au fost recunoscute cu niciunul dintre tipurile te parametri. În schimb, există un număr considerabil (aproximativ 8%) de cuvinte care au fost recunoscute doar cu un singur tip de parametri. Prin combinarea scorurilor obţinute cu cele două tipuri de parametri, există un potenţial de creştere a ratei de recunoaştere până la aproximativ 88%. O metodă de a le combina este prin introducerea conceptului de *stream* (adică rezultate care provin din surse diferite).

Tabelul 5.4. Rata de recunoaștere obținută utilizând tipuri diferite de parametri de voce

Tipul de parametri	MFCC	PLP	LPC
Rata de recunoaştere [%]	83.6	84.69	62

Tabelul 5.5. Distribuția cuvintelor recunoscute cu cele două tipuri de parametri de voce MFCC și PLP

Cuvinte recunoscute cu amândouă tipuri de parametri	79.98 %
Cuvinte recunoscute doar cu MFCC	3.62 %
Cuvinte recunoscute doar cu PLP	4.71 %
Cuvinte recunoscute eronat cu amândouă tipuri de parametri	11.69 %

După alegerea tipului parametrilor, se pune problema numărului optim de componente ale mixturii gaussiene. Mixturile modelează distribuţia valorilor pentru un parametru. Dacă

valorile parametrului sunt concentrate în jurul unei singure valori, atunci o singură componentă gaussiană este de ajuns pentru modelarea distribuţiei, cum este cazul recunoaşterii dependent de vorbitor (vezi Tabelul 5.6.). Când baza de date este mai variată, de exemplu pentru recunoaştere independentă de vorbitor, o singură funcţie nu mai este de ajuns şi se folosesc două sau trei componente ale mixturii gaussiene. Dacă se obţine aceeaşi rată de recunoaştere pentru un număr diferit de mixturi, atunci se preferă soluţia cu număr mai mic de mixturi, deoarece cu creşterea numărului de mixturi creşte şi numărul de calcule necesare la recunoaştere.

Tabelul 5.6. Variația ratei de recunoaștere cu numărul de componente ale mixturii gaussiene

<table>
<tr><th>Identificator SRLV</th><th>Vorbitor</th><th colspan="2">HMM</th><th>Restricții gramaticale</th><th>Rata de recunoaștere [%]</th></tr>
<tr><td>011</td><td rowspan="3">toți vorbitorii</td><td rowspan="6">• modelare de foneme
• 6 stări
• MFCC_E_D</td><td>2 componente de mixturi gaussiene</td><td rowspan="6">• fără restricții gramaticale
• toate cuvintele din dicționar</td><td>48.93</td></tr>
<tr><td>016</td><td>3 componente de mixturi gaussiene</td><td>56.04</td></tr>
<tr><td>017</td><td>4 componente de mixturi gaussiene</td><td>35.23</td></tr>
<tr><td>029</td><td rowspan="3">2</td><td>1 componentă de mixturi gaussiane</td><td>78.07</td></tr>
<tr><td>020</td><td>2 componente de mixturi gaussiene</td><td>78.07</td></tr>
<tr><td>059</td><td>3 componente de mixturi gaussiene</td><td>76.36</td></tr>
</table>

Studiile arată că o îmbunătăţire semnificativă a acurateţei recunoaşterii se obţine prin introducerea derivatelor de ordin întâi şi doi ale parametrilor de voce. Derivatele de ordin doi se mai numesc accelerații. Derivatele măsoară viteza cu care se schimbă parametrul vocal şi această viteză este caracteristică fiecărui fonem, de aceea derivatele contribuie semnificativ la discriminarea între foneme. Tabelul 5.7. arată cum se modifică ratele de recunoaştere funcţie de introducerea derivatelor. Testele au fost făcute pe o bază de date independă de vorbitor. După cum se observă, introducerea derivatelor creşte semnificativ rata de recunoaştere iar introducerea acceleraţiilor face ca rata de recunoaştere să scadă. Această scădere poate avea două motive: fie baza de date nu este destul de mare astfel încât să construiască un model care să generalizeze acceleraţiile pentru toate variaţiile unui fonem, fie variaţiile fonemelor sunt foarte mari astfel încât este nevoie de un model cu mai multe componente pentru mixturile gaussiene pentru modelarea acceleraţiilor. Şi în acest caz, dacă se obţine aceeaşi rată de recunoaştere atât cu derivate cât şi fără derivate, se preferă soluția fără derivate, care implică un număr mai mic de calcule.

Tabelul 5.7. Efectul introducerii derivatelor parametrilor de voce asupra ratei de recunoaștere

<table>
<tr><th>Identificator SRLV</th><th>Vorbitor</th><th colspan="2">HMM</th><th>Restricții gramaticale</th><th>Rata de recunoaștere [%]</th></tr>
<tr><td>011</td><td rowspan="3">toți vorbitorii</td><td rowspan="3">• modelare de foneme
• 6 stări
• 2 componente de mixturi gaussiene</td><td>MFCC_E_D</td><td rowspan="3">• fără restricții gramaticale
• toate cuvintele din dicționar</td><td>48.93</td></tr>
<tr><td>018</td><td>MFCC_E_D_A</td><td>35.00</td></tr>
<tr><td>048</td><td>MFCC_E</td><td>19.66</td></tr>
</table>

5.1.4. Variaţia unităţilor de vorbire folosite pentru modelare

Unitățile de vorbire care pot fi modelate într-un sistem de recunoaștere a vorbirii sunt fonemele, difonemele, trifonemele, silabele și cuvintele. Alegerea unității cele mai potrivite depinde de aplicația pentru care se folosește sistemul de recunoaștere, de dimensiunea bazei de date, caracteristicile ei și de caractersticile limbii. Criteriile după care se alege tipul modelului sunt:

- Număr de apariții cât mai mare, care face ca modelele să fie cât mai generale. Pentru o bază de date dată cel mai mare număr de apariții îl au fonemele și apoi difonemele, alofonele, etc.
- Datele folosite pentru modelare să prezinte cât mai puține variații, pentru că astfel, se obțin modele mai robuste. Din acest motiv, în cazul în care parametri unui fonem diferă mult de context (de pildă, dacă fonemul face parte în silaba accentuată sau nu), se preferă construirea unui model separat pentru fiecare caz.
- Modelele trebuie să prezinte cât mai multe constrângeri deoarece acestea îmbunătățesc acuratețea recunoașterii. În acest sens, cuvintele prezintă cele mai mari constrângeri și apoi silabele, alofonele, difonemele și fonemele. Trebuie menționat că aici este vorba de constrângerile inerente pe care le prezintă modelul. De exemplu, odată recunoscut cu probabilitate mare alofonul *c-a+s,* asta impune ca înaintea lui sa fie un alofon din clasa **-c+a*, iar după el să fie un alofon din clasa *a-s+**. Alte constrângeri se pot impune prin gramatică (vezi capitolul 5.1.5.).

Pentru o aplicație interactivă om-mașină bazată pe comenzi se pot alege drept modele cuvintele (comenzile). O aplicație de recunoaștere a vorbirii continue independentă de vorbitor prezintă o variabilitate sporită din punct de vedere a modelării care provine din vocea vorbitorului, accentul, modul frazei (afirmativ, interogativ, imperativ), tonalitatea, etc. Modelarea cuvântului în acest caz nu este potrivită. De aceea se folosesc fonemele sau alofonele.

În această lucrare, au fost testate două tipuri de modele: fonemele și alofonele. Rezultatele obținute sunt prezentate in Tabelul 5.8.

Tabelul 5.8. Rata de recunoaștere funcție de tipul unității de vorbire

<table>
<tr><th>Identificator SRLV</th><th>Vorbitor</th><th colspan="2">HMM</th><th>Restricții gramaticale</th><th>Rata de recunoaștere [%]</th></tr>
<tr><td>011</td><td>toți vorbitorii</td><td rowspan="4">• 6 stări
• 2 componente de mixturi gaussiene
• MFCC_E_D</td><td rowspan="2">• modelare de foneme
• fără inițializare</td><td rowspan="4">• fără restricții gramaticale
• toate cuvintele din dicționar</td><td>48.93</td></tr>
<tr><td>019</td><td>1</td><td>77.67</td></tr>
<tr><td>042</td><td>toți vorbitorii</td><td rowspan="2">• modelare de alofone
• inițializare: HMM pentru foneme</td><td>73.21</td></tr>
<tr><td>038</td><td>1</td><td>90.05</td></tr>
</table>

Testele au fost făcute atât pentru cazul dependent de vorbitor cât și pentru cazul independent de vorbitor. Avantajul folosirii alofonelor este evident. Cu această bază de date, unde fiecare alofon are un număr considerabil de apariții (vezi capitolul 4.1.3), alofonele au suficiente date pentru a se antrena. În alte condiții, folosirea fonemelor ar fi fost obligată. Două sunt motivele pentru care alofonele oferă rezultate mai bune: a) constrângerile pe care le aduc, b) diferențierea fonemului după context (fonemele vecine), care duce la scăderea variației datelor de antrenare și rezultând, astfel, unele modele robuste.

5.1.5. Restricţii de limbă. Penalizarea de inserţie

Restricțiile de limbă vin în ajutorul sistemului de recunoaștere prin eliminarea unor combinații de modele din cele posibile. Se pleacă de la premisa că nu toate succesiunile sunt posibile, și atunci prin anumite tehnici se decide o mulțime mai restrânsă de succesiuni din care sistemul de recunoaștere poate să aleagă. În capitolul 5.1.4., este arătat cum trecerea de la foneme la alofone aduce niște constrângeri suplimentare. O altă constrângere este definirea unui vocabular (set de cuvinte), presupunând că vorbitorii vor pronunța cuvinte doar din acel dicționar.

Din testele făcute, s-a observat că o mare parte din erori proveneau din faptul că un cuvând mai lung sau compus era fragmentat în două sau mai multe cuvinte mai scurte asemănătoare. Impunând o gramatică (în sensul de restricție cum este definită în capitolul 2.4.), astfel încât sistemul să recunoască doar succesiunea *liniște-cuvânt-liniște,* rata de recunoaștere crește semnificativ (în valoare absolută cu 4 procente, vezi Tabelul 5.9.). Această constrângere este pusă știind dinainte că toate fișierele din baza de date 4 (vezi capitolul 4.1.2) conțin un singur cuvânt.

Un alt test s-a făcut folosind un dicționar redus cu care s-au obținut rezultate și mai bune (98,49 %). Dicționarul redus conține doar cuvintele din baza de date de test (1000 din cele 10000 totale), de aceea sistemul, având de ales dintr-o mulțime mai redusă, recunoaște mai bine. Reducerea dicționarului se justifică prin faptul că există aplicații unde numărul de cuvinte folosite (sub formă de comenzi, de exemplu) este mic (sub 1000 de cuvinte). În aceste condiții, aceste teste capătă importanță.

Tabelul 5.9. Rata de recunoaștere funcție de restricțiile impuse

Recunoaștere dependentă de vorbitor (Vorbitorul 2, HMM cu 6 stări, 2 componente de mixturi gaussiene, MFCC_E_D)	Rata de recunoaștere [%]
Foneme (fără restricții)	74,25
Alofone	90,25
Alofone + Gramatică	94,68
Alofone + Gramatică + Dicționar redus	98,49

Analizând cuvintele recunoscute greşit, se observă că, în multe cazuri, în locul cuvântului corect au fost recunoscute 2-3 cuvinte mai scurte dar similare cu părţi ale cuvântului corect. Această problemă se poate rezolva prin introducerea unei penalizări de inserţie. Penalizarea de inserţie constă în adăugarea unui cost mai mare la tranziţia între cuvinte. în graful stărilor. Probabilităţii logaritmice a jetonului (token) care ajunge la sfârşit de cuvânt, i se va scădea o valoare predefinită (având în vedere că la recunoaștere se folosește ca scor de similitudine probabilitatea logaritmică, scăderea unei valori din acest scor este echivalentă cu creșterea costului). În Tabelul 5.10., sunt prezentate câteva valori a ratei de recunoaştere funcţie de valoarea penalizării de inserţie. HMM utilizat are 12 stări 3 componente de mixturi gaussiene, MFCC_E_D, iar recunoașterea este independentă de vorbitor.

Tabelul 5.10. Variația ratei de recunoaștere funcție de valoarea penalizării de inserție

Valoarea penalizării de inserţie	Rata de recunoaştere [%]
0	84.5
20	88.24
40	89.91
60	90.26
80	90.14
100	89.2

Penalizarea de inserţie avantajează cuvintele lungi, de aceea există o valoare optimă. Pentru acest sistem şi pentru baza de date folosită această valoare este aproximativ 60.

5.1.6. Variația ratei de recunoaștere cu mărirea bazei de date

În orice sistem SRLV, cu cât baza de date este mai mare, cu atât cresc șansele ca modelele acustice obținute prin antrenarea ei, să reprezinte mai bine unitățile de limbă alese. Procesul de achiziție a bazei de date este costisitor și atunci e bine de știut care este îmbunătățirea modelelor la mărirea bazei de date. Această îmbunătățire se va măsura prin rata de recunoaștere obținută pentru diferite dimensiuni ale bazei de date. Este de aștepat, ca odată

cu mărirea bazei de date, rata de recunoaștere să se satureze. Acest fenomen se datorează faptului că, deși variabilitatea vorbirii este mare, ea nu este infinită și atunci există o dimensiune a bazei de date peste care orice adăugare de cuvinte nu modifică semnificativ modelul acustic.

S-au făcut, așadar, antrenări pentru dimensiuni diferite ale bazei de date, începând cu 100 de cuvinte până la 9000 de cuvinte. Testele s-au făcut pe o bază de date dependentă de vorbitor, folosind un HMM pentru foneme cu 12 stări şi 3 mixturi. Parametrii vocali folosiţi sunt MFCC și energia împreună cu derivatele sale. Rezultatele sunt prezentate în Fig 5.5.

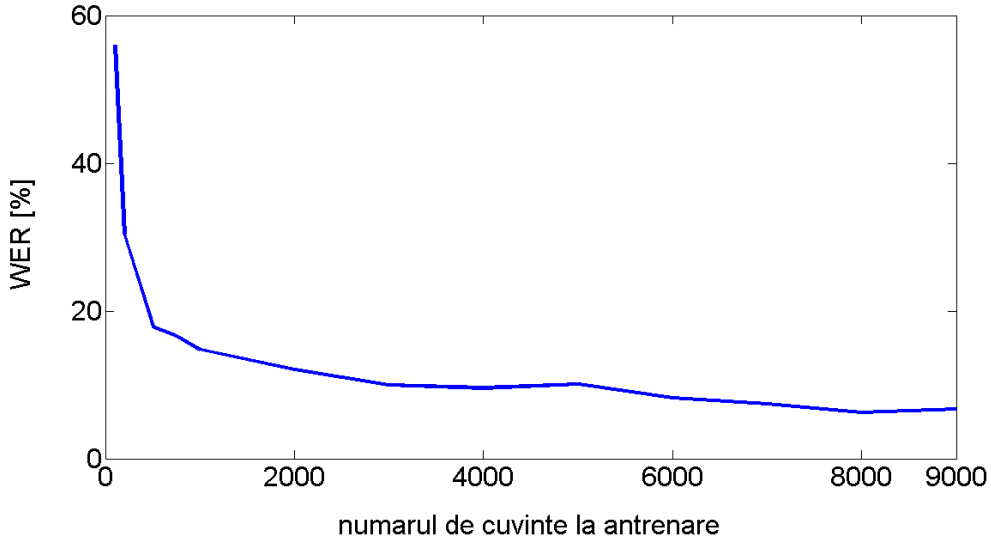

Fig. 5.5. Variația ratei de recunoaştere cu mărirea bazei de date

Se observă că pentru valori mari ale bazei de date (8000-9000 de cuvinte), rata de recunoaștere începe să se satureze. Astfel se poate trage concluzia că pentru o bază de date de cuvinte izolate dependentă de vorbitor sunt necesare aproximativ 8000 de cuvinte pentru obținerea ratei de recunoaștere maxime.

5.1.7. Precizia măsurătorilor funcție de baza de date de test aleasă

O practică răspândită în domeniul recunoașterii limbajului vorbit, care s-a folosit și în această lucrare, este cea de a alege 90% din baza de date pentru antrenare și 10% pentru testare. Cuvintele se aleg în mod aleator astfel încât sistemul să rămână echilibrat, adică să nu se aleagă pentru testare cuvinte cu o caracteristică particulară. Dar cât de mult variază rata de recunoaștere funcție de ce cuvinte se aleg pentru testare/antrenare? În acest scop s-a făcut următorul experiment: Cele 10 000 de cuvinte ale bazei de date au fost sortate aleator și s-au format mulțimi disjuncte de câte 1000 de cuvinte. Pentru fiecare mulțime de 1000 de cuvinte folosită pentru testare, s-au construit modele acustice cu restul de 9000 de cuvinte. Pentru fiecare caz s-a măsurat rata de recunoaștere. Experimentele s-au făcut pe o bază de date dependentă de vorbitor, folosind un HMM pentru foneme cu 12 stări și 3 mixturi. Parametrii

vocali folosiți sunt MFCC și energia împreună cu derivatele sale. Rezultatele sunt prezentate în Fig. 5.6.

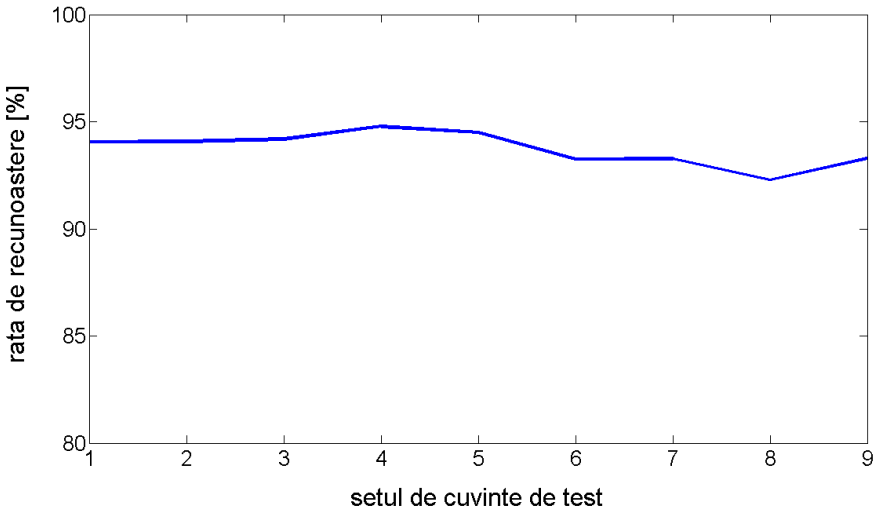

Fig. 5.6. Fluctuația ratei de recunaștere funție de baza de date de tes aleasă

Media ratei de recunoaștere este 93.75%, iar deviația standard este 0.79. Deviația standard nu este mare, dar trebuie ținută cont de ea la interpretarea rezultatelor pentru teste diferite.

5.1.8. Îmbinarea fonem-alofon

In capitolul 5.1.4., s-a făcut o comparaţie între unităţile de vorbire şi anume foneme şi alofone, din punct de vedere ale constrângerilor pe care le întroduc şi ale rezultatelor care se obţin de consecinţă. Rezultatele experimentale arată că la folosirea alofonelor se obţine o rată de recunoaştere mai mare. Însă, comparând mulţimile de cuvinte recunoscute corect în cazul fonemelor şi în cazul alofonelor se observă că aceste mulţimi se intersectează dar mulţimea cuvintelor din cazul fonemelor nu este o submulţime a mulţimii cuvintelor din cazul alofonelor.

O analiză mai detaliată s-a făcut pentru o configuraţie cu 12 stări şi 3 componente de mixturi gaussiene cu rezultatele prezentate în Tabelul 5.11.

Histograma din Fig. 5.7. arată câte cuvinte au fost recunoscute în ambele cazuri şi câte doar într-unul din cele două cazuri. 79% din cuvinte au fost recunoscute în ambele cazuri, iar 5% din cuvinte nu au fost recunoscute în nici un caz. Aceste date arată că prin alegerea unui mod mai inteligent de a lua decizii există potenţial de creştere pentru rata de recunoaştere până la 95%.

Tabelul 5.11. Rezultatele obținute cu cele două tipuri de unități de vorbire

Configurație HMM		Informații de limbă		Algoritm de recunoastere		Rata de recunoaștere [%]
		Dimensiunea dicționarului	Restricții gramaticale	Tip	Descrierea etapelor	
12 stări 3 componente de mixturi gaussiene MFCC_E_D	modelare de foneme	10000 cuvinte	orice succesiune de cuvinte	recunoaștere într-un pas	recunoaște cuvinte	84.89
	modelare de alofone					88.59

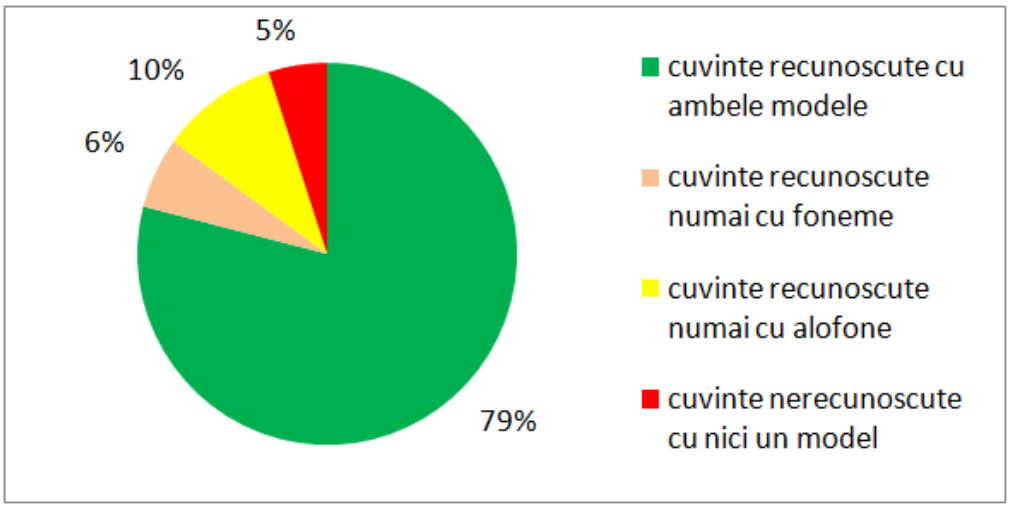

Fig. 5.7. Histograma cuvintelor recunoscute cu cele două tipuri de unități de vorbire.

Utilizând probabilitatea logaritmică de recunoaştere ca un criteriu decizie, putem construi următorii paşi pentru o decizie combinată fonem-alofon:

1. Se rulează un proces de recunoaştere cu foneme.
2. Se rulează un proces de recunoaştere cu alofone.
3. Se compară probabilitaţile logaritmice obţinute în cele două cazuri şi se alege soluţia cu cel mai bun scor.

O observaţie merită facută în legătură cu comparaţia directă a probabilităţilor logaritmice obţinute în cele două cazuri. Această comparaţie este posibilă doar în cazul în care folosim acelaşi număr de parametri şi de mixturi atât în cazul recunoaşterii cu foneme cât şi în cazul recunoaşterii cu alofone. Doar în acest caz, sumele pentru calculul probabilităţii logaritmice vor avea acelaşi număr de termeni, în celelalte cazuri acest număr fiind diferit. Dacă topologiile sunt diferite, se pot folosi ponderi, deşi alegerea ponderilor este dificilă.

Rezultatele obţinute astfel sunt prezentate în Tabelul 5.12. Creşterea absolută ratei de recunoaştere este semnificativă, cu 4.1% faţă de recunoaşterea cu alofone. Însă, această metodă nu a atins valoarea potenţială de 95% pentru rata de recunoaştere; ar mai fi 2% de

îmbunătăţit. Cu toate acestea, acest rezultat este important. El arată că având rezultate de recunoaştere provenind din surse diferite, se poate obţine o rată de recunoaştere mai mare prin alegerea unui mod inteligent de decizie.

Tabelul 5.12. Rezultatele recunoașterii prin îmbinarea fonem-alofon

Modele HMM	Informații de limbă		Algoritm de recunoastere		Rata de recunoaștere [%]
	Dimensiunea dicționarului	Restricții gramaticale	Tip	Descrierea etapelor	
foneme	10000 cuvinte	orice succesiune de cuvinte	recunoaștere într-un pas	recunoaște cuvinte	84.89
alofone					88.59
Combinate fonem-alofon					92.72

5.2. OPTIMIZĂRI PENTRU CREŞTEREA VITEZEI DE RULARE

5.2.1 Factori care determină viteza de rulare a algoritmului de recunoaștere

Recunoaşterea limbajului vorbit este un proces care necesită o putere de procesare mare. Ea se bazează pe algoritmul de căutare Viterbi explicat în capitolul 2.2.5. În experimente s-a folosit o variantă particulară al acestui algoritm numită Pasarea Jetonului (Token Passing [212]). În această variantă a algoritmului, se calculează, la fiecare nod, scorurile parţiale ale drumurilor numite jetoane (tokens). Există câteva caracteristici ale sistemului care afectează timpul de prelucrare. Aceste caracteristici sunt:

- Numărul de modele – un număr mai mare de modele implică o reţea mai mare de noduri (proporţională cu numărul de modele) şi implicit un număr mai mare de jetoane.
- Numărul de stări - un număr mai mare de stări implică o reţea mai mare de noduri (proporţională cu numărul de stări) şi implicit un număr mai mare de jetoane.
- Numărul de parametri - numărul de operaţii aritmetice (în mod principal înmulţiri) care se fac la fiecare nod (stare) este proporţional cu numărul de parametri.
- Numărul de componente ale mixturi gaussiene – numărul de operaţii aritmetice (în mod principal înmulţiri) care se fac la fiecare nod (stare) este proporţional cu numărul de mixturi.
- Numărul de jetoane – un număr mare de jetoane implică un număr mare de comparaţii la fiecare nod.
- Dimensiunea dicţionarului (numărul de intrări ale dicţionarului) – Un număr mare de intrări înseamnă un număr mare de combinaţii ale succesiunilor de modele şi de consecinţă şi un număr mai mare jetoane.
- Complexitatea restricţiilor gramaticale – O complexitate ridicată duce la micşorarea numărului de combinaţii ale succesiunilor de modele şi de consecinţă la un număr mai

mic de jetoane. Trebuie remarcat că o complexitate ridicată ne îndepărtează de vorbirea spontană, unde vorbitorul nu vorbește strict după reguli strict gramaticale.

O dată aleasă unitatea de limbă (fonem, alofon, etc.), numărul de modele nu se poate modifica, de exemplu, în cazul fonemelor din limba română acest număr este 34. Numărul de alofone (7372) fiind mai mare decât numărul de foneme (34), implică o rețea ma mare de noduri și de consecință și o viteză de rulare mai scăzută. Pe de altă parte, alofonele introduc restricții suplimentare ceea ce duce la reducerea numărului de jetoane.

La căutarea configurației optime, se variază numărul de stări, de parametrii de voce și de componente ale mixturii gaussiene. Cu toate acestea, dacă la marirea numărului de stări, de parametrii vocali sau de componente ale mixturii gaussiene nu se obține o creștere semnificativă a ratei de recunoaștere, atunci, funcție de aplicație, se poate sacrifica creșterea (ușoară) a ratei de recunoaștere în avantajul vitezei de rulare.

5.2.2. Variația lărgimii fascicolului de căutare

Așa cum indică algoritmul Token Passing, la fiecare nod se calculează scorurile (probabilitățile logaritmice) jetoanelor care provin din nodurile cadrului anterior și se alege jetonul cu scorul cel mai mare. Prin această metodă, de la un nod continuă doar jetonul cu scorul cel mai mare, iar restul sunt eliminate. La fiecare cadru, avansează un număr de jetoane egal cu numărul total de stări. Cu toate acestea, rămân "în joc" și multe jetoane cu scoruri mici. Principiul fascicolului de căutare constă în comparația jetoanelor "câștigătoare" pentru un cadru și păstrarea jetoanelor al căror scor nu este mai mic decât un prag. Pragul este stabilit ca fiind valoarea scorului maxim pentru cadrul respectiv minus o valoare configurabilă care se numește lărgimea fascicolului. Astfel, vor rămâne doar jetoane cu scoruri apropiate cu primul, scăzând semnificativ timpul de rulare al procesului de decodare. În Tabelul 5.13., sunt prezentate rezultatele recunoașterii și timpii de rulare pentru diferite valori ale fascicolului. Durata medie al unui fișier este 1.65s.

Tabelul 5.13. Variația ratei de recunoaștere și a timpului de rulare funcție de lărgimea fasciculului

Lărgimea fascicolului	Rata de recunoaștere [%]	Timpul de rulare per fișier [s]
Fără fascicul	73.21	52.3
250	72.41	15.4
200	72.97	14.8
150	72.43	10.7
140	72.12	8.5
130	71.58	7.7
120	70.61	6.9
110	69.43	6.5
100	66.41	6.1

Rezultatele au fost obținute pentru un sistem cu 6 stări, 2 mixturi și MFCC_E_D. După cum se observă, prin introducerea fascicolului, timpul de rulare scade semnificativ până la un ordin de mărime. În schimb, rata de recunoaștere scade puțin pentru valori mari a

lărgimii de fascicul şi sensibil pentru valori mici a lărgimii de fascicol. Cauza acestei scăderi se datorează faptului că unele jetoane corecte care nu obţin scoruri suficiente în primele cadre sunt eliminate. Funcție de aplicație se poate alege un compromis între viteza de rulare şi rata de recunoaştere. De exemplu, pentru experimentele din capitolul 5.1. s-a ales valoarea de 150 pentru lărgimea fasciculului.

5.2.3. Reducerea timpului de recunoaştere prin introducerea restricţiilor gramaticale şi reducerea dimensiunii dictionarului

Restricţiile gramaticale şi reducerea dicţionarului duc la micşorarea numărului de jetoane. Cu cât este mai mică lista de cuvinte din care se va decide cu atât mai puţine vor fi jetoanele de comparat la fiecare nod. Din acest motiv recunoaşterea la nivel de fonem este mult mai rapidă decât recunoaşterea la nivel de cuvânt, dar nu beneficiază de avantajul constrângerilor pentru o rată mai bună de recunoaştere [48].

Rezultatele prezentate în Tabelul 5.14. sunt obţinute pentru un sistem cu 12 stări, 3 mixturi şi MFCC_E_D. Restricţiile folosite sunt cele explicate în capitolul 5.1.5. şi anume gramatică de tip *linişte-cuvânt-linişte*, şi reducerea dicţionarului de la 10000 la 1000 de cuvinte.

Tabelul 5.14. Variația ratei de recunoaștere și a timpului de rulare funție de mărimea dicționarului și de restricții gramaticale

Configurație HMM	Informații de limbă		Algoritm de recunoastere		Rata de recunoaștere [%]	Timpul mediu de recunoaștere per fișier [s]
	Dimensiunea dicționarului	Restricții gramaticale	Tip	Descrierea etapelor		
foneme	10000 cuvinte	orice succesiune de cuvinte	recunoaștere într-un pas	recunoaște cuvinte	84.89	5.34
alofone					88.59	2.92
foneme	10000 cuvinte	liniște-cuvânt-liniște	recunoaștere într-un pas	recunoaște cuvinte	91.53	1.45
alofone					89.01	1.17
foneme	1000 cuvinte	orice succesiune de cuvinte	recunoaștere într-un pas	recunoaște cuvinte	96.24	0.31
alofone					92.92	0.25

Din rezultatele obţinute se observă că atât introducerea restciţiilor cât şi reducerea dicţionarului aduc o creştere a ratei de recunoaştere şi o micşorare a timpului de rulare. Acelaşi rezultat se obţine fie alegând fonemele ca unitate de limbă pentru modelare, fie alegând alofonele. Cu toate acestea, reducerea dicţionarului nu este întotdeauna aplicabilă, de aceea trebuie găsită o metodă care să reducă adaptiv dimensiunea dicţionarului fără a fi nevoie de a renunţa definitiv şi arbitrar la o mulţime de cuvinte.

5.2.4. Recunoaşterea limbajului vorbit în trei paşi. Reducerea adaptivă a dicţionarului fonetic.

În capitolul 5.2.3, s-a arătat că reducerea dicţionarului fonetic reduce semnificativ timpul de rulare şi creşte acurateţea de recunoaştere. Reducerea dicţionarului se face de obicei la începutul sesiunii de recunoaştere prin alegerea domeniului de vorbire [39], [82]. Însă, odată ales domeniul, dimensiunea dicţionarului nu se mai poate modifica.

Recunoaşterea limbajului vorbit în trei paşi este o metodă propusă cu scopul de a rezolva problema ridicată în capitolul 5.1.5., adică să reducă adaptiv dicţionarul fără a se renunţa la vreo submulţime de cuvinte. Metoda constă în următorii paşi:

1. Recunoaştere la nivel de fonem. Acest proces este rapid pentru că reţeaua de noduri devine simplă implicând astfel un număr mic de jetoane. Rezultatul acestui pas este obţinerea unei succesiuni de foneme împreună cu probabilităţile de recunoaştere corespunzătoare. Dintre fonemele recunoscute, trebuie ales fonemul sau fonemele care au cea mai mare probabilitate de recunoaştere.
2. Alegerea unui dicţionar de dimensiune mai mică pe baza fonemelor cu probabilitatea de recunoaştere cea mai mare. Se identifică fonemele cu rata de recunoaştere cea mai mare şi se alege un dicţionar cu cuvintele care conţin doar acele foneme. Acest dicţionar, fie este creat în timpul rulării fie se creează dinainte o bibliotecă de dicţionare, fiecare conţinând cuvinte cu un anumit fonem.
3. Recunoaştere la nivel de cuvânt folosind dicţionarul obţinut la pasul 2. Acest dicţionar are dimensiune mai mică decât dicţionarul iniţial şi are probabilitate mare să conţină cuvântul corect.

La pasul 1, trebuie găsit fonemul care are cea mai mare probabilitate să fie recunoscut corect. Această decizie este importantă pentru că pe baza ei se va alege dicţionarul doar cu cuvinte care conţin acest fonem. O alegere greşită a dicţionarului la acest pas poate face ca în mulţimea cuvintelor dicţionarului ales să nu se găsească cuvântul corect. Nu întotdeauna probabilitatea de recunoaştere a fonemului este cel mai bun criteriu pentru a alege fonemul cel mai de încredere. Probabilitatea de recunoaştere este în fond un scor care depinde mult de modul în care au fost construite modelele, de baza de date, etc. În primul rând, scorurile trebuie normate la numărul de cadre. Apoi, o indicaţie importantă este probabilitatea de recunoaştere medie pentru un fonem.

Pentru a obţine probabilităţile de recunoaştere medii pentru foneme, se face o recunoaştere la nivel de fonem pentru toată baza de date (de antrenare şi de test). Fiindcă nu toate rezultatele recunoaşterii sunt corecte s-a folosit o aliniere a fonemelor recunoscute cu cele corecte prin ajutorul algoritmului DTW. S-au luat în considerare doar fonemele recunoscute corect cu probabilităţile lor de recunoaştere şi s-a calculat pentru fiecare fonem probabilitatea de recunoaştere medie (normată la numărul de cadre). Valorile probabilităţilor de recunoaştere medii pentru câteva dintre foneme sunt prezentate în Tabelul 5.15.

Tabelul 5.15. Probabilitățile de recunoaștere medii per cadru obținute pentru câteva foneme

Fonem	a	b	d	e
Probabilitatea logaritmică medie ($\overline{recProb}$)	-54.54	-58.06	-60.74	-55.49

Se observă că probabilităţile logaritmice medii variază mult de la fonem la fonem. Din acest motiv în loc să folosim probabilitatea de recunoaștere ca și criteriu pentru alegerea celui mai „sigur" fonem, folosim probabilitatea de recunoaștere normată la probabilitatea de recunoaștere medie calculată cu formula 5.2.:

$$recTrust = recProb - \overline{recProb} \tag{5.2}$$

unde *recTrust* este scorul de încredere, *recProb* este probabilitatea de recunoaștere iar $\overline{recProb}$ este probabilitate de recunoaștere medie. Un *recTrust* pozitiv înseamnă că fonemul este recunoscut cu încredere mare. Aşadar, rezultatul primului pas va fi o succesiune de foneme recunoscute şi sortate dupa *recTrust.*

La pasul 2, se alege un dicţionar care conţine doar cuvintele care au în componența lor cele mai de încredere foneme rezultate la pasul 1. Această sarcină, însă, nu este uşoară pentru că se pune problema numărului de foneme care trebuie luat în considerare la alegerea dicționarului. Folosind următoarele notații: *f1*, *f2* şi *f3* pentru fonemele cu cel mai mare *recTrust*, putem construi următoarele dicționare:

- Numai cuvinte cu f1
- Numai cuvinte cu f1 şi f2
- Numai cuvinte cu f1 şi f1 şi f3
- Numai cuvinte cu f1 sau f2
- Numai cuvinte cu f1 sau f2 sau f3
- Altă logică

Pentru a evalua care tip de dicţionar este cel mai potrivit am sumarizat câteva caracteristici ale dicţionarelor în Tabelul 5.16. Dimensiunea dicţionarului pentru un anumit tip este o medie ponderată a dimensiunii tuturor dicţionarelor care aparţin acelui tip. Ponderile sunt calculate ca raportul între numărul de cuvinte pentru care s-a folosit dicţionarul şi numărul total de cuvinte. Se calculează şi rata de alegere corectă a dicţionarului în procente. Se consideră că un dicţionar s-a ales corect dacă conţine cuvântul corect.

Din aceste rezultate se observă că dimensiunea dicţionarului devine din ce în ce mai mare cu cât restricţiile sunt mai puţin dure (ceea ce era de aşteptat). Dimensiunea medie a dicţionarului în fiecare caz trebuie comparată cu cea a dicţionarului iniţial care este de 10000 de cuvinte. Numai ultimile două reguli au o rată de alegere corectă a dicţionarului acceptabilă. Pentru experimente s-a folosit ultima regulă din tabel. S-a observat că pentru fonemele {a, e, i, i3, l, o, ş, u, z} rata de alegere corectă a dicţionarului este foarte mare (aproape de 100%), de aceea, dacă cel mai sigur fonem recunoscut este din această mulţime nu mai folosim fonemul *f2*. În caz contrar, se foloseşte şi *f2*.

Tabelul 5.16. Dimensiunea dicționarelor și rate de selecție corectă a dicționarului

Logica de selecție a dicționarului	Dimensiunea medie a dicționarului [nr. cuvinte]	Rata de selecție corectă a dicționarului [%]
f1 & f2 & f3	215	63.70
f1 & f2	n/a	74.14
f1	3460	86.62
f1 | f2	5470	97.97
f1 dacă f1 ∈ {a, e, i, i3, l, o, s1, u, z}, f1 | f2 în rest	4756	96.84

La pasul 3, se rulează un proces de recunoaştere de cuvinte folosind dicţionarul ales la pasul 2. Rezultatele obţinute sunt prezentate în Tabelul 5.17.

Tabelul 5.17. Rata de recunoaștere și viteza de rulare pentru metoda de selecție adaptivă a dicționarului

Modele HMM	Informații de limbă		Algoritm de recunoaștere		Rata de recunoaștere [%]	Timpul mediu de recunoaștere per fișier [s]
	Dimensiunea dicționarului	Restricții gramaticale	Tip	Descrierea etapelor		
foneme	10000 cuvinte	un singur cuvânt	recunoaștere într-un pas	recunoaște cuvinte	91.53	1.45
alofone					89.01	1.17
foneme	Pas 1: 41 phones Pas 3: average of 4756 cuvinte	Pas 1: orice combinație de foneme Pas 3: un singur cuvânt	Recunoaștere în trei pași	Pas 1: recunoaște foneme Pas 2: alege dicționarul Pas 3: recunoaște cuvinte	89.36	0.91
alofone					86.82	0.56

Se observă că la înjumătăţirea dimensiunii dicţionarului, timpul de rulare a procesului de recunoaştere scade semnificativ: 37% în cazul recunoaşterii de vorbire cu foneme şi 52% în cazul recunoaşterii de vorbire cu alofone. Timpul de recunoaştere de sub 1s pentru un fişier de durată medie de 1.65s, nu numai că face ca procesul să fie de timp real, dar lasă spaţiu şi pentru prelucrări suplimentare care cresc rata de recunoaştere (de ex. implementarea unui modul de reducere a zgomotului). Pe de altă parte, rata de recunoaştere scade şi ea cu o valoare absolută de 2.18%. Această scădere este din cauza ratei de alegere corectă a dicţionarului, care nu este 100%. Acest reuzultat este, totuşi, important pentru că rata de alegere a dicţionarului se poate îmbunăţi printr-un studiu al distribuţiei probabilităţilor de recunoaştere la calculul scorului de încredere (aici s-a folosit numai rata de recunoaştere medie).

5.2.5. Recunoaşterea limbajului vorbit în trei paşi pentru un sistem care foloseşte îmbinarea fonem-alofon

Metoda folosită în capitolul 5.1.8 a adus o creştere a ratei de recunoaştere iar metoda folosită în capitolul 5.2.4 a adus o scădere a duratei de rulare. Combinând cele două metode putem beneficia de ambele avantaje. Tabelul 5.18. prezintă rezultatele obţinute.

Tabelul 5.18. Rezultatele obținute prin selecția adaptivă a dicționarului și îmbinarea fonem-alofon

<table>
<tr><td rowspan="2">Modele HMM</td><td colspan="2">Informații de limbă</td><td colspan="2">Algoritm de recunoaștere</td><td rowspan="2">Rata de recunoaștere [%]</td><td rowspan="2">Timpul mediu de recunoaștere per fișier [s]</td></tr>
<tr><td>Dimensiunea dicționarului</td><td>Restricții gramaticale</td><td>Tip</td><td>Descrierea etapelor</td></tr>
<tr><td>foneme</td><td rowspan="3">Pas 1: 41foneme
Pas 3: în medie 4756 cuvinte</td><td rowspan="3">Pas 1: any combination of any phones
Pas 3: only one word</td><td rowspan="3">recunoaștere în trei pași</td><td rowspan="3">Pas 1: recunoaște foneme
Pas 2: alege dicționarul
Pas 3: recunoaște cuvinte</td><td>89.36</td><td>0.91</td></tr>
<tr><td>alofone</td><td>86.82</td><td>0.56</td></tr>
<tr><td>Combinate fonem-alofon</td><td>91.95</td><td>1.56</td></tr>
</table>

Rata de recunoaştere este îmbunătăţită cu o valoare absolută de 2.69% faţă de metoda în trei paşi din capitolul 5.2.4, iar durata medie de recunoaştere a unui fonem creşte la valoarea 1.56s. Astfel, am crescut rata de recunoaştere şi am menţinut caracterul de prelucrare în timp real al procesului de recunoaştere.

CAPITOLUL 6

METODE DE OPTIMIZARE BAZATE PE INCERTITUDINE ȘI RECUNOAȘTEREA LIMBAJULUI VORBIT ÎN CONDIȚII ADVERSE

6.1. INCERTITUDINEA MODELULUI ACUSTIC ÎN RECUNOAŞTEREA LIMBAJULUI VORBIT

Recunoaşterea limbajului vorbit este, în fond, un proces de luare de decizii şi ca în orice asemenea proces deciziile se iau pe baza unor criterii. În RLV, criteriile de decizie sunt bazate pe parametri de semnal vocal (LPC, MFCC, PLP, etc.). Din procesul de antrenare, rezultă nişte modele acustice (sub forma de HMM) care încearcă să modeleze evoluţia în timp a parametrilor pentru unităţile de limbă (foneme, alofone, etc.). În mod ideal, dacă distribuţiile parametrilor nu se suprapun atunci criteriile de decizie sunt în stare să discearnă perfect între ipoteze. În realitate, distribuţiile parametrilor pentru diferite unităţi de limbă se suprapun din mai multe motive. În primul rând, diferite unităţi de limbă au în comun părţi ale spectrelor lor. Alţi factori care duc la suprapunere sunt erorile din baza de date de antrenare şi imperfecţiuni de aliniere în timpul antrenării (*embedded* cu algoritmul Baum-Welch). Această suprapunere a distribuţiilor parametrilor duce la incertitudine în procesul de decizie. În acest capitol, se studiază în detalii incertitudinea şi se scoate în evidenţă faptul că nu toţi parametrii contribuie la fel în procesul de decizie, iar contribuţia unora este nesemnificativă. În urma acestei analize, contribuţia parametrilor este ponderată, iar dacă contribuţia unuia dintre ei este nesemnificativă, acel parametru nu este luat în calcul reducând astfel semnificativ numărul de calcule în procesul de recunoaştere.

6.1.1. Distribuţia parametrilor de voce şi modelarea lor prin HMM

Despre modul de calcul al parametrilor de voce s-a discutat în capitolul 2.3., iar despre utilizarea lor în cadrul HMM s-a discutat în capitolul 2.2. Pentru a scoate în evidenţă contribuţia unui singur parametru în procesul de decizie, vom presupune iniţial că recunoașterea se face după un singur parametru, de exemplu energia. Pentru o secvență de semnal vocal corespunzătoare unei unități de limbă, evoluția energiei în timp poate arăta ca

cea prezentată în Fig. 6.1. Aceasta, însă, reprezintă evoluția parametrului pentru o realizare particulară a unității de limbă. Dacă luăm în considerare mai multe eșantioane ale aceleiași unitate de limbă, vom observa că liniile realizărilor particulare vor ocupa o anumită arie în planul energie-timp cu o anumită densitate (Fig. 6.2.).

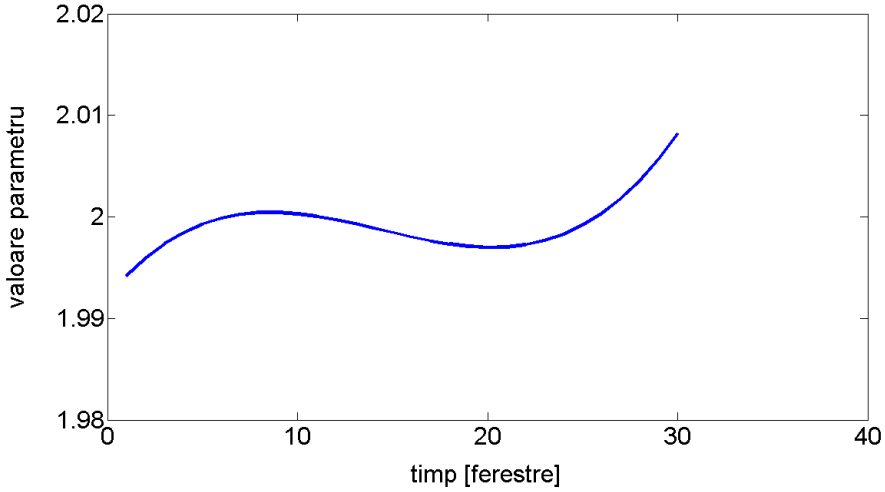

Fig. 6.1. Variația unui parametru în timp pentru o realizare particulară

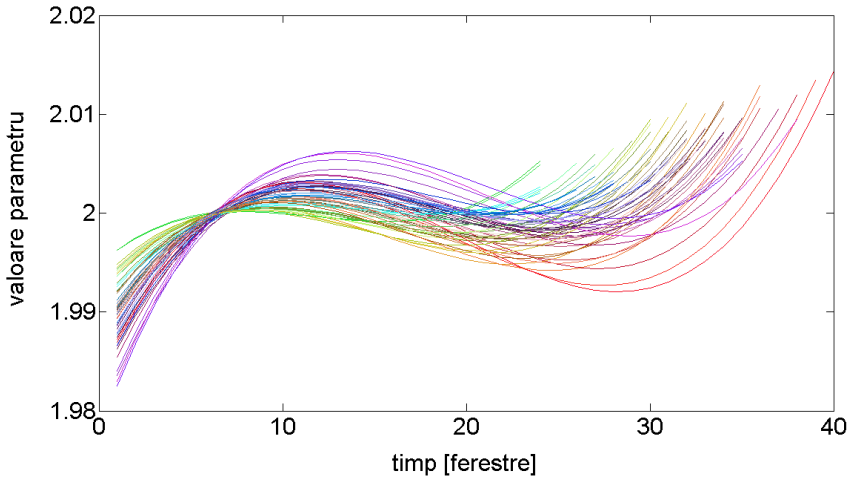

Fig. 6.2. Variația unui parametru în timp pentru mai multe realizări particulare

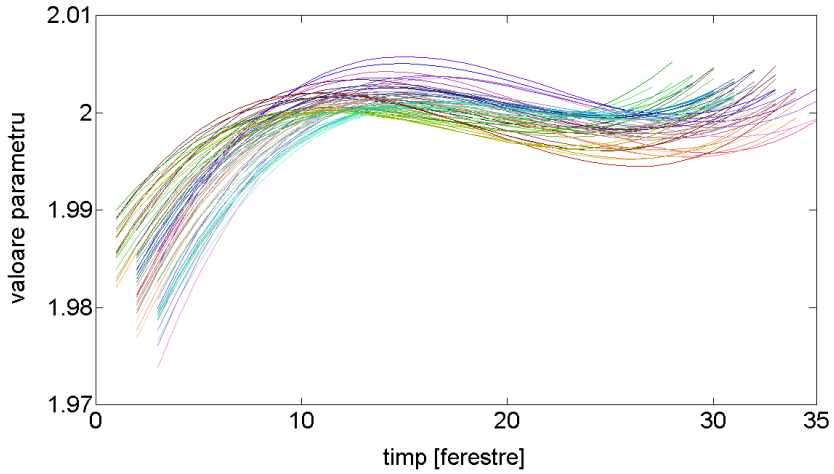

Fig. 6.3. Alinierea în timp a realizărilor particulare pentru un parametru

În contextul HMM-urilor, valoarea energiei pentru un cadru este ieşirea generată de una dintre stările HMM-ului. Pentru construirea unor modele mai bune, cadrele cu valori similare trebuie considerate ca fiind ieşirea aceleiaşi stări. Pentru a realiza această "grupare", este nevoie de o aliniere a tuturor realizărilor ca în Fig. 6.3. Unul dintre criteriile de aliniere poate fi condiţia ca liniile evoluţiilor să ocupe aria minimă. În algoritmul Baum-Welch (vezi capitolul 2.2.7.), criteriul de aliniere este maximizarea probabilităţii logaritmice globale. Odată făcută alinierea, se pot calcula distribuţiile parametrilor (în exemplul de faţă, distribuţia energiei), pentru fiecare cadru. Astfel, se obţine evoluţia distribuţiei în timp cum este prezentat în Fig. 6.4. [35].

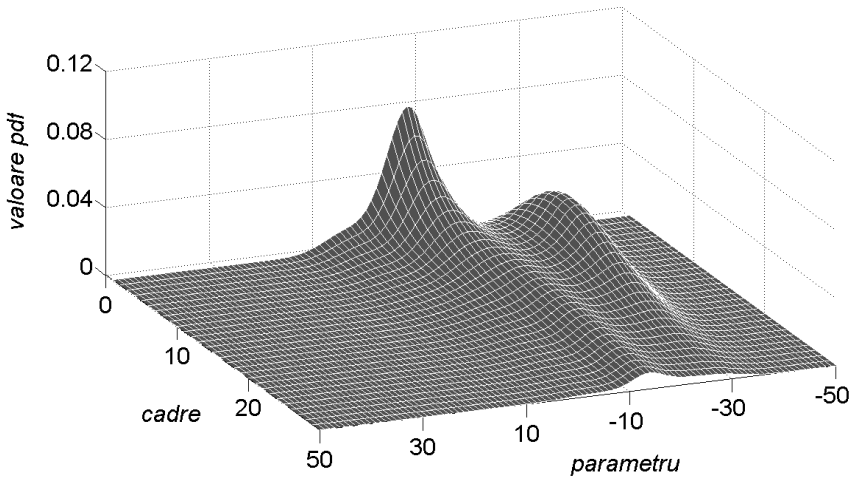

Fig. 6.4. Evoluția în timp a distribuției unui parametru de voce

Aceste distribuţii sunt calculate pentru toate modelele. Prin măsurarea suprapunerii distribuţiilor între două modele, se poate obţine o măsură care să ne dea o indicaţie despre cât

de mult un criteriu (parametru) poate diferienţia modelele (Fig. 6.5.). Vom defini această măsură ca fiind *Incertitudine Acustică a Modelelor* (IAM).

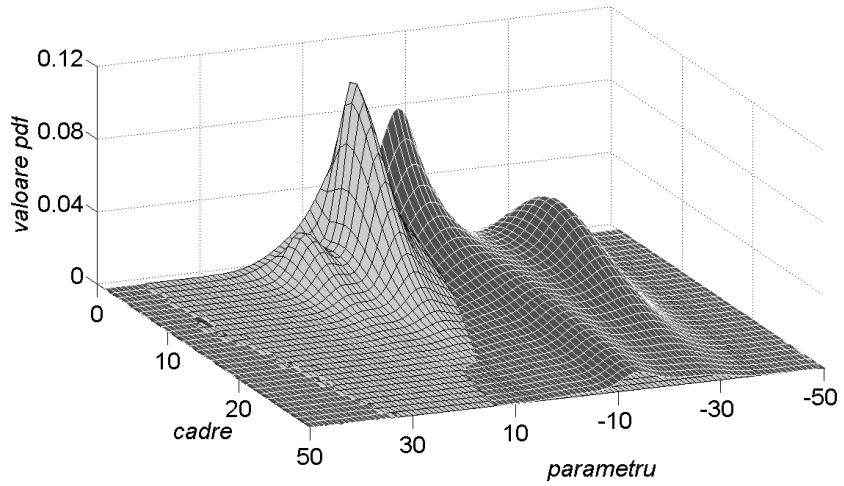

Fig. 6.5. Suprapunerea evoluțiilor distribuțiilor unui parametru pentru două modele diferite. *pdf* reprezintă funcția de densitate de probabilitate (din engleză, Probability Density Function).

Luând în calcul că în procesul de decizie participă *n* parametrii, un model va ocupa un anumit volum în spaţiul *n+1* dimensional (*n* parametrii + timp). Ideal, modelele antrenate n-ar trebui să se suprapună, dar acest lucru nu este posibil pentru că fonemele însele au părţi comune în spectrele lor. În plus, vom vedea că o pereche de modele se suprapun mai mult decât o altă pereche de modele şi că nu toate distribuţiile parametrilor se suprapun la fel.

Aşa cum s-a explicat în capitolul 2.2.3., decizia în procesul de recunoaştere cu HMM-uri se face pe baza probabilităţilor logaritmice calculate cu formula (6.1):

$$\log\left[P(O \mid M)\right] = \log\left\{\max_{x}\left[a_{x(0)x(1)}\prod_{t}^{T} b_{x(t)}(O_t)a_{x(t)x(t+1)}\right]\right\} \tag{6.1}$$

unde $P(O|M)$ este probabilitatea ca modelul M să fi generat vectorul de observaţii O. a_{ij} este probabilitatea de tranziţie din starea i în starea j, $b_{x(t)}(O_t)$ este scorul obţinut de starea x la timpul (cadrul) t pentru observaţia O_t şi T este numărul total de cadre.

Scorul pe care starea j îl obţine la fiecare cadru este dat de formula (6.2):

$$b_j(O_t) = \sum_{g=1}^{G} w_{jg} N(O_t, \mu_{jg}, \Sigma_{jg}) \tag{6.2}$$

unde:

$$N(O, \mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} e^{-\frac{1}{2}(O-\mu)'\Sigma^{-1}(O-\mu)} \tag{6.3}$$

$O=[o_1,o_2,..,o_n]$' este vectorul de observaţie n-dimensional, n este numărul de parametrii de voce pe care ii extragem din observaţii, $|\sum|$ este determinantul matricii diagonale de

covarianţă, G este numărul de component ale mixturii şi w_{jg} este ponderea componentei g ale mixturii stării j. Înlocuind vectorul O cu componenţii lui, forumula (6.3) devine:

$$N(O,\mu,\Sigma)=\frac{1}{\sqrt{(2\pi)^n\sigma_1^2\sigma_2^2...\sigma_n^2}}e^{-\frac{1}{2}\left[\frac{(o_1-\mu_1)^2}{\sigma_1^2}+\frac{(o_2-\mu_2)^2}{\sigma_2^2}+...+\frac{(o_n-\mu_n)^2}{\sigma_n^2}\right]} \tag{6.4}$$

Cum la calculul probabilităţii se foloseşte logaritmul, formula (6.4) se mai poate scrie:

$$\log N(O,\mu,\Sigma)=C-\frac{1}{2}\sum_i^n\frac{(o_i-\mu_i)^2}{\sigma_i^2} \tag{6.5}$$

unde:

$$C=-\frac{1}{2}\ln(2\pi)^N|\Sigma| \tag{6.6}$$

Cum arată (6.5), toţi parametrii au aceeaşi pondere în scorul final, deşi nu toţi contribuie la fel la discriminarea între foneme. În continuare, cu ajutorul IAM vom încerca calcularea unor ponderi care să mărească contribuţia parametrilor relevanţi în avantajul parametrilor mai puţin relevanţi.

6.1.2. Tipuri de IAM

IAM se poate clasifica în următoarele tipuri funcţie de cauzele care o produc [35]:

1. Tipul 1: Când un parametru ia valori similare pentru două modele şi, în consecinţă, cele două distribuţii se suprapun chiar dacă ele sunt antrenate bine (Fig. 6.6.). Din această cauză, dacă în timpul decodării parametrul ia valori din zona de suprapunere, modelele vor obţine scoruri similiare.
2. Tipul 2: Când un parametru, pentru un anumit model, ia valori în toată plaja (Fig. 6.7.). Aceasta înseamnă că acest parametru nu caracterizează modelul respectiv, şi, în consecinţă, scorul pe care îl obţine modelul la fiecare cadru nu reflectă realitatea.
3. Tipul 3: Când valoarea observaţiei curente nu este în niciuna dintre plajele celor două distribuţii (Fig. 6.8.). În acest caz, distribuţia care se află mai aproape de valoarea observaţiei va avea un scor mai bun decât cealaltă. Din acest motiv, ar fi rezonabil ca ambele modele să obţină acelaşi scor (mic).

Această clasificare presupune că toate modelele sunt bine antrenate şi că observaţiile nu sunt degradate. Orice degradare în timpul antrenării sau al decodării, însă, poate duce la una din situaţiile de mai sus. De exemplu, dacă baza de date de antrenare conţine erori, distribuţia unui parametru poate acoperi toată plaja de valori ca şi în cazul IAM de tip 2 (Fig. 6.7.). În timpul decodării, nu contează dacă erorile din baza de date constituie cauza acestei distribuţii sau pur şi simplu parametrul ia valori din toată plaja; efectul va fi că acest parametru (în calitate de criteriu de decizie) nu va duce la decizii bune şi, în consecinţă, ar trebui să aibă o pondere mai mică. O distorsiune a vectorului de observaţie poate duce la o IAM de tipul 1 prin plasarea parametrului în zona de suprapunere (Fig. 6.6.), sau poate duce la o IAM de tip 3 prin plasara parametrului în afara celor două distribuţii.

În continuare, fiecare tip de IAM se va trata separat. Scopul principal este acela de a-i identifica pe cei mai relevanţi parametri şi de a cuantifica relevanţa lor. Apoi, fie se vor aplica

ponderi funcție de relevanța parametrilor cu scopul de a mări rata de recunoaștere, fie se pot elimina paremetrii cei mai puțin relevanți cu scopul de a micșora numărul de calcule.

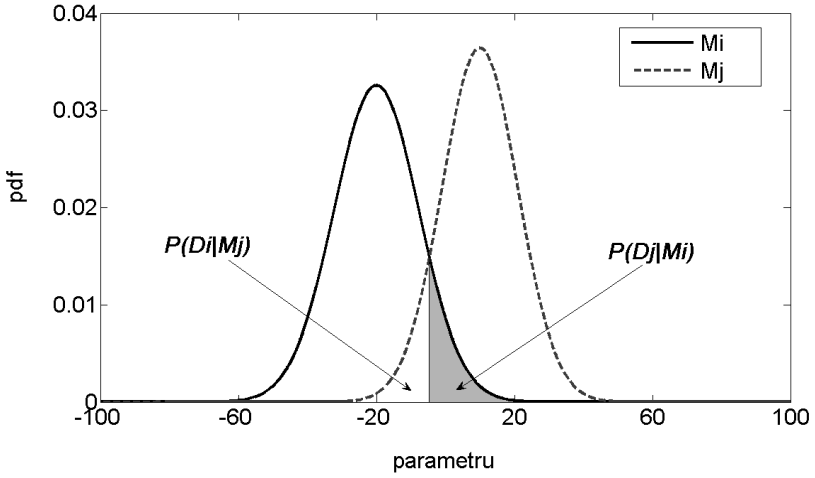

Fig. 6.6. IAM de tip 1 și probabilitatea de eroare în procesul de decizie. *Mi* și *Mj* sunt distribuțiile parametrilor pentru modelele *i* și respectiv *j*. Pe axa ordonatelor sunt valorile funcției de densitate de probabilitate. Zona în gri reprezintă *P(Dj|Mi)*, probabilitatea să se decidă greșit că modelul *j* este detectat în locul modelului *i*.

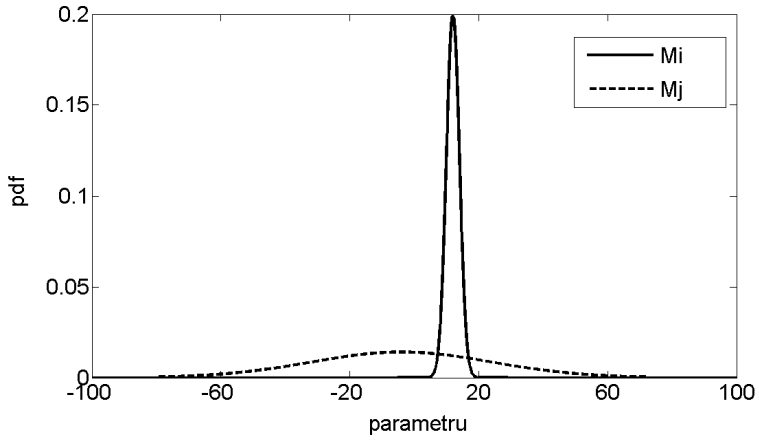

Fig. 6.7. IAM de tip 2. *Mi* și *Mj* sunt distribuțiile parametrului pentru modelele *i* și respectiv *j*. Pe axa ordonatelor sunt valorile funcției de densitate de probabilitate.

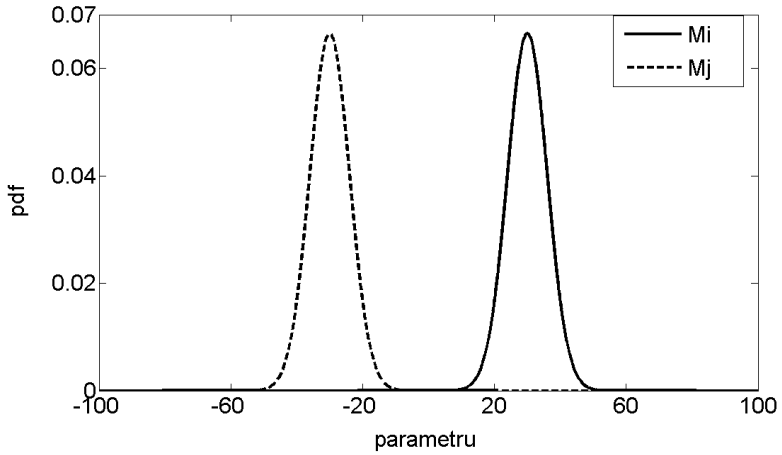

Fig. 6.8. IAM de tip 3. *Mi* și *Mj* sunt distribuțiile parametrului pentru modelele *i* și respectiv *j*. Pe axa ordonatelor sunt valorile funcției de densitate de probabilitate.

6.1.3. Utilizarea IAM de tip 1

Pentru a calcula IAM este nevoie să se construiască evoluțiile în timp ale distribuțiilor parametrilor. Aceste distribuții sunt calculate din HMM-uri deja antrenate cu algoritmul Baum-Welch. Apoi, evoluția unei distribuții se construiește prin înșiruirea distribuțiilor parametrului pentru fiecare cadru. Distribuția pentru un cadru (a modelului *i*) este calculată prin însumarea distribuțiilor tuturor stărilor, ponderate cu probabilitatea ca HMM să fie în acea stare la acel cadru (notată cu $p_s(t)$):

$$d_t^{(i)}(x) = \sum_{s=1}^{S} d_s^{(i)}(x) \cdot p_s(t) \tag{6.7}$$

unde $d_s^{(i)}(x)$ este funcția de densitate de probabilitate a ieșirii stări *s*; *S* reprezintă numărul total de stări. În cazul în care ieșirea unei stări este o mixtură gaussiană (GMM), funcția de densitate de probabilitate este calculată cu formula:

$$d_s^{(i)}(x) = \sum_{g=1}^{G} w_{sg} N(x, \mu, \sigma) \tag{6.8}$$

unde $N(x,\mu,\sigma)$ este distrbuția normal de medie μ și varianță σ^2, și w_{sg} este ponderea componentei *g* ale mixturii stării *s* (numărul de componente este *G*). Probabilitatea ca starea *s* a unui HMM să emită la cadrul *t* este calculată recursiv:

$$p_s(t) = \sum_{k=2}^{S} p_k(t-1) \cdot a_{ks} \tag{6.9}$$

unde a_{ks} este probabilitatea tranziției din starea *k* în starea *s*, cu inițializarea $p_k(1)=a_{1k}$. S-a ales ca prima și ultima stare ale HMM să fie non-emisive.

Odată construite evoluțiile distribuțiilor, IAM pentru un parametru dat se poate calcula prin însumarea suprapunerilor distribuțiilor modelelor la fiecare cadru. Așadar IAM între modelele *i* și *j* pentru criteriul *k* este dată de formula (6.10):

$$C_k^{(i,j)} = \sum_{t=1}^{T} C_{kt}^{(i,j)} \tag{6.10}$$

unde *T* este numărul de cadre pentru care IAM este calculat, și IAM $C_{kt}^{(i,j)}$ este calculat clasic ca în Fig. 6.6. Curbele prezentate în acest grafic sunt distribuțiile $d_t^{(i)}$ calculate cu formula (6.2). *P(Dj|Mi)* este probabilitatea de a decide eronat că modelul *j* este detectat în locul modelului *i*, adică probabilitatea de a confunda modelul *j* cu modelul *i*. Așadar, $C_{kt}^{(i,j)}$= *P*(*Dj*|*Mi*).

Trebuie observat că *T* în formula (6.10) se alege arbitrar. Evoluția distribuțiilor se întinde până la infinit, dar datorită scăderii exponențiale ale $p_s(t)$ (vezi formula (6.9)), ea se "stinge" după un anumit număr de cadre. O valoare rezonabilă pentru *T* este 50, așa că, ținând cont că o valoare obișnuită pentru durata unui cadru este de 10 ms, 50 de cadre sunt echivalente cu 500 ms.

IAM ($C_k^{(i,j)}$) pentru oricare două modele este calculată cu formula (6.10) pentru fiecare parametru. Aceasta înseamnă că dacă avem *M* modele și *n* parametrii, vor rezulta *n* tabele *M*x*M* cu valori de IAM cum este arătat în Tabelul 6.1. Tabelul 6.1. este matricea de confuzie clasică dacă decizia s-ar lua doar după parametrul *k*.

Tabelul 6.1. IAM între oricare două modele pentru un anumit criteriu *k*

IAM pentru parametrul *k*	Model 1	Model 2	...	Model *M*
Model 1	$C_k^{(1,1)}$	$C_k^{(1,2)}$	...	$C_k^{(1,M)}$
Model 2	$C_k^{(2,1)}$	$C_k^{(2,2)}$	...	$C_k^{(2,1)}$
...	...	...	...	...
Model *M*	$C_k^{(M,1)}$	$C_k^{(M,2)}$	...	$C_k^{(M,M)}$

Trebuie observat că $C_k^{(i,i)}$=1.

Parametrul *k* poate să nu discrimine bine între modelele *i* și *j*, dar poate să ajute la discriminarea între modelul *i* și un alt model *l*. Totodată, modelul *i* nu este confundat la fel cu toate modelele. Ținând cont de aceste observații, putem da ponderi mai mari parametrilor care ajută modelul *i* să fie discriminat de modelele cu care el este confundat cel mai des. Așadar, pe lângă o măsură a relevanței parametrilor, ne trebuie încă o măsură a confuziei între modele. O asemenea măsură se poate deduce fie din distribuții, prin ajutorul IAM, fie din rezultatele de recunoaștere prin numărarea cazurilor în care un fonem este ales în locul altuia.

În capitolul 6.1.1, s-a arătat cum IAM reprezintă un volum obținut prin suprepanurea distribuțiilor, dar acela este IAM produs doar de un singur parametru. IAM totală produsă de toți parametrii reprezintă un volum în spațiul *n*+1 dimensional și este calculat cu formula (6.11):

$$C^{(i,j)} = \prod_{k=1}^{N} C_k^{(i,j)} \tag{6.11}$$

IAM totală este produsul tuturor IAM produse de fiecare parametru, pentru că se consideră că parametrii sunt statistic independenți. Această presupunere este în line cu procesul de decodare unde și acolo parametrii sunt considerați independenți.

Astfel, se obține o matrice de confuzie ca cea din Tabelul 6.1., de data aceasta conținând IAM totală.

Algoritmul 1

Prima metodă constă în ponderarea criteriilor prin menținerea IAM totale la un nivel minim. Se calculează IAM medie a modelului *i* cu toate celelalte modele pentru fiecare criteriu *k*:

$$C_k^{(i)} = \frac{1}{M} \sum_{j=1}^{M} C_k^{(i,j)} \tag{6.12}$$

Deoarece parametrii cu IAM mai mică trebuie să aibă ponderi mai mari, ponderile pot fi invers proporționale cu IAM:

$$w_k^{(i)} = \frac{(C_k^{(i)})^{-1}}{\sum_{l=1}^{N} (C_l^{(i)})^{-1}} \tag{6.13}$$

Aici, numitorul este folosit pentru normare. $w_k^{(i)}$ sunt ponderile pentru parametrul *k* pentru modelul *i*. Modelul *i* poate fi foarte diferit de unele modele și să nu se confundă niciodată cu ele, dar el poate fi foarte apropiat de alte modele și se poate confunda des cu ele. Din acest punct de vedere, IAM medie în formula (6.12) este calculată prin însumarea valorilor din cele *m* modele (în loc de *M*) cu care modelul *i* se confundă cel mai des (*m* reprezintă numărul de modele cu care un model este confundat cel mai des). *m* este un parametru de ajustare și poate fi diferit pentru modele diferite, dar în testele acestei lucrări s-a folosit o valoare fixă pentru el.

Algoritmul 2

Acest algoritm este folosit numai atunci când se dorește eliminarea parametrilor mai puțin relevanți. El constă în următorii pași:

1. Inițializare

Stabilirea numărului *K* de parametri care trebuie scoși.

Golirea listei de parmetri eliminați.

Pentru i=1:*M* //pentru fiecare model.

Pentru l=1:K //pentru fiecare parametru care trebuie eliminat.

2. Găsirea max($C^{(i,j)}$) și indexul respectiv j.

3. Găsirea max($C_k^{(i,j)}$) și indexul respectiv k, pentru j calculat la pasul 2.

4. Introducerea lui k în lista parametrilor eliminați.

5. Recalcularea lui $C^{(i,j)}$ cu formula (6.11) excluzând (neînmulțind) factorii din lista parametrilor eliminați.

Scopul algoritmului este eliminarea parametrilor cei mai puțin relevanți. Aici, parametrii cei mai puțin relevanți sunt considerați aceia care introduc cea mai mare cantitate de IAM (pașii 2 și 3 ai algoritmului). După fiecare iterație, IAM medie este calculată excluzând parametrii deja eliminați (pasul 5 al algoritmului).

6.1.4. Utilizarea IAM de tip 1 ținând cont de confuziile între foneme

Această metodă nu înlocuiește **Algoritmul 1**, dar îl completează. Principiul care stă la baza acestei metode este acela de a obține o indicație din rezultatele recunoașterii despre cât de des se confundă două modele și de a construi, astfel, o matrice de confuzie. Aceasta este realizată prin analizarea rezultatelor de recunoaștere la nivel de fonem cu baza de date de antrenare și cea de testare. Se numără de câte ori modelul j a fost recunoscut în locul modelului i. Trebuie observat că numărul de confuzii calculat astfel depinde de baza de date aleasă pentru antrenare și de contextul în care se găsește unitatea de limbă (de exemplu, fonemul într-un cuvânt). Această informație este utilă pentru că de multe ori confuziile calculate astfel nu coincid cu cele calculate din distribuții (ca în capitolul 6.1.3.).

În experimentele aferente acestei lucrări, au fost folosite pentru modelare fonemele și, de aceea, trebuie aliniate cuvântul recunoscut cu cel original și trebuie văzut care fonem este recunoscut în mod eronat. Erorile în recunoaștera limbajului vorbit la nivel de fonem pot fi substituții, inserții și ștergeri. Aceste erori sunt numărate automat prin aplicarea algoritmului DTW așa cum e arătat în Fig. 6.9. Fonemele de pe axa ordonatei aparțin cuvântului recunoscut, iar fonemele de pe axa abscisei aparțin cuvântului original din înregistrare. Algoritmul caută drumul cu cel mai mic cost care își are originea în punctul (f1,f1) și se termină în punctul (fn,fm). Costurile sunt prezentate în Tabelul 6.2.

Costurile au fost alese astfel încât să poată acoperi toate tipurile de erori ale unui SRLV (substituții, inserții și ștergeri). Dacă nu sunt erori, drumul optimal este diagonala și costul lui este egal cu 0. Tranzițiile orizontale corespund ștergerilor. Tranzițiile verticale, care corespund inserțiilor, au cost mai mic decât cele orizontale pentru că aceasta înseamnă că două foneme au fost recunoscute în locul uneia. De multe ori, fonemele recunoscute sunt fonemul correct și încă unu, sau fonemul correct recunoscut de două ori.

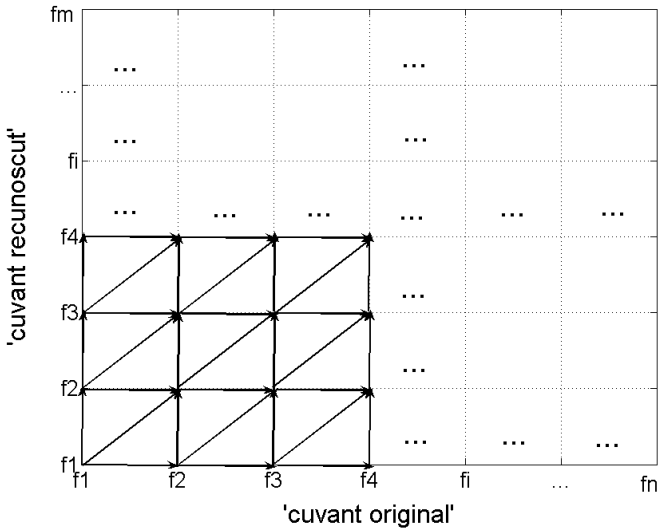

Fig. 6.9. Alinierea cuvintelor cu algoritmul DTW

Tabelul 6.2. Costurile algoritmului DTW

	Cost
Nod cu foneme identice	0
Nod cu foneme diferite	4
Tranziție diagonală	0
Tranziție orizontală	2
Tranziție verticală	1

După găsirea drumului cu cel mai mic cost, care înseamnă că s-a găsit cea mai bună aliniere, se analizează nodurile: dacă fonemele (de pe axele orizontale și verticale) nu sunt identice atunci contorul E_{ij} este incrementat, unde i este fonemul de pe axa orizontală (cel corect) iar j este fonemul de pe axa verticală (cel recunoscut). Aceste contoare sunt normate cu numărul total de aparații pentru fonemul i care se notează cu n_i:

$$e_{ij} = \frac{E_{ij}}{n_i} \tag{6.14}$$

După analizarea tuturor cuvintelor, tabelul este completat cu valorile e_{ij}.

Tabelul 6.3. Matricea de confuzie obținută din rezultatele recunoașterii

model \ confundat cu	Model 1	Model 2	...	Model *M*
Model 1	e_{11}	e_{12}	...	e_{1M}
Model 2	e_{21}	e_{22}	...	e_{2M}
...	...	...	...	...
Model *M*	e_{M1}	e_{M2}	...	e_{MM}

Algoritmul folosit la această metodă este acelaşi cu **Algoritmul 1**, dar în formula (6.12), se foloseşte media geometrică $\sqrt{C_k^{(i,j)} e_{ij}}$ în locul IAM $C_k^{(i,j)}$. La descrierea rezultatelor experimentale, ne vom referi la această metodă cu numele **Algoritm 3**. **Algoritmul 1** diferă de **Algoritmul 3** pentru că atunci când se calculează ponderile, el foloseşte doar distribuţiile antrenate ale modelelor. La **Algoritmul 3**, la calculul ponderilor contribuie şi confuziile obţinute din rezultatele de recunoaştere. S-a folosit media geometrică pentru că aceasta dă rezultate mai bune decât media aritmetică. Ne aşteptăm să avem valori similare pentru $C_k^{(i,j)}$ şi e_{ij}, dar în rezultatele recunoaşterii, unele modele se confundă mai des cu un model anume. Această situaţie nu reflectă realitatea, dar apare din cauza bazei de date şi a numărului limitat de contexte pentru un fonem. Media geometrică poate reduce acest efect pentru că în comparaţie cu media aritmetică, ea ia valori mai aproape de cel mai mic scalar.

6.1.5. Rezultate experimentale folosind IAM de tip 1

Experimentele au fost făcute cu baza de date 4 care conţine 10 000 de cuvinte, aşa cum este explicat în capitolul 4.1.2. HMM folosit are 12 stări unde prima şi ultima sunt non-emisive. Sunt permise doar tranziţii de la stânga la dreapta astfel încât de la o stare *j* nu se poate tranzita decât la stările *j*, *j*+1 şi *j*+2. Ieşirea unei stări este o mixtură de 3 distribuţii gaussiene. Parametrii de voce aleşi sunt MFCC (12 coeficienţi) incluzând energia şi derivatele de ordin I, pentru un total de 26 parametri. Procesul de antrenare este cel descris la capitolele 2.2.7. şi 4.2., iar pentru decodare s-a folosit algoritmul Token Passing (vezi capitolul 2.2.6.). Testele de recunoaştere au fost rulate în trei moduri:

- Fără Restricţii (FR) – Nici o restricţie nu s-a impus în legătură numărul de cuvinte sau ordinea lor.
- Gramatică Simplă (GS) – Se pleacă de la premisa că se ştie a priori că există doar un cuvânt în fişier. Aşadar, doar secvenţa linişte – cuvânt – linişte este permisă.
- Gramatică Simplă cu Dicţionar Redus (GS-DR) – Este aceeaşi gramatică folosită la GS dar dicţionarul este redus la 1000 de cuvinte (în loc de 10 000 folosite la FR şi GS).

Gramatica (restricţiile) sunt folosite pentru că în aplicaţii de interacţiune om-maşină bazate pe un singur cuvânt (de exemplu, comenzi), multe errori sunt cauzate de imperfecţiunea sistemului de recunoaştere de a împărţi un cuvânt compus în două cuvinte mai scurte. Prin aplicarea gramaticii ca în cazul GS se elimină aceste erori, care puteau fi eliminate şi prin aplicarea penalizării de inserţie. Reducerea dicţionarului se justifică prin existenţa unor aplicaţi pentru care nu este nevoie de un dicţionar mare (de exemplu, navigarea prin meniul unui dispozitiv mobil). În Tabelul 6.4., sunt prezentate valorile de referinţă pentru recunoaştere care se vor folosi la analiza rezultelor:

Tabelul 6.4. Ratele de erori fără aplicarea ponderilor

Modul de recunoaştere	WER [%]
FR	15.5
GS	8.22
GS-DR	2.47

Rata cuvintelor recunoscute eronat (WER – Word Error Rate) este măsurată ca raportul între numărul de cuvinte recunoscute greşit şi numărul total de cuvinte de test.

Două direcţii au fost folosite pentru teste: 1) ponderile au fost folosite pentru obţinerea unei rate de erori mai mică şi 2) eliminarea parametrilor (ponderi de 0 şi 1) s-a folosit pentru reducerea numărului de calcule, menţinând aceeaşi rată de erori.

Rezultatele obţinute prin aplicarea **Algoritmului 1** sunt prezentate în Tabelul 6.5., unde *m* reprezintă numărul de modele cu care un anumit model este confundat cel mai des:

Tabelul 6.5. Ratele de erori la folosirea Algoritmului 1

	WER [%]		
m	FR	GS	GS-DR
1	15.56	8.19	2.47
2	15.18	7.77	2.4
3	14.95	7.82	2.44
4	15.08	8.35	2.49
5	15.68	8.44	2.53

Rezultatele arată că există o valoare optimă pentru *m*. Aceasta era de aşteptat, pentru că ponderile sunt calculate din IAM a unui model cu modelele cu care el se confundă cel mai des. Pentru valori mici ale lui *m* sistemul va continua să confundă un anumit model cu modele similare care nu sunt incluse in *m,* iar pentru valori mari ale lui *m* sistemul nu mai este orientat să discrimine un model de modelele cu care se confundă cel mai des. Pentru FR valoarea optimă pentru *m* este 3, iar pentru GS şi GS-DR această valoare este 2, cea ce înseamnă că, în medie, un model este confundat cel mai mult cu alte 2 sau 3 modele. În aceste condiţii, este indicat ca ponderile pentru un model să fie calculate pe baza IAM lui cu 2-3 modele cu care el se confundă cel mai des. Îmbunătăţirea relativă este de 4.2%, care este o valoare încurajătoare, pentru că prin rafinarea funcţiei care calculează ponderile, se pot obţine rezultate şi mai bune.

Acelaşi algoritm este folosit şi pentru eliminarea parametrilor. Pentru fiecare gramatică, s-a ales din Tabelul 6.5. numărul *m* care a dat cele mai bune rezultate. În Tabelul 6.6., sunt prezentate rezultatele obținute pentru diferite valori ale numărului parametrilor eliminați (*K*):

Tabelul 6.6. Ratele de erori la aplicarea Algoritmului 1 pentru eliminarea parametrilor

	WER [%]		
Numărul (*K*) al parametrilor eliminați	FR (*m*=3)	GS (*m*=2)	GS-DR (*m*=2)
1	15.8	8.3	2.47
2	15.96	8.38	2.51
3	16.58	8.88	2.82
4	18.88	9.4	3.11
5	20.12	10.36	3.39

Rezultatele obţinute prin aplicarea **Algoritmului 2** sunt prezentate în Tabelul 6.7.:

Tabelul 6.7. Ratele de erori la aplicarea Algoritmului 2 pentru eliminarea parametrilor

	WER [%]		
Numărul (*K*) al parametrilor eliminați	FR	GS	GS-DR
1	15.6	8.22	2.47
2	15.94	8.38	2.49
3	16.08	8.56	2.58
4	16.22	9.1	2.74
5	17.47	10.3	3.02

Cu **Algoritmul 2** se obţin rezultate mai bune decât cu **Algoritmul 1**. Amândoi arată că pentru aceste modele toți parametrii contribuie la discriminarea modelelor pentru că ratele de erori cresc uşor când se elimină mai mulți parametri, dar, în acelaşi timp, se confirmă faptul că nu toate criteriile au aceeaşi importanţă pentru că creşterea ratei de erori este mică. Pentru unele aplicaţii implementate pe dispozitive cu putere de procesare mică se poate face un compromis între reducerea numărului de calcule şi descreşterea ratei de recunoaştere. Cum numărul de parametrii folosiți este 26, eliminarea a trei parametri înseamnă o reducere absolută a numărului de calcule cu 11.5%, iar eliminarea a 5 criterii implică o reducere a numărului de calcule cu 19.2%.

Testele de mai sus au fost făcute numai prin analiza modelelor antrenate. Aşa cum s-a explicat şi în capitolul 6.1.4., **Algoritmul 3** ţine cont şi de o indicaţie, care provine din rezultatele de recunoaştere, despre cât de des se confundă două modele. Rezultatele obţinute cu acest algoritm, atât pentru ponderare cât şi pentru eliminarea parametrilor, sunt prezentate în Tabelele 6.8. şi 6.9.

Tabelul 6.8. Ratele de erori la aplicarea Algoritmului 3

	WER [%]		
m	FR	GS	GS-DR
1	15.52	8.13	2.47
2	15.03	7.69	2.38
3	14.87	7.79	2.47
4	15.04	8.4	2.47
5	15.56	8.48	2.57

Tabelul 6.9. Ratele de erori la aplicarea Algoritmului 3 pentru eliminarea parametrilor

	WER [%]		
Numărul (*K*) al parametrilor eliminați	FR (*m*=3)	GS (*m*=2)	GS-DR (*m*=2)
1	15.6	8.3	2.47
2	16.02	8.38	2.51
3	16.12	8.66	2.58
4	18.02	9.66	2.97
5	19.5	10.4	3.39

Aceste rezultate arată că **Algoritmul 3** obţine rezultate mai bune pentru ponderarea parametrilor. Această îmunătăţire derivă din faptul că, faţă de **Algoritmul 1**, **Algoritmul 3** foloseşte şi informaţia care provine din rezultatele de recunoaştere. De multe ori, confuzia fonemelor este influenţată de context (de exemplu, poziţia unui fonem într-un cuvânt), dar confuzia calculată din modelele antrenate (ca în **Algoritmul 1**) nu are această informaţie de context. **Algoritmul 3**, în schimb, prin folosirea acestei informaţii, calculează ponderi mai precise.

În ceea ce priveşte eliminarea parametrilor, rezultatele sunt mai bune decât cele obţinute cu **Algoritmul 1**, dar nu sunt la fel de bune ca cele ale **Algoritmului 2**. Rezultatele se pot îmbunătăţi dacă se găseşte o metodă de a combina **Algoritmul 2** şi **Algoritmul 3**.

Rezultatele arată că se poate optimiza contribuţia parametrilor în discriminarea modelelor prin introducerea ponderilor care se calculează cum este descris în capitolul 6.1.3. WER scade atât cu **Algoritmul 1** cât şi cu **Algoritmul 3**. Deşi ameliorarea a fost mică, ea este încurăjătoare pentru că algoritmii de caclul pentru ponderi poate fi îmbunătăţit, de exemplu, prin utilizarea unui alte funcţie de ponderare, sau prin calcularea IAM la nivel de stare în loc de model (vezi capitolul 6.1.6.). În cazul eliminării parametrilor, WER creşte pentru toţi algoritmii, dar creşte foarte puţin. În aplicaţii unde puterea de procesare este limitată, descreşterea ratei de recunoaştere poate fi sacrificată pentru creşterea eficienţei de calcul.

6.1.6. Utilizarea IAM de tip 1 în combinație cu decodarea N-best

În capitolul 6.1.5. s-a tras concluzia că IAM are cel mai mare succes dacă este calculat şi aplicat la nivel de stare. Pentru două modele, distribuțiile unor stări se pot suprapune foarte mult, iar pentru alte stări nu se suprapun deloc. Însă, pentru a calcula IAM între două stări oarecare ale două modele oarecare, este nevoie ca stările să fie aliniate, altfel o comparaţie directă este complicată. Un caz în care fonemele sunt aliniate este cazul decodării ”N-best”. Aici, stările candidate sunt aliniate şi IAM între ele se poate calcula în timpul recunoaşterii sau poate fi calculat a priori şi citit apoi dintr-ul tabel.

Primul pas constă într-o decodare *N*-best ASR. În Fig. 6.10., este prezentată acurateţea recunoaşterii în sensul WER pentru diferite valori ale lui *N*. WER este calculat numărând ca eroare cazul în care cuvântul corect nu se găseşte în nici una dintre cele *N* ipoteze. Caracteristicile sistemului de recunoaştere cu care s-au obţinut aceste rezultate sunt descrise în capitolul 6.1.5. Performanţa sistemului fără nici o îmbunătăţire este cea dată de valoarea lui WER pentru *N*=1 (18.5%). Aceasta înseamnă că prin găsirea unei metode de a face decizii mai bune printre cele mai bune *N* ipoteze, se poate scade WER.

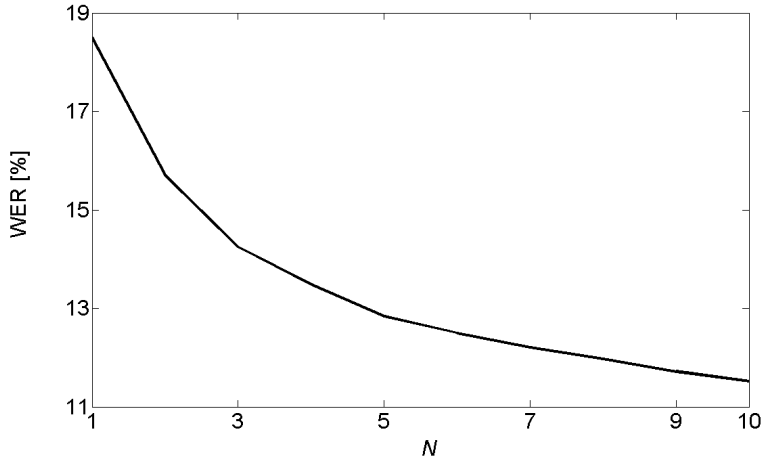

Fig. 6.10. Variaţia ratei de erori cu numărul de ipoteze *N*.

Aşa cum este prezentat şi în capitolul 6.1.1., scorul (probabilitatea logaritmică) este calculat cu formula (6.2), iar formula (6.5) arată că toţi parametrii au aceeaşi pondere la scorul final deşi nu toţi contribuie la fel la discriminarea modelelor. Relevanţa lor o măsurăm cu IAM de tip 1 calculată, însă, numai pentru o stare (şi nu pentru întregul model ca în capitolul 6.1.3.). Dacă ponderăm parametrii corespunzător IAM de tip 1, putem minimiza eroarea de recunoaştere. Se vor folosi două funcţii monoton descrescătoare pentru calculul ponderilor date de formulele (6.15) și (6.16) [36]:

$$w_i = \frac{(u_i)^{-1}}{\sum_k^n (u_i)^{-1}} \tag{6.15}$$

unde u_i este IAM de tip 1, iar numărătorul este folosit pentru normare.

$$w_i = \frac{\exp(-a \cdot u_i)}{\sum_k^n \exp(-a \cdot u_k)} \tag{6.16}$$

unde a este un parametru de ajustare, iar numărătorul este folosit, şi aici, pentru normare.

În formula (6.15), ponderea este invers proporţională cu IAM, iar în formula (6.16), ponderea variază exponenţial cu IAM. Aceste formule presupun că decizia se face numai între două modele. Când decizia de recunoaştere se ia pe un număr mai mare de modele, calculul ponderilor nu se mai poate face cu formulele (6.15) şi (6.16), pentru că un model are IAM diferite cu diferite modele, pentru stări diferite. Din acest motiv, IAM şi ponderile sunt aplicate numai celor *N* ipoteze rezultate din decodarea *N*-best. Ipotezele sunt grupate în perechi şi din fiecare pereche se alege un câştigător pentru runda următoare de comparaţii, până când este ales un câştigător final. Aşadar, e bine ca numărul *N* să fie egal cu o putere de a lui 2, dar metodă funcţionează pentru orice valoare a lui *N*. Prima ipoteză este împerechiată cu ultima, a doua cu penultima, ş.a.m.d.

Din decodarea *N*-best se obţine succesiunea de stări, de aceea distribuţiile folosite pentru calcularea scorurilor sunt ştiute. Ipotezele din aceeaşi pereche sunt aliniate şi ponderile pentru parametrii sunt aplicate pentru fiecare cadru de semnal vocal. IAM sunt calculate, în timpul procesului de recunoaştere, ca fiind area gri din Fig. 6.6. O altă abordare ar fi de a calcula dinainte toate IAM pentru oricare pereche de stări ale oricăror două modele şi de a salva aceste valori într-un tabel.

Formula pentru calculul probabilităţii logaritmice per cadru devine:

$$\log N(O,\mu,\Sigma) = C - \frac{1}{2}\sum_{i}^{n} w_i \frac{(o_i - \mu_i)^2}{\sigma_i^2}, \tag{6.17}$$

La descrierea rezultatelor experimentale, ne vom referi la această metoda cu numele **Metoda 1**.

Ca alternativă pentru probabilitatea logaritmică, se poate folosi un alt scor. Scorurile (probabilitatea logaritmică) per cadru se calculează, în continuare, la fel ca şi până acum cu formula (6.17), iar scorul final pentru o ipoteză va contoriza numărul de cadre pentru care ea a avut scor mai bun decât cealaltă ipoteză din aceeaşi pereche. Această metoda dă rezultate bune pentru semnale de voce degradate, unde scorul unor cadre degradate poate afecta scorul final. La descrierea rezultatelor experimentale, ne vom referi la această metoda cu numele **Metoda 2** şi se va arăta că ea dă rezultate bune şi cu semnale de voce curate.

6.1.7. Rezultatele experimentale aplicând IAM de tip 1 la ipotezele decodării *N*-best

Pentru experimente s-a folosit aceeaşi configuraţie ca în capitolul 6.1.5. astfel încât rezultatele se pot compara. Decodarea *N*-best se face cu algoritmul Token Passing descris în capitolul 2.2.6. Rezultatele obţinute cu **Metoda 1** sunt prezentate în Tabelul 6.10.:

Tabelul 6.10. Rezultatele experimentale obţinute cu Metoda 1

N	*Scorul maxim*	WER [%] cu funcţia invers proporţională	WER [%] cu funţia exponenţială, a=1	WER [%] cu funţia exponenţială, a=0.5	WER [%] cu funţia exponenţială, a=0.1
1	18.5	18.5	18.5	18.5	18.5
2	15.71	18.63	17.46	16.52	17.61
3	14.25	18.53	16.97	15.34	17.15
4	13.49	18.57	16.56	15.03	16.94
5	12.84	18.52	16.38	14.85	16.81
6	12.5	18.51	16.32	14.65	16.72
7	12.21	18.49	16.25	14.61	16.61
8	11.97	18.48	16.14	14.56	16.56
9	11.72	18.46	16.1	14.54	16.52
10	11.52	18.47	16.08	14.48	16.48

Experimentele au fost făcute pentre diferite valori ale lui *N*. Coloana *Scorul maxim* arată îmbunătăţirea potenţială care poate fi făcută prin utilizara tehnicii *N*-best. Dacă cuvântul corect este găsit într-una din cele *N*- ipoteze, atunci recunoaşterea se consideră corectă. Restul de coloane arată rezultatele pentru **Metoda 1**, unde ponderile sunt calculate cu formulele (6.15) şi respectiv (6.16) cu trei valori diferite pentru parametrul de ajustare a. Ponderile care sunt calculate invers proporţionale cu IAM nu aduc vreo îmbunătăţire din punct de vedere al WER. Cu toate acestea, nu toate cuvintele care au fost recunoscute corect în acest caz sunt aceleaşi cu cuvintelele recunoscute corect la pasul 1. De exemplu, pentru N=4, 2% dintre cuvintele recunoscute corect la primul pas au fost recunoscute eronat la al doilea pas. Aceste rezultate se pot explica prin modul în care sunt calculate ponderile, adică invers proporţionale cu IAM. Această metodă poate duce la diferenţe mari între ponderi şi în unele cazuri decizia este luată, practic, cu câţiva parametri. Ponderile calculate cu funcţia exponenţială corectează această problemă şi, de consecinţă, se obţin rezultate mai bune. Parametrul de ajustare a determină sensibilitatea ponderilor la IAM. Pentru valori mari ale lui a, ponderile variază într-o plajă mare de valori, iar pentru valori mai mici de-ale lui a, ponderile tind să iau valori apropiate. De aceea, există o valoare optimă pentru a. La aceste experimente, cele mai bune rezultate au fost obţinute pentru a=0.5.

WER scade când N creşte, dar pentru valori mai mari ale lui N, WER se saturează. Valori mari ale lui N implică şi un număr mai mare de calcule, de aceea valoarea potrivită pentru N este aleasă funcţie de capacitatea de procesare a sistemului. O reducere relativă de 22% este obţinută pentru WER.

Rezultatele obţinute cu **Metoda 2** sunt prezentate în Tabelul 6.11. Ele sunt similare cu cele obţinute cu **Metoda 1** deşi diferă puţin. Aceasta poate fi explicată prin faptul că pentru unele foneme ale cuvântului corect, scorurile obţinute in cadrele respective sunt foarte mici, afectând, astfel scorul total, care poate fi în avantajul unui cuvânt similar. WER, în cazul funcţiei exponenţiale cu a=0.5, este redus cu o valoare relativă de 2.7% faţă de **Metoda 1.**

Tabelul 6.11. Rezultatele experimentale obţinute cu Metoda 2

N	*Scorul maxim*	WER [%] cu funcţia invers proporţională	WER [%] cu funţia exponenţială, a=1	WER [%] cu funţia exponenţială, a=0.5	WER [%] cu funţia exponenţială, a=0.1
1	18.5	18.5	18.5	18.5	18.5
2	15.71	18.63	17.21	16.26	17.45
3	14.25	18.53	16.62	15.05	16.91
4	13.49	18.53	16.26	14.71	16.71
5	12.84	18.52	16	14.51	16.48
6	12.5	18.53	15.91	14.29	16.41
7	12.21	18.47	15.84	14.23	16.32
8	11.97	18.47	15.78	14.18	16.24
9	11.72	18.46	15.73	14.12	16.22
10	11.52	18.47	15.72	14.09	16.19

Pentru a evalua generalitatea **Metodei 1**, aceasta a fost implementată și pe o bază de date standard în limba engleză. S-a ales baza de date TIMIT care are dimensiuni comparabile cu baza noastră de date, dar oferă o variabilitate mai ridicată datorită numărului mare de vorbitori și de dialecte. Caracteristicile celor două baze de date sunt prezentate în Tabelul 6.12.

Tabelul 6.12. Caracteristicile bazelor de date în română și TIMIT

	Nr. de cuvinte	Nr. de cuvinte distincte	Nr. de vorbitori	Nr. de dialecte	Nr. de foneme
Baza de date în română	50000	10000	5	1	34
TIMIT	50000	6250	630	8	75

Rezultatele obținute cu baza de date TIMIT sunt prezentate în Tabelul 6.13. Ele sunt asemănătoare cu cele obținute cu baza de date în limba română. Cele mai bune rezultate se obțin pentru a=0.5, iar cu creșterea lui N WER scade. Și în acest caz, WER se saturează cu creșterea lui N (WER descreşte cu o valoare absolută de 1.05% când N creşte de la 7 la 10). Îmbunătățirea adusă de Metoda 1 este o reducere relativă a WER cu 39%.

Tabelul 6.13. Rezultatele obținute cu Metoda 1 și baza de date TIMIT

N	Rezultate N-best	WER [%] cu funţia exponenţială, a=1	WER [%] cu funţia exponenţială, a=0.5	WER [%] cu funţia exponenţială, a=0.1
1	37.13	37.13	37.13	37.13
2	26.79	34.12	29.32	34.31
3	21.53	32.22	27.56	32.55
4	18.75	31.07	26.28	31.44
5	16.18	29.98	25.12	30.53
6	14.69	28.81	24.12	29.66
7	13.38	27.96	23.67	28.79
8	12.29	27.03	23.12	27.94
9	11.21	26.62	22.88	27.41
10	10.28	26.30	22.62	27.10

6.1.8. Utilizarea IAM de tip 2

IAM de tip 2 scoate în evidenţă criteriile irelevante. De aceea, ea poate fi folosită doar pentru eliminarea parametrilor. O dată stabilit numărul (K) de parametri care trebuie eliminați, se elimină parametri ai căror distribuţii au varianța cea mai mare. Cum parametrii diferiți iau valori în plaje diferite și sunt caracterizate de varianțe diferite, ei nu pot fi comparați direct. În acest scop, am definit o nouă mărime numită *varianța relativă* (VR). Ideea de bază este ca pentru fiecare parametru să se construiască o distribuție globală cu valorile pe care el le ia pentru diferite foneme. VR se va calcula pentru fiecare distribuție a unui parametru pentru fiecare dintre stările modelelor, ca fiind raportul între varianța distribuției și varianța distribuției globale. Deci, primul pas este construirea varianței globale.

Date fiind două distribuții $f_1(x)$ cu media μ_1 și varianța σ_1^2 și $f_2(x)$ cu media μ_2 și cu varianța σ_2^2, care au fost calculate din același număr de observații, atunci funcția de distribuție care ar rezulta din uniunea tuturor observațiilor se poate calcula din cele două cu formula:

$$f(x) = \frac{1}{2}(f_1(x) + f_2(x)) \tag{6.18}$$

Calculând σ^2 cu formula clasică:

$$\sigma^2 = \int_{-\infty}^{\infty} (x-\mu)^2 f(x) dx \tag{6.19}$$

se poate deduce σ^2 din variațele și mediile distribuțiilor $f_1(x)$ și $f_2(x)$:

$$\sigma^2 = \frac{\sigma_1^2 + \sigma_2^2}{2} + \frac{(\mu_1 - \mu_2)^2}{4} \tag{6.20}$$

unde

$$\mu = \mu_1 + \mu_2 \tag{6.21}$$

Se poate demonstra prin inducție că pentru $M*S$ distribuții (M fiind numărul de modele iar S numărul de stări ale modelului), varianța poate fi calculată cu formula (6.22):

$$\sigma^2 = \frac{1}{(MS)} \sum_{i=1}^{MS} \sigma_i^2 + \sum_{i=1}^{MS-1} \sum_{j>i}^{MS} (\mu_i - \mu_j)^2 \tag{6.22}$$

unde

$$\mu = \frac{1}{MS} \sum_{i=1}^{MS} \mu_i \tag{6.23}$$

Pentru simplitate, s-a considerat că pentru toate modelele s-a folosit același număr de eșantioane pentru antrenare.

Așadar, *varianța relativă* σ_{ri}^2 poate fi definit ca raportul între varianța distribuției σ_i^2 și *varianța globală* σ^2, și se calculează cu formula:

$$\sigma_{ri}^2 = \frac{\sigma_i^2}{\sigma^2} \tag{6.24}$$

Atunci, algoritmul de eliminare a parametrilor cu IAM de tip 2 este:

1. Se calculează *varianța globală* σ^2 cu formula (6.22).
2. Se calculează *variațele relative* σ_{ri}^2 cu formula (6.24) pentru fiecare stare ale fiecărui model.
3. Se elimină cei K parametri cu *variațele relative* cele mai mici, adică li se asignează ponderi egale cu 0.

La discutarea rezultatelor experimentale, ne vom referi la acest algoritm cu numele **Algoritm 4**.

6.1.9. Rezultatele experimentale aplicând IAM de tip 2

Rezultatele experimentale obținute prin aplicarea **Algoritmului 4** sunt prezentate în Tabelul 6.14. S-au făcut teste pentru toate tipurile de gramatici așa cum este explicat în capitolulul 6.1.5., iar acuratețea recunoașterii este măsurată cu rata de erori (WER). Comparând aceste rezultate cu rezultatele obținute cu **Algoritmul 2**, fiind cel mai bun algoritm la eliminarea criteriilor folosind IAM de tip 1, se poate observa că **Algoritmul 4** dă rezultate puțin mai bune cu toate tipurile de gramatici. De exemplu, dacă se decide să se elimine 3 criterii, atunci îmbunătățirea relativă este de 0.8%, 1.8% și 1.9% pentru cazurile FR, respectiv GS și GS-DR. Această îmbunătățire se explică prin faptul că IAM de tip 2 este calculat la nivel de stare. Aplicarea IAM la nivel de stare face ca evaluarea criteriilor, din punct de vedere al relevanței lor, să fie mai fină. Această caracteristică s-a manifestat și în cazul decodării *N*-best unde, de asemenea, IAM era calculat la nivel de stare.

Tabelul 6.14. Rezultatele experimentale obținute cu Algoritmul 4

	WER [%]		
Numărul (*K*) al parametrilor eliminați	FR	GS	GS-DR
1	15.57	8.22	2.47
2	15.76	8.31	2.47
3	15.95	8.41	2.53
4	16.07	9.03	2.65
5	17.37	9.96	2.94

6.1.10. Utilizarea IAM de tip 3

IAM de tip 3 este cauzată de suprapunerea a două distribuții bine antrenate care nu sunt aproape una față de cealaltă ($\mu_i+3\sigma_i< \mu_j-3\sigma_j$, așa cum este arătat în Fig. 6.8.). Problema apare în cazul în care trebuie să decidem între mai multe modele prin mai mulți parametri. Dacă observația are o valoare în afara ambelor intervale [μ_i-3σ_i; μ_i+3σ_i] și [μ_j-3σ_j; μ_j+3σ_j], atunci cel mai probabil nici unul din cele două modele M_i și M_j nu a emis această observațite. Însă, dacă valoarea observației este mai aproape de distribuția lui M_i, contribuția acestui parametru în probabilitatea totală va fi în avantajul lui M_i față de M_j.

Pentru a corecta acest inconvenient, se poate modifica funcția de distribuție $f(x)$ a ieșirii stărilor HMM-ului, în $g(x)$, ca în formula (6.25):

$$g(x)=\begin{cases} f(x), & x\in[\mu-3\sigma:\mu+3\sigma] \\ c, & \hat{i}n \quad rest \end{cases} \tag{6.25}$$

unde c este o constantă de valoare mică și care poate fi determinată empiric. Astfel, toate distribuțiile pentru care observația x ia valori în afara intervalului [μ-3σ; μ+3σ], vor avea aceeași contribuție la scorul final.

Rezultatele experimentale, însă, nu au adus vreo îmbunătățire din punct de vedere al ratei de recunoaștere, dar nici o scădere a ei. Acest fapt poate fi explicat prin numărul extrem de mare de parametri care participă la scorul final. Pentru un model, la fiecare cadru contribuie la scor 26 de parametri per componentă a mixturii gaussiene (12 MFCC + energia împreună cu derivatele lor). Aceasta înseamnă că pentru 1s de semnal vocal, cu un cadru de 10ms, avem 100 de cadre. În total, la 1s de semnal vocal, contribuie 26*100=2600 de parametri. Din acest motiv, contribuțiile parametrilor care se află în situația de IAM de tip 3 se compensează între ele făcând ca această incertitudine să nu fie decizivă la luarea deciziei finale. Pe de altă parte, faptul că folosirea funcției $g(x)$ în locul $f(x)$ nu a adus scădere a ratei de recunoaștere arată că acele intervale ale funcției nu ajută la luarea deciziilor bune.

6.1.11. Protecția relativă față de zgomot folosind IAM

Distorsiunea semnalului vocal din cauza zgomotului afectează scorul obținut de modele la fiecare cadru și scorul total. Din punct de vedere al GMM, un parametru ia o decizie bună cât timp valoarea observației nu depășește un prag (Fig. 6.11.). Acest prag este valoarea pentru care scorul modelului corect este egal cu scorul unui model incorect. Considerând distribuțiile pentru două modele, să presupunem că o observație dată aparține modelului *j* (linia întreruptă). Dacă valoarea observației este mai mică decât valoarea de prag (valorea pentru care cele două distrbuții sunt egale), atunci scorul pentru modelul *j* va fi mai mare decât scorul pentru modelul *i*, altfel va fi mai mic. În primul caz, parametrul a ajutat în decizia de a a alege modelul *j* ca fiind cel mai probabil. În cazul unui semnal afectat de zgomot, valoarea distorsionată a observației poate fi mai mare sau mai mică decât valoarea originală, dar decizia va fi corectă cât timp valoarea distorsionată nu va depăși pragul. Se definește *marginea de eroare* ca fiind distanța valorii originale față de prag. *Marginea de eroare* este, în medie, mai mare dacă IAM este mai mică și cu cât este mai mare *marginea de eroare* cu atât mai bine este protejat un model față de zgomot. Așadar, asignând ponderi mai mari criteriilor cu IAM mai mică ajută recunoașterea în condiții de zgomot.

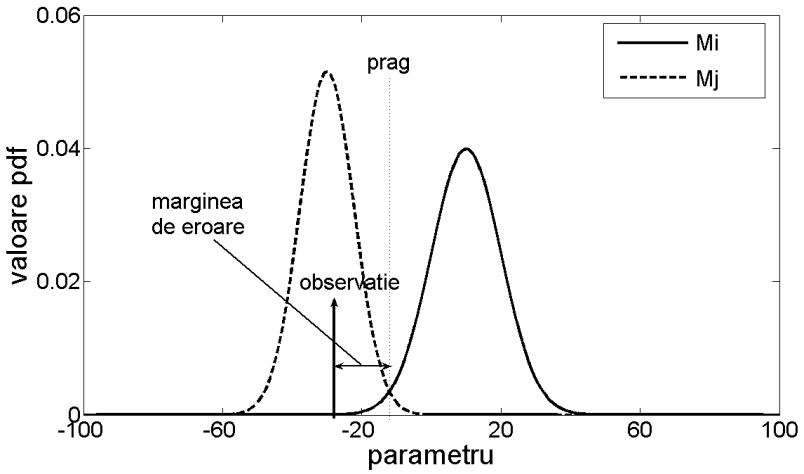

Fig. 6.11. Marginea de eroare și protecția față de zgomot

Rezultatele obținute pentru semnale vocale distorsionate de zgomot sunt prezentate în 6.2.1.

6.2. APLICAREA METODELOR BAZATE PE IAM ÎN CONDIȚII DE ZGOMOT, PIERDERE DE PACHETE ȘI SEMNAL VOCAL CODAT

6.2.1. Aplicarea metodelor bazate pe IAM în condiții de zgomot

Pentru evaluarea metodelor bazate pe IAM în condiții de zgomot s-a ales un zgomot alb gaussian. Acuratețea recunoașterii s-a măsurat pentru diferite valori ale raportului semnal zgomot (RSZ). Experimentele au fost făcute folosind același SRLV neoptimizat ca și în capitolul 6.1.5, pe care au fost aplicate metodele de optimizare **Metoda 1** și **Metoda 2** din capitolul 6.1.6. Rezultatele experimentale sunt prezentate în Tabelul 6.15. Se observă că sistemul neoptimizat nu este robust față de zgomot. Când RSZ se apropie de valoarea 30 dB, WER crește mult. Prin aplicarea metodelor bazate pe IAM, pentru același RSZ, WER este mai mic. Capicitatea metodelor de a recupera din cuvintele recunoscute eronat este explicată în capitolul 6.1.11. Cu toate acestea, trebuie ținut cont de un lucru: chiar și în condiții de zgomot **Metoda 1** și **Metoda 2** contribuie la reducerea efectului incertitudinii acustice, dar pe lângă aceasta, contribuie și la creșterea robusteții față de zgomot. Pentru o valoare dată a RSZ, nu avem o măsură care să ne spună câte cuvinte au fost recuperate prin reducerea efectului incertitudinii acustice și câte cuvinte au fost recuperate prin reducerea efectului zgomotul. Totuși, așa cum s-a explicat și în capitolul 6.1.11., există un prag pentru nivelul zgomotului peste care **Metoda 1** și **Metoda 2** nu mai pot recupera cuvinte recunoscute eronat. Acest lucru se poate observa și din datele din tabelul 6.15.. În absența zgomotului (RSZ=40dB), WER în cazul **Metodei 1** este cu 4% mai mic față de cazul neoptimizat. Pentru un RSZ $\in [20,30]$ dB, WER în cazul **Metodei 1** este cel puțin 10% mai mic decât în cazul neoptimizat. Iar pentru RSZ=15dB, diferența în WER între cele două cazuri este de 6%.

Tabelul 6.15. Aplicarea metodelor bazate pe IAM în condiții de zgomot

RSZ [dB]	WER [%] neoptimizat	WER [%] **Metoda 1** (10-*best*)	WER [%] **Metoda 2** (10-*best*)
40	18.5	14.5	14.0
35	21.6	17.3	17.0
30	39.8	27.1	27.4
25	69.2	54.7	57.3
20	90.4	70.4	73.7
15	98.3	92.3	95.9

6.2.2. Aplicarea metodelor bazate pe IAM în condiții de pierdere de pachete

Pentru evaluarea efectului pierderii de pachete asupra acurateței RLV, trebuie mai întâi alese modelele de pierderi. Fiecare canal de comunicație are modelul lui de pierderi de pachete și acestea pot fi modelate prin modelul Gilbert-Elliot. După cum s-a prezentat în capitolul 3.6.1, acest model are doi parametri: p responsabil pentru rata de pachete pierdute și q responsabil pentru pierderea de pachete în rafală. Experimentele au fost făcute pentru două cazuri de pierderi de pachete:

1. Pierderile sunt perfect distribuite (q=0) pentru a vedea influența ratei de pierderii asupra acurateței RLV.
2. Pierderile au rată constantă, dar variem numărul de pachete pierdute conzecutive (adică variem q), pentru a vedea efectul pierderilor de pachete în rafală asupra acurateței RLV.

Trebuie menționat, că rata de recunoaștere, pentru aceeași rată de pierderi, poate avea valori diferite funcție de poziționarea pachetelor pierdute în semnal. Cu toate acestea, experimentele făcându-se pe un număr mare de înregistrări această componentă aleatoare converge către valoarea ei medie.

O altă problemă care apare la pierderea de pachete este cum se formează semnalul distorsionat. Au fost analizate două opțiuni: fie pachetele pierdute sunt înlocuite cu zerouri, fie se concatenează pachetele recepționate constituind, astfel, semnalul vocal. Rezultatele pentru cazul pierderii distribuite sunt prezentate în Fig. 6.12., iar rezultatele pentru cazul pierderilor în rafală sunt prezentate în tabelele 6.16. și 6.17.

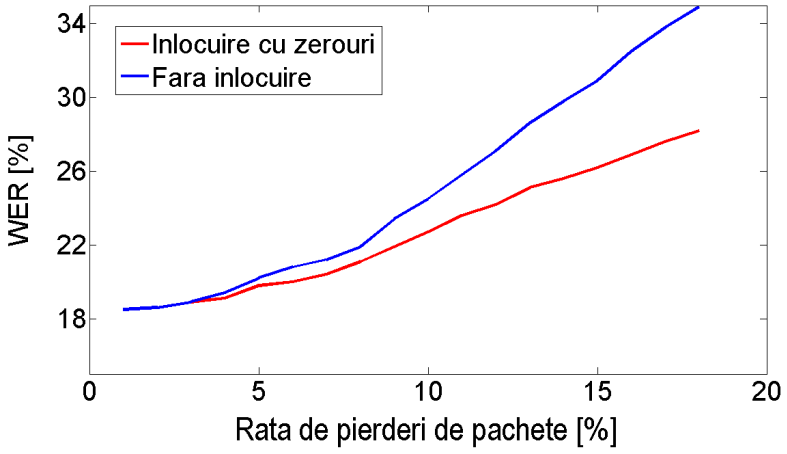

Fig. 6.12 Variația ratei de erori funcție de rata de pierderi distribuite de pachete

În cazul în care pierderea de pachete este distribuită, rezultatele cele mai bune sunt obținute dacă nu se înlocuiesc pachetele pierdute cu zerouri. Acest rezultat era de așteptat dacă se ține cont de modul în care lucrează algoritmul Viterbi cu HMM (vezi capitolul 2.2.5.). La căutarea drumului optim, cadrele de voce cu zerouri ar avantaja în mod arbitrar unele modele pentru care distribuțiile parametrilor au medii aproape de zero.

În cazul pierderilor în rafală, rezultatele cele mai bune sunt obținute dacă se înlocuiesc pachetele pierdute cu zerouri. Acest rezultat se explică prin faptul că HMM utilizat pentru modelarea fonemelor nu are tranziții posibile decât de la starea i la stările i, i+1 și i+2. Concatenarea pachetelor recepționate are succes cât timp HMM poate tranzita de la starea corespunzătoare unui cadru de semnal vocal la starea corespunzătoare cadrului succesiv. În primul caz, acest lucru era posibil pentru numărul de pachete conzecutive pierdute era egal cu 1 (q=0). Având în vedere aceste considerații, la experimente s-a folosit înlocuirea cu zerouri pentru pierderi în rafală, și concatenarea pentru pierderi distribuite.

Tabelul 6.16. Rata de erori pentru pierderi de pachete în rafală cu o rată de 5%

p	q	Rata de pierderi de pachete [%]	WER [%] (Fără înlocuire)	WER [%] (Înlocuire cu zerouri)
0	0	0	18.5	18.5
0.05	0.2	5	20.1	20
0.03	0.4	5	21	21.1
0.02	0.5	5	23.9	22.6
0.02	0.6	5	24.9	24.1
0.02	0.7	5	27.8	26.8
0.03	0.7	5	30.7	29.2
0.01	0.8	5	36.3	32

Tabelul 6.17. Rata de erori pentru pierderi de pachete în rafală cu o rată de 21%

p	q	Rata de pierderi de pachete [%]	WER [%] (Fără înlocuire)	WER [%] (Înlocuire cu zerouri)
0	0	0	18.5	18.5
0.28	0	21	29.5	30
0.26	0.1	21	30.7	31.2
0.22	0.2	21	34.9	34.1
0.18	0.3	21	39.2	37.5
0.18	0.4	21	43.4	41.9
0.14	0.5	21	48.4	45.2
0.1	0.6	21	55.6	51.8

În tabelele 6.16. și 6.17., s-a păstrat rata de pierderi de pachete constantă și s-a variat numărul de pachete pierdute conzecutive (parametrul q). Din aceste tabele se observă că, pentru aceeași rata de pierderi de pachete, pierderea de pachete în rafală degradează mai mult acuratețea RLV. Aceste rezultate confirmă ce s-a studiat și în [165].

În continuare, s-au folosit metodele bazate pe IAM de tip 1 aplicate pe cele mai bune 10 (N=10) ipoteze, așa cum s-a explicat la capitolul 6.1.6. În Fig. 6.13., sunt prezentate rezultatele în condițiile în care pachetele pierdute sunt distribuite (q=0).

Metodele bazate pe IAM de tip 1 rezultă eficiente și în cazul în care există pierderi de pachete. **Metoda 1** și **Metoda 2** obțin aproximativ aceeași acurateță. În multe dintre cazuri, **Metoda 2** obține rezultate mai bune. Acest fenomen poate fi explicat prin faptul că **Metoda 2** nu ține cont de probabilitatea de tranziție între diferite stări. Pachetele pierdute pot duce la situația în care probabilitatea de tranziție între două stări corespunzătoare a două cadre succesive de semnal vocal, să fie mică. Acest lucru este evitat în cazul **Metodei 2**. La o analiză cantitativă, era de așteptat ca metodele de optimizare să nu fie eficiente la rate mari de pierderi de pachete, adică pentru multe pachete pierdute metodele să nu mai poată recupera cuvintele recunoscute greșit. Însă, dacă cuvintele corecte se află printre cele mai bune N ipoteze, deși n-au fost alese ca fiind cele mai probabile la prima iterație, ele pot fi recuperate.

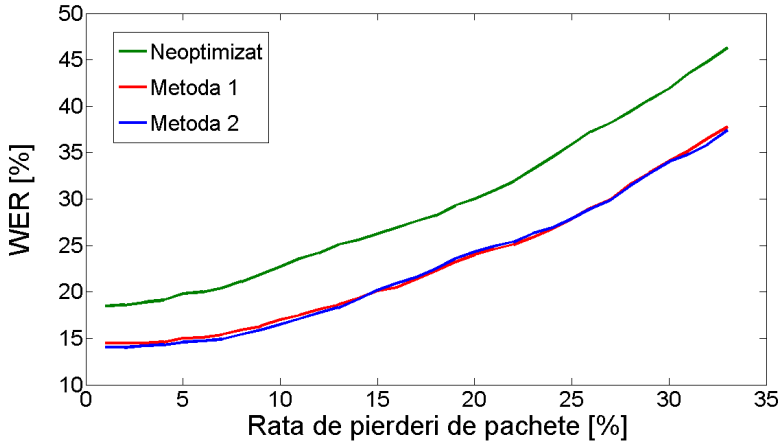

Fig. 6.13. Rata de erori pentru pierderi distribuite de pachete folosind metodele bazate pe IAM

În tabelele 6.18., 6.19. și 6.20., sunt prezentate rezultatele obținute prin aplicarea metodelor bazate pe IAM de tip 1 în condiții de pierderi în rafală. La fiecare tabel, s-a menținut constantă rata de pierderi și s-a crescut numărul de pachete pierdute conzecutive (prin mărirea lui *q*). Se observă că pentru rată de pierderi mică cele două metode obțin rezultate mai bune decât cazul neoptimizat. În schimb, pentru rate mari de pierderi de pachete aceste metode nu mai pot recupera din cuvintele recunoscute greșit.

Tabelul 6.18. Rata de erori pentru pierderi de pachete în rafală cu o rată de 5% și aplicând metodele bazate pe IAM.

p	q	Rata de pierderi de pachete [%]	WER [%] neoptimizat	WER [%] **Metoda 1** (10-*best*)	WER [%] **Metoda 2** (10-*best*)
0	0	0	18.5	14.5	14
0.05	0.2	5	20	15.1	14.7
0.03	0.4	5	21.1	15.9	15.7
0.02	0.5	5	22.6	17.2	17.1
0.02	0.6	5	24.1	18.6	18.9
0.02	0.7	5	26.8	20.1	20.3
0.03	0.7	5	29.2	23.4	23.7
0.01	0.8	5	32	26.1	26.3

Tabelul 6.19. Rata de erori pentru pierderi de pachete în rafală cu o rată de 21% și aplicând metodele bazate pe IAM.

p	q	Rata de pierderi de pachete [%]	WER [%] neoptimizat	WER [%] **Metoda 1** (10-*best*)	WER [%] **Metoda 2** (10-*best*)
0	0	0	18.5	14.5	14
0.28	0	21	30	24	24.3
0.26	0.1	21	31.2	25.1	25.2
0.22	0.2	21	34.1	26.5	26.8
0.18	0.3	21	37.5	28.9	28.9
0.18	0.4	21	41.9	32.5	32.4
0.14	0.5	21	45.2	37.3	37.1
0.1	0.6	21	51.8	43	42.9

Tabelul 6.20. Rata de erori pentru pierderi de pachete în rafală cu o rată de 46% și aplicând metodele bazate pe IAM.

p	q	Rata de pierderi de pachete [%]	WER [%] neoptimizat	WER [%] **Metoda 1** (10-*best*)	WER [%] **Metoda 2** (10-*best*)
0	0	0	18.5	14.5	14
0.5	0.4	46	59.2	58.9	59.1
0.45	0.5	46	62.5	62.2	62.2
0.38	0.6	46	66.5	66.4	66.5
0.25	0.7	46	70.2	70.3	70.3
0.19	0.8	46	76.1	76	76.1
0.21	0.8	46	84.3	84.3	84.3
0.08	0.9	46	88.2	88.2	88.2

6.2.3. Efectul codării asupra acurateței semnalului vocal

Pentru evaluarea efectului codării asupra semnalului vocal s-au făcut experimente cu următoarele codecuri: G.711 A-Law, G.711 μ-Law, G.723, G.726 (16, 24, 32 kb/s) și G.729. Rezultatele experimentale sunt prezentate în tabelul 6.21. Modelul acustic a fost antrenat cu semnale vocale aparținând unui singur vorbitor și codate cu codecul respectiv. Topologia HMM utilizată este de 6 stări, 2 componente de mixturi gaussiene, 12 MFCC + energia împreună cu derivatele de ordin I. Unitatea de limbă aleasă este alofonul. Stările 3 și 4 sunt legate pentru alofonele cu același fonem central.

Tabelul 6.21. Variația ratei de recunoaștere funcție de codec

Codec	Rata de recunoaștere [%]
Necodat (16 kHz)	92.30
G.711 A-Law	87.63
G.711 μ-Law	87.61
G.723 6.3 kb/s	83.20
G.723 5.3 kb/s	82.43
G.726 32 kb/s	83.81
G.726 24 kb/s	82.74
G.726 16 kb/s	81.99
G.729 A	74.75

Este important de observat că rata de recunoaștere nu crește monoton cu rata de transmisie a codecului. De exemplu codecul G.723 deși are cea mai mică rată de transmisie are o rată de recunoaștere mai bună decât G.729 care are o rată de transmisie de 10 kb/s. Codecurile G.711 deși nu fac compresie de semnal codat totuși nu ating performanțele semnalului necodat, care folosește o frecvență de eșantionare de 16 khz. Aceasta înseamnă că în semnalul vocal există componente importante pentru recunoaștere între frecventele 4 – 8 kHz (ceea ce era de asteptat).

CAPITOLUL 7

CONCLUZII ŞI CONTRIBUŢII PERSONALE

7.1. CONTRIBUŢII PERSONALE

Această lucrare are două obiective principale: dezvoltarea unui SRLV în limba română şi analiza comportării acestuia în reţele de telecomunicaţii. Aceasta din urmă presupune optimizarea RLV în vederea implementării pe dispozitive mobile şi reducerea efectelor perturbaţiilor pe care semnalul vocal le suferă în canalul de telecomunicaţii.

La îndeplinirea obiectivelor acestei lucrări şi la rezolvarea problemelor tratate, am adus următoarele contribuţii personale:

- Am implementat o infrastructură pentru antrenarea, testarea şi verificarea rezultatelor SRLV în limba română. Infrastructura a fost construită având la bază utilitarele HTK [68]. Însă, aceste utilitare sunt puţin flexibile în ceea ce priveşte datele de intrare. În acest scop, au fost definite proceduri în Matlab™ care permit prelucrarea rapidă a unor cantităţi mari de date şi care permit o flexibilitate ridicată a datelor de intrare.
- Am stabilit proceduri pentru definirea topologiei HMM funcţie de numărul mare de parametrii ai modelului. Scripturile de verificare şi evaluare a sistemului au fost scrise astfel încât să ajute procesul de investigaţie (au ajutat, în mod special, la stabilirea parametrilor relevanţi descrişi în Capitolul 6). Totodată, scripturile au simplificat procesul de rulare astfel încât utilitarele să pot fi folosite şi de către studenţi mai puţin experţi în domeniul RLV.
- Am făcut un studiu detaliat (prezentat în Capitolul 4) despre obţinerea unei baze de date mari, variată, fără erori şi cu puţine resurse la dispoziţie. Din încercările făcute am ajuns la concluzia că cel mai rapid mod de a obţine o bază de date etichetată este prin înregistrare directă, în care vorbitorul pronunţă cuvinte/fraze prestabilite. Metodele de etichetare manuală a unor înregistrări sunt consumătoare de timp şi

vulnerabile la erori. Metoda prin înregistrare directă este mai robustă la erori, dar oferă o variabilitate scăzută și, de consecință, un model acustic mai puțin general. Pentru RLV continuă, etichetarea trebuie făcută maxim la nivel de frază. Pentru etichetări mai rare algoritmul de antrenare Baum-Welch nu reușește să alinieze semnalul vocal cu transcrierile [28], [167], [29].

- Am construit cu ajutorul unor studenţi baze de date etichetate cu caracteristici diferite funcţie de aplicaţie (recunoaştere a vorbirii continue, spontane, de cuvinte izolate, etc., descrise în Capitolul 4). Un aspect important la construirea bazei de date este numărul echilibrat de apariții ale fonemelor [167], [170], [27].
- Am definit şi am implementat o strategie de antrenare (descrisă în Capitolul 4) care să prevină următoarelor probleme [28]:
    - *Alinierea între semnalul vocal și transcrierea fonetică datorită etichetării rare ale bazei de date* a fost rezolvată prin antrenarea pe mai multe nivele (fonem izolat, fonem embedded, alofon embedded).
    - *Izolarea erorilor de etichetare* a fost rezolvată prin trecerea sub procesul de recunoaștere a bazei de date pentru antrenare.
    - *Numărul mic de apariții ale alofonelor* a fost rezolvat prin tehnica stărilor legate.
- Am realizat un studiu detaliat şi un număr mare de experimente pentru găsirea configuraţiei optime a modelului acustic (descrise în Capitolul 5). Configurația optimă este stabilită având în vedere două criterii: creșterea acurateței recunoașterii și reducerea, pe cât posibil, a numărului de calcule necesare în vederea implementării pe dispozitive mobile. În urma experimentelor realizate prin variația parametrilor HMM (număr de stări, de parametri, de componente ale mixturii, etc.), am ajuns la concluzia că modelul trebuie să fie cu atât mai detaliat cu cât baza de date este mai variată. Pentru fiecare element variabil există o valoare optimă, deoarece un model detaliat necesită mai multe date (și mai variate) la antrenare. Fiecare detaliere a modelului, însă, crește complexitatea sistemului și numărul de calcule necesare. De aceea, funcție de resursele de calcul disponibile, se propune un compromis [170].
- Am analizat influenţa măririi bazei de date asupra acurateţei recunoaşterii. La mărirea bazei de date rata de recunoaștere crește, dar după o anumită dimensiune începe să se satureze. În cazul RLV în limba română, rata de recunoaștere se saturează atunci când aparițiile fonemelor în baza de date sunt de câteva mii.
- Am implementat un script în Matlab™ care să permită rularea automată și verificarea unei cantități mari de teste. Rezultatele obținute, (prezentate în Capitolul 5) împreună cu interpretarea lor, dau o imagine clară a influenței variațiilor unor parametrii asupra performanței SRLV (atât din punct de vedere al acurateței cât și al vitezei de rulare).
- Am propus o metodă de îmbunătăţire a acurateţei de recunoaştere care se bazează pe combinarea scorurilor obţinute din două SRLV diferite, unul antrenat cu foneme şi altul antrenat cu alofone. Am încercat diferite metode de a pondera scorurile provenite din cele două surse și am ajuns la concluzia că cea mai bună metodă este să se decidă

pe baza probabilității logaritmice. Această soluție este posibilă datorită faptului că probabilitatea logaritmică nu depinde de unitatea de limbă, ci doar de numărul de parametri vocali. Soluția și rezultatele experimentale obținute sunt prezentate în Capitolul 5 [48].

- Am realizat un studiu asupra factorilor care influențează complexitatea SRLV (și implicit și viteza de rulare), cum ar fi numărul de stări ale HMM, unitatea de limbă aleasă, dimensiunea vocabularului, dimensiunea fascicolului de căutare, numărul de parametrii vocali, numărul de componente ale mixturii gaussiene, etc. Printre factorii care afectează cel mai mult viteza de rulare, am evidențiat: dimensiunea rețelei de noduri, numărul de comparații ale scorurilor intermediare la fiecare nod și dimensiunea vocabularului. Dimensiunea rețelei de noduri depinde de numărul total de modele și de numărul de stări al fiecărui model. Din acest punct de vedere, fonemele sunt preferate față de alofone pentru că sunt mai puține la număr. Numărul de comparații la noduri se poate reduce prin micșorarea dimensiunii fascicolului de căutare. Micșorarea fascicolului de căutare duce la scăderea ratei de recunoaștere, pentru că este posibili ca drumul cel corect să obțină scoruri mici la primele ferestre de timp și să fie eliminat. Utilizarea alofonelor contriubuie, de asemenea, la reducerea numărului de jetoane care ajung la nod pentru că nu toate treanzițiile sunt posibile în acest caz. Mărirea vocabularului duce la mărirea spațiului de căutare în rețeaua de noduri și de consecință la creșterea timpului de calcul. Însă, în majoritatea cazurilor, vocabularul este impus de aplicație. Acestea sunt prezentate în Capitolul 5 [48], [170].
- Am propus și elaborat o metodă care se bazează pe recunoaștere în trei pași (descrisă în Capitolul 5). La primul pas se face o recunoaștere la nivel de fonem care este rapidă (vocabularul fiind mic). Din ea se evidențiază fonemele (unu sau două la număr) care au fost recunoscute cu cea mai mare probabilitate. Pe baza lor se alege un dicționar fonetic care conține numai cuvinte cu aceste foneme. Cu acest dicționar (care este mai mic decât cel care conține toate cuvintele vocabularului) se face o recunoaștere la nivel de cuvânt [48].
- Prin experimentele făcute am demonstrat că nu toți parametrii vocali discrimină la fel între cuvinte [35].
- Am definit o nouă mărime, *incertitudinea acustică a modelelor* (IAM), ca fiind suprapunerea distribuţiilor unui parametru pentru modele diferite. Am demonstrat experimental că parametrii vocali a căror IAM este mai mică sunt mai importanți. Am definit trei tipuri de IAM, fiecare reprezentând o sursă diferită de erori. Această mărime a fost folosită în metodele bazate pe IAM descrise în Capitolul 6 [35], [36].
- Am demonstrat experimental că prin eliminarea parametrilor cei mai puțini importanți rata de recunoaștere rămâne aceeași [35].
- Am demonstrat experimental că prin ponderarea parametrilor vocali funcție de relevanța lor se obține o creștere a ratei de recunoaștere [35], [36].
- Prin experimentele făcute am ajuns la concluzia că un model nu este confundat cu toate celelalte modele în aceeași măsură, ci există o mulțime de modele cu care un

model dat este confundat cel mai des. De aceea, analiza pentru eliminarea/ponderarea parametrilor se poate face doar pentru această mulțime [35].

- Prin experimentele făcute am ajuns la concluzia că utilizarea IAM la nivel de stare este mai eficientă decât IAM la nivel de model [35], [36].
- Prin experimentele făcute am ajuns la concluzia că dintre cele trei tipuri de IAM, sursa principală de erori este IAM de tip I [35].
- Am propus şi implementat un algoritm care ponderează parametrii vocali (un algoritm de *rescoring*) folosind IAM la nivel de stare ca un criteriu de relevanţă. IAM la nivel de stare a fost aplicată celor *N*-best ipoteze rezultate din prima etapă de recunoaștere. Rezultatele experimentale (prezentate în Capitolul 6) au demonstrat eficienţa algoritmului atât în condiţii normale cât şi în prezenţa factorilor perturbatori de zgomot şi pierderi de pachete [36].
- Am propus şi implementat un algoritm (prezentat în Capitolul 6) care reduce complexitatea sistemului prin eliminarea parametrilor care prezintă cea mai mare IAM. Pentru optimizarea algoritmului, IAM pentru un model dat a fost calculată numai cu modele cu care acest model se confundă cel mai des. Identificarea modelelor care se confund cel mai des a fost realizată prin construirea matricii de confuzie. La construirea acestei matrici, a fost nevoie de implementarea unui algoritm DTW adaptat pentru alinierea cuvintelor corecte și recunoscute [35].
- Am analizat performanţa SRLV în limba română în condiţii de zgomot. La aceste experimente s-a folosit un zgomot alb gaussian cu diverse nivele pentru RSZ.
- Am analizat performanţa SRLV în limba română în condiţii de pierderi de pachete. Au fost încercate diferite modele de pierderi de pachete și pentru aceasta a fost nevoie de implementarea unui generator de pierderi de pachete după modelul Gilbert-Elliot.
- Am realizat evaluarea unor codecuri în vederea utilizării lor în RLV. Codecurile evaluate sunt G.711 A-Law, G.711 μ-Law, G.723, G.726 (16, 24, 32 kb/s) și G.729.

7.2. DIRECȚII DE CERCETARE VIITOARE

Există trei direcții principale de dezvoltare a acestor studii:

- Creșterea acurateței de recunoaștere prin implementarea unui model de limbă pentru limba română.
- Reducerea complexității sistemului de recunoaștere și realizarea unei implementări *embedded* pe dispozive mobile.
- Creșterea robusteții sistemului de recunoaștere în condiții de zgomot și pierderi de pachete.

Construirea unui model de limbă nu a fost unul dintre obiectivele acestei lucrări, dar el este indispensabil într-un sistem de recunoaștere de vorbire continuă. Un asemenea model poate fi construit cu ajutorul *n*-gramelor, însă este nevoie de un corpus foarte mare în limba română. Acesta poate fi obținut implementând un sistem care să obțină date din Internet.

Implementarea *embedded* pe un dispozitiv mobil poate fi realizată utilizând concluziile din experimentele făcute în această lucrare şi aplicând metode de cuantizare pentru prelucrări în virgulă fixă.

Robusteţea sistemului poate fi crescută în continuare prin implementarea metodelor bazate pe compensarea parametrilor şi pe compensarea modelului acustic. Implementarea algoritmului de „parametri lipsă” poate aduce o creştere suplimentară a robusteţei. Efectul pierderii de pachete poate fi anulat prin metode de mascare a pierderilor bazate pe interpolare.

Lista de lucrări

1. C. Burileanu, C.S. Petrea, **A. Buzo**, H. Cucu, A. Pasca, "Report on Building a Tool for Romanian Spontaneous Speech Recognition", The Phonetician, ISSN 0741-6164, Number 97/2008-I-II, pp. 68-98, 2008.
2. C. Petrea, D. Ghelmez-Hanes, V. Popescu, **A. Buzo**, C. Burileanu, "Spontaneous Speech Database for Romanian Language", ECIT, Iasi, 2008, CD, 15p, Session X: Natural Language & Speech Technology.
3. C. Petrea, D. Ghelmez-Hanes, **A. Buzo**, V. Popescu, C. Burileanu, "Spontaneous Speech Database for the Romanian Language with Medical Applicability", Proc. BIOSTEC, pp. 78-86, 2009, ISBN 978-989-8111-78-4.
4. C. Petrea, D. Ghelmez-Hanes, **A. Buzo**, C. Burileanu, "Statistical Results in the Context of Romanian Spontaneous Speech Recognition", Proc. SPECOM, pp. 126-129, 2009, pp. 458-463, ISBN 978-5-8088-0442-5.
5. C. Petrea, **A. Buzo**, H. Cucu, M. Paşca, C. Burileanu, "Speech Recognition Experimental Results for Romanian Language", Proc. ECIT2010 – The 6th European Conference on Intelligent Systems and Technologies, Iasi, Romania, 2010, CD ISSN: 2069-038X, Invited Plenary Session I.
6. C. Burileanu, **A. Buzo**, C. S. Petre, D. Ghelmez-Hanes, H. Cucu, "Romanian Spoken Language Resources and Annotation for Speaker Independent Spontaneous Speech Recognition," Proc. ICDT, pp.7-10, 2010, Scopus art. no. 5532390, ISBN: 978-0-7695-4071-9.
7. C. Burileanu, V. Popescu, **A. Buzo**, C. S. Petrea, D. Ghelmez-Hanes, "Spontaneous Speech Recognition for Romanian in Spoken Dialogue Systems", Proc. Romanian Academy Series A-Mathematics, Physics, Technical Sciences, Information Science, Vol. 11. No. 1, pp. 83-91, 2010, ISSN 1454-9069.
8. C. Burileanu, C. S. Petrea, **A. Buzo**, H. Cucu, "Speech Recognition Experiments Starting from Isolated Words for Spoken Romanian Language", Multilinguality and Interoperability in Language Processing with Emphasis on Romanian, Romanian Academy Publishing House, Bucharest, ISBN: 978-973-27-1972-5, pp. 229-242, 2010.
9. **A. Buzo**, H. Cucu, C. Burileanu, M. Paşca, V.Popescu, "Word Error Rate Improvement and Complexity Reduction in Automatic Speech Recognition by analyzing Acoustic Model Uncertainty and Confusion", Proc. SpeD, pp. 67-74, 2011, ISBN 978-1-4577-0439-0, IEEE Catalog Number CFP1155H-DVD.
10. **A. Buzo**, H. Cucu, C. Burileanu, "Improving Automatic Speech Recognition Robustness for the Romanian Language", Proc. EUSIPCO, pp. 2119-2122, 2011, ISSN 2076-1465.
11. H. Cucu, L. Besacier, C. Burileanu, **A. Buzo**, "Enhancing Automatic Speech Recognition for Romanian by Using Machine Translated and Web-based Text Corpora", SPECOM, 2011, 6 p. (accepted paper).

12. H. Cucu, **A. Buzo**, C. Burileanu, “Optimization Methods for Large Vocabulary, Isolated Words Recognition in Romanian Language”, Buletin UPB, Seria C, Vol. 73, Nr. 2, pp. 179-192, 2011, ISSN 1454-234x.
13. H. Cucu, L. Besacier, C. Burileanu, **A. Buzo**, “Investigating The Role of Machine Translated Text in ASR Domain Adaptation: Unsupervised and Semi-supervised Methods”, Proc. ASRU (accepted paper, to appear in December 2011).
14. C. Burileanu, H. Cucu, A. Buzo, Miruna Paşca, Cristina S. Petrea, V. Popescu, “Towards Continuous Speech Recognition in the Romanian Language”, book chapter in H.-N. Teodorescu, L. C. Jain, C. Burileanu, editors, “Intelligent Systems Technologies: from Tools to Applications”, Springer Verlag, 2011, 27 p. (in press)
15. A. Buzo, H. Cucu, C. Burileanu, “Improving the Speech Recognition Robustness in Conditions of Noise and Packet Loss by reducing the Acoustic Model Uncertainty”, book chapter in H.-N. Teodorescu, L. C. Jain, C. Burileanu, editors, “Intelligent Systems Technologies: from Tools to Applications”, Springer Verlag, 2011, 15 p. (in press).

Bibliografie

[1] A. Acero, “Acoustical and Environmental Robustness in Automatic Speech Recognition”, Kluwer Academic Publishers, 1993.
[2] A. Acero, L. Deng, T. Kristjansson, J. Zhang, “HMM Adaptation Using Vector Taylor Series for Noisy Speech Recognition,” Proc. ICSLP, pp. 121-124, 2000.
[3] A. Acero, T. Kristjansson, J. Zhang, J., ”Hmm Adaptation using Vector Taylor Series for Noisy Speech Recognition”, Proc. ICSLP, Vol.3, pp. 869-872, 2000.
[4] A. V. Aho, R. Sethi, J D. Ullman. “Compilers: Principles, Techniques, and Tools”, Reading, Massachusetts: Addison-Wesley, 1986.
[5] G. Alon, “Key-Word Spotting-The Base Technology for Speech Analytics”, White Paper, Natural Speech Communication Ltd, 2005.
[6] A. Amir, A. Efrat, S. Srinivasan, “Advances in Phonetic Word Spotting”, Proc. CIKM , pp. 580-582, 2001.
[7] V. Apopei, “Analiza unor sisteme neliniare cu aplicaţii în prelucrarea semnalelor”, Teză de doctorat, Institutul de Informatică Teoretică, Iaşi, 2008.
[8] L.R. Bahl, et al., “The IBM Large Vocabulary Continuous Speech Recognizer for the ARPA NAB News Task,” Proc. ARPA Spoken Language Systems Technology Workshop, pp. 121-126, Austin, Texas, USA, January 1995.
[9] J. Baker, et al., “Large Vocabulary Recognition of Wall Street Journal Sentences at Dragon Systems,” Proc. DARPA Speech & Natural Language Workshop, pp. 387-392, Harriman, NY, February 1992.
[10] J. Barker, M. Cooke, P. D. Green, “Robust ASR based on Clean Speech Models: an Evaluation of Missing Data Techniques for Connected Digit Recognition in Noise,” Proc. Eurospeech, pp. 213–216, 2001.
[11] J.R. Bellegarda, D. Nahamoo, “Tied Mixture Continuous Parameter Modeling for Speech Recognition,” IEEE Transactions on Acoustics Speech and Signal Process., Vol. 38, No. 12, pp. 2033-2045, 1990.
[12] M. Benzeghiba, et al., "Automatic Speech Recognition and Speech Variability: a Review", in Speech Communication, Vol. 49, Issue 10-11, pp. 763-786, 2007.
[13] A. Bernard, A. Alwan, “Low-Bitrate Distributed Speech Recognition for Packet-Based and Wireless Communication”, IEEE Trans. on Speech and Audio Processing, Vol. 10, No. 8, pp. 570–579, 2002.
[14] L. Besacier, C. Bergamini, D. Vaufreydaz, E. Castelli, “The Effect of Speech and Audio Compression on Speech Recognition Performance”, IEEE Multimedia Signal Processing Workshop, Cannes, France, October 2001.
[15] L. Besacier, J.-F. Bonastre, P. Mayorga, C. Fredouille, S. Meignier, “Overview of Compression and Packet Loss Effects in Speech Biometrics”, IEEE Proceedings Vision, Image & Signal Processing - Special issue on Biometrics on the Internet”, Vol. 150, No. 6, pp. 455-458, 2003.
[16] E. Bocchieri, ”Vector Quantization for the Efficient Computation of Continuous Density Likelihoods”, Proc. ICASSP, Vol. 2, pp. 692-695, 1993.
[17] E. Bocchieri, B. Mak, “Subspace Distribution clustering Hidden Markov Model,” IEEE Trans. ASSP, Vol. 9, pp.264-275, 2001.
[18] E. Bocchieri, D. Blewett, “A decoder for LVCSR Based on Fixed-Point Arithmetic”, Proc. ICASSP, pp. 769-772, 2006.
[19] M. Boldea, ”Contribuții la recunoașterea automată a vorbirii continue în limba română”, Teză de doctorat, Universitatea ”Politehnică” din Timișoara, 2003.
[20] S. Boll, “Suppression of Acoustic Noise in Speech Using Spectral Subtraction," IEEE Transactions on Acoustics Speech and Signal Processing, Vol. 27, pp. 113–120, 1979.

[21] J. Bolot “End-to-End Packet Delay and Loss Behavior in the Internet ACM Sigcomm”, pp. 289–298, San Francisco, CA, USA, 1993.
[22] J. Bolot, H. Crepin, A. Vega “Analysis of Audio Packet Loss in the Internet”, Proc. of International Workshop on Network and Operating System Support for Digital Audio and Video, pp. 163–174. Durham, New Hampshire, USA.
[23] C. Boulis, M. Ostendorf, E. Riskinm, S. Otterson, “Graceful Degradation of Speech Recognition Performance over Packet-Erasure Networks”, IEEE Trans. on Speech and Audio Processing, Vol. 10, No. 8, pp. 580–590, 2002.
[24] J.S. Bridle, M. D. Brown, R. M. Chamberlain, “An Algorithm for Connected Word Recognition”, Proc. ICASSP, pp. 899-902, 1982.
[25] F. Brugnara, M. Cettolo, M. Federico, D. Giuliani, “A Baseline for the Transcription of Italian Broadcast News,” Proc. ICASSP, pp. 441-444, 2000.
[26] C. Burileanu, V. Teodorescu, G. Stolojanu, C. Radu, “Sistem cu logică programată pentru recunoaşterea cuvintelor izolate, independent de vorbitor”, Inteligenţa artificială şi robotica, Editura Academiei Române, Bucureşti, Vol. 1, pp.266-274, 1983.
[27] C. Burileanu, C.S. Petrea, A. Buzo, H. Cucu, A. Pasca, “Report on Building a Tool for Romanian Spontaneous Speech Recognition”, The Phonetician, ISSN 0741-6164, Number 97/2008-I-II, pp. 68-98, 2008.
[28] C. Burileanu, A. Buzo, C. S. Petre, D. Ghelmez-Hanes, H. Cucu, “Romanian Spoken Language Resources and Annotation for Speaker Independent Spontaneous Speech Recognition,” Proc. ICDT, pp.7-10, 2010, Scopus art. no. 5532390, ISBN: 978-0-7695-4071-9.
[29] C. Burileanu, V. Popescu, A. Buzo, C. S. Petrea, D. Ghelmez-Hanes, “Spontaneous Speech Recognition for Romanian in Spoken Dialogue Systems”, Proc. Romanian Academy Series A-Mathematics, Physics, Technical Sciences, Information Science, Vol. 11. No. 1, pp. 83-91, 2010, ISSN 1454-9069.
[30] C. Burileanu, C. S. Petrea, A. Buzo, H. Cucu, “Experimente de recunoaştere a limbajului vorbit, în limba română, pornind de la cuvinte izolate”, Conferința “Resurse lingvistice și instrumente pentru prelucrarea limbii române” ConsILR, București 2010.
[31] C. Burileanu, C. S. Petrea, A. Buzo, H. Cucu, “Speech Recognition Experiments Starting from Isolated Words for Spoken Romanian Language”, Multilinguality and Interoperability in Language Processing with Emphasis on Romanian, Romanian Academy Publishing House, Bucharest, ISBN: 978-973-27-1972-5, pp. 229-242, 2010.
[32] D. Burileanu, M. Sima, C. Negrescu, V. Croitoru, “Robust Recognition of Small-Vocabulary Telephone-Quality Speech”, in Speech Technology and Human-Computer Dialogue, Publishing House of the Romanian Academy, Bucharest, pp. 145-154, 2003, ISBN 973-27-0963-4.
[33] D. Burileanu, A. Dervis, “An Efficient Keyword Identification Method for Continuous Speech”, Proc. 30th International Conference “Modern Technologies in the XXI Century”, pp. 195-201, 2003 (CD-ROM Proceedings, ISBN 973-640-012-3).
[34] D. Burileanu, C. Negrescu, “Prosody Modeling for an Embedded TTS System Implementation”, Proc. EUSIPCO, pp. 715-718, 2006.
[35] A. Buzo, H. Cucu, C. Burileanu, M. Paşca, V.Popescu, “Word Error Rate Improvement and Complexity Reduction in Automatic Speech Recognition by analyzing Acoustic Model Uncertainty and Confusion”, Proc. SpeD, pp. 67-74, 2011, ISBN 978-1-4577-0439-0, IEEE Catalog Number CFP1155H-DVD.
[36] A. Buzo, H. Cucu, C. Burileanu, “Improving Automatic Speech Recognition Robustness for the Romanian Language”, Proc. EUSIPCO, pp. 2119-2122, 2011, ISSN 2076-1465.
[37] W.M. Campbell, “Generalized Linear Discriminant Sequence Kernels for Speaker Recognition”, Proc. ICASSP, pp. 161-164, 2002.
[38] A. Cardenal, L. Docio, C. Garcia, “Soft Decoding Strategies for Distributed Speech Recognition over IP Networks”, Proc. ICASSP, pp. 921-924, 2004.
[39] L. Chase, et al., “Improvements in Language, Lexical and Phonetic Modeling in Sphinx-II,” Proc. ARPA Spoken Language Systems Technology Workshop, pp. 60-65, Austin, TX, January 1995.
[40] S. Chen, E. Eide, M. Gales, R. Gopinath, P. Olsen, “Recent Improvements in IBM’s Speech Recognition System for Automatic Transcription of Broadcast Speech”, Proc. DARPA Broadcast News Workshop NIST, Herndon,VA, USA, February, 1999.

[41] S.F. Chen, D. Beeferman, R. Rosenfeld, "Evaluation Metrics for Language Models", DARPA Broadcast News Transcription and Understanding Workshop NIST, pp. 275-280, Landsdowne,VA, USA, February 1998.
[42] S.F. Chen, J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," Computer, Speech & Language, Vol. 13(4), pp. 359-394, 1999.
[43] A. Chiţu, "Algoritmi de prelucrarea a semnalului vocal cu procesoare de semnal", Teză de doctorat, Universitatea Politehnica Bucuresti, 2005.
[44] M. Cohen, "Phonological Structures for Speech Recognition", PhD Thesis, U. Ca. Berkeley, 1989.
[45] M. Collins, "A New Statistical Parser Based on Bigram Lexical Dependencies", Proc. Annual Meeting of the Association of Computational Linguistics, pp.184-191, 1996.
[46] P. Cosi, "Hybrid HMM-NN Architectures for Connected Digit Recognition," Proc. International Joint Conference on Neural Networks, Vol. 5, pp. 393-396, 2000.
[47] H. Cucu, L. Besacier, C. Burileanu, A. Buzo."Enhancing Automatic Speech Recognition for Romanian by Using Machine Translated and Web-based Text Corpora", SPECOM, 2011, 6 p. (accepted paper).
[48] H. Cucu, A. Buzo, C. Burileanu, "Optimization Methods for Large Vocabulary, Isolated Words Recognition in Romanian Language", Buletin UPB, Seria C, Vol. 73, Nr. 2, pp. 179-192, 2011, ISSN 1454-234x.
[49] X. Cui, M. Iseli, Q. Zhu, A. Alwan, "Evaluation of Noise Robust Features on the Aurora Databases," Proc. ICSLP, Vol.1, pp.481-484, 2002.
[50] S. Davis, P. Mermelstein, "Comparison of Parametric Representations for Monosyllable Word Recognition in Continuously Spoken Sentences," IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 28, No. 4, pp. 357-366, 1980.
[51] B. Delaney, "Increased Robustness against Bit Errors for Distributed Speech Recognition in Wireless Environments", Proc. ICASSP, pp. 637-640, 2005.
[52] L. Deng, A. Acero, M. Plumpe, X. D. Huang, "Large-Vocabulary Speech Recognition under Adverse Acoustic Environments," Proc. ICSLP, pp. 806–809, 2000.
[53] V. Digalakis, H. Murveit, "Genones: Optimization the Degree of Tying in a Large Vocabulary HMM-based Speech Recognizer," Proc. ICASSP, Vol. 1, pp. 537-540, 1994.
[54] V. Digalakis, L. Neumeyer, M. Perakakis, "Quantization of Cepstral Parameters for Speech Recognition over the WWW", IEEE Journal on Selected Areas in Communications, Vol. 17, pp. 82–90, 1999.
[55] S. Doh, R. Stern, "Inter-Class MLLR for Speaker Adaptation," in Proc. ICASSP, Vol. 3, pp. 1755-1758, 2000.
[56] J. Domokos, "Contribuţii la recunoaşterea vorbirii continue şi la procesarea limbajului natural", Teză de doctorat, Universitatea Tehnică din Cluj-Napoca, 2009.
[57] M. Dragănescu, C. Burileanu, coordonatori, "Analiza şi sinteza semnalului vocal", Editura Academiei Române, Bucureşti, 1986.
[58] J. Droppo, L. Deng, A. Acero, "Evaluation of the SPLICE Algorithm on the Aurora2 Database (web update)," Proc. Eurospeech, pp. 217–220, 2001.
[59] J. Droppo, A. Acero, L. Deng, "Uncertainty Decoding with SPLICE for Noise Robust Speech Recognition," Proc. ICASSP, pp. 935-938, 2002.
[60] E.O. Elliott, "Estimates of Error Rates for Codes on Burst-Noise Channels", Bell System Technical Journal, pp. 1977–1997, 1963.
[61] ETSI ES 202 211 v.1.1.1., "Distributed Speech Recognition; Extended Front-End Feature Extraction Algorithm; Compression Algorithms; Back-End Speech Reconstruction Algorithm", Technical Report ETSI ES 202 211, 2001.
[62] ETSI ES 201 108 v.1.1., "Distributed Speech Recognition; Front-End Feature Extraction Algorithm; Compression Algorithms", Technical Report ETSI ES 201 108, 2003,
[63] ETSI ES 202 050 v.1.1.3., "Distributed Speech Recognition; Advanced Front-End Feature Extraction Algorithm; Compression Algorithms", Technical Report ETSI ES 202 050, 2003
[64] ETSI ES 202 212 v.1.1.1., "Distributed Speech Recognition; Extended Advanced Front-End Feature Extraction Algorithm; Compression Algorithms; Back-End Speech Reconstruction Algorithm", Technical Report, 2003.
[65] ETSI TIPHON TS 101-329-5, "QoS Measurements for Voice over IP", Technical Report ETSI TIPHON TS 101-329-5, 2000.

[66] ETSI TR 101 085 v8.0.0, ”Digital Cellular Telecommunications System (Phase 2+) (GSM); Performance Characterization of the GSM Enhanced Full Rate (EFR) Speech” Technical Report ETSI TR 101 085, 2000.
[67] S. Euler, J. Zinke, “The Influence of Speech Coding Algorithms on Automatic Speech Recognition”, Proc. ICASSP, pp. I–621–624, 1994
[68] G. Everman, et al., “The HTK Book”, Version 3.0, Cambridge University Engineering Department, 2005.
[69] W.M. Fisher, W.S. Liggett, A. Le, J.G. Fiscus, D.S. Pallett, “Data Selection for Broadcast News CSR Evaluations,” Proc. DARPA Broadcast News Transcription & Understanding Workshop NIST, pp. 12-15, Landsdowne,VA, USA, February 1998.
[70] H. Fujihara, M. Goto, J. Ogata. “Hyperlinking Lyrics: A method for Creating Hyperlinks between Phrases in Song Lyrics”, Proc. International Conference on Music Information Retrieval ISMIR, 2008.
[71] M. Fujimoto, Y. Ariki, “ Robust Speech Recognition in Additive and Channel Noise Environments Using GMM and EM Algorithm”, Proc. ICASSP, Vol. 1, pp. 941-944, 2004.
[72] M.J.F. Gales, S.J. Young, “Cepstral Parameter Compensation for HMM Recognition in Noise,” Speech Communication, Vol. 12, No. 3, pp. 231–239, 1993.
[73] M.J.F. Gales, S.J. Young, “Robust Speech Recognition in Additive and Convolutional Noise Using Parallel Model Combination,” Computer Speech and Language, Vol. 9, No. 4, pp. 289–308, 1995.
[74] M.J.F. Gales, “Model-based Techniques for Noise Robust Speech Recognition”, PhD thesis, Cambridge University, 1995.
[75] M.J.F. Gales, “The Generation and Use of Regression Class Trees for MLLR Adaptation”, Technical Report Technical Report CUED/F-INFENG/TR263, Cambridge University, 1996.
[76] M.J.F. Gales, “The Generation and Use of Regression Class Trees for MLLR adaptation," Cambridge University, Tech. Rep. CUED/F-INFENG/TR263, 1996.
[77] M.J.F. Gales, S.J. Young, “ Robust continuous speech recognition using parallel model combination. IEEE Transactions on Speech and Audio Processing, pp. 352-359, 1996.
[78] A. Ganapathiraju, J. Hmaker, J. Picone, “Hybrid SVM/HMM Architectures for Speech Recognition,” Proc. ICSLP, Vol. 4, pp. 504–507, 2000.
[79] Y. Gao, B. Ramabhadran, J. Chen, H. Erdogan, M. Picheny, “Innovative Approaches for Large Vocabulary Name Recognition”, Proc. ICASSP, pp. 481-483, 2001.
[80] J.S. Garofolo, et al., “TIMIT Acoustic-Phonetic Continuous Speech Corpus”, Linguistic Data Consortium, Philadelphia, USA, 1993.
[81] J.L. Gauvain, C.H. Lee, “ Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains” IEEE Transactions on Speech and Audio Processing, Vol.2, No. 2, pp. 291–298,1994.
[82] J.L. Gauvain, L.F. Lamel, M. Adda-Decker, “Developments in Continuous Speech Dictation using the ARPA WSJ Task,” Proc. ICASSP, pp. 65-68, 1995.
[83] J. L. Gauvain, L. Lamel, “Large Vocabulary Continuous Speech Recognition: Advances and Applications”, Proc. IEEE, Vol. 88, No. 8, pp. 1181-1200, 2000.
[84] I. Gavăt, M. Zirra, “Fuzzy Models in Vowel Recognition for Romanian Language”, Proc. Fuzzy-IEEE, pp. 1318-1326, 1996.
[85] I. Gavăt, M. Zirra, “A Hybrid NN-HMM System for Connected Digit Recognition over Telephone in Romanian Language”, Proc. IVTTA, pp. 37-40, 1996.
[86] I. Gavăt, et al., “Elemente de Sinteza şi Recunoaşterea Vorbirii”, Editura Printech, Bucureşti, 2000.
[87] D. Gelbart, N. Morgan, “Evaluating Long-Term Spectral Subtraction for Reverberant ASR,” Proc. ASRU, pp. 103–106, 2001.
[88] E. Giachin, A.E. Rosenberg, C.H. Lee, “Word Juncture Modeling using Phonological Rules for HMM-based Continuous Speech Recognition,” Computer Speech & Language, Vol. 5, pp. 155-168, 1991.
[89] E. N. Gilbert, “Capacity of a Burst-Noise Channel”, Bell System Technical Journal, pp. 1253–1265, 1960.
[90] M. Giurgiu, G. Toderean, “Results on Speech Recognition using Artificial Neuronal Networks and Constrained Clustering Segmentation”, ITHURS, 1996.

[91] A. Gomez, A. Peinado, V. Sanchez, B. Milner, A. Rubio, "Statistical-based reconstruction methods for speech recognition in IP networks", Proc. of Robust (Cost 278 and ISCA-ITRW), pp. 367-370, 2004.
[92] D. Haneş, C. S. Petrea, A. Buzo, V. Popescu, C. Burileanu, "Baza de date în limba română pentru recunoaşterea vorbirii spontane", Conferința "Resurse lingvistice și instrumente pentru prelucrarea limbii române" ConsILR, Iași, 2008.
[93] H. Hermansky, "Perceptual Linear Predictive (PLP) Analysis of Speech," Journal of the Acoustical Society of America, Vol. 87. No. 4, pp. 1738-1752,1990.
[94] H. Hermansky, N. Morgan, "RASTA Processing of Speech," IEEE Transactions on Speech and Audio Processing, Vol. 2, No. 4, pp. 578–589, 1994.
[95] L. Hetherington, "PocketSUMMIT: Small Footprint Continuous Speech Recognition," Proc. Interspeech, pp. 1465–1468, 2007.
[96] H.G. Hirsch, D. Pearce, "The AURORA Experimental Framework for the Performance Evaluation of Speech Recognition Systems under Noisy Conditions", Proc. ISCA ITRW ASR2000, pp. 353-356, 2000.
[97] H.G. Hirsch, "The Influence of Speech Coding on Recognition Performance in Telecommunication Networks", Proc. ICSLP, pp. 1877–1880, 2002.
[98] J.E. Hopcroft, J.D. Ullman, "Introduction to Automata Theory, Languages and Computation", Addison-Wesley, 1979.
[99] http://www.cbsnews.com/stories/2010/02/15/business/main6209772.shtml
[100] http://portal.etsi.org/stq/kta/DSR/dsr.asp
[101] C. Huang, T. Chen, S. Li, E. Chang, J. Zhou, "Analysis of Speaker Variability," Proc. Eurospeech, pp. 1377–1380, 2001.
[102] X. Huang, K.F. Lee, "On Speaker-Independent, Speaker-Dependent, and Speaker-Adaptive Speech Recognition", IEEE Transactions on Speech and Audio Processing, Vol. 1, No. 2, pp. 150–157, 1993.
[103] X. Huang, A. Acero, H. Wuen-Hon, "Spoken Language Processing – A Guide to Theory, Algorithm, and System Development", Prentice Hall, 2001.
[104] D. Huggins-Daines, et al., "PocketSphinx: a free, real-time continuous speech recognition system for handheld devices," Proc. ICASSP, pp. 185–188, 2006.
[105] M. Hwang, X. Huang, "Subphonetic Modeling with Markov States - Senone," Proc. ICASSP, Vol. 1, pp. 33-36, 1992.
[106] V. Ion, R. Haeb-Umbach, "A Comparison of Soft-Feature Speech Recognition with Candidate Codecs for Speech Enabled Mobile Services", Proc. ICASSP, pp. 841-844, 2005.
[107] M. Ioniță, C. Burileanu, M. Ioniță, "DTW Algorithm with Associated Matrix for a Passworded Access System", Emerging techniques for communication terminals (ENSEEIHT), pp.265 -270, 1997.
[108] A. James, A. Gomez, B Milner B, "A Comparison of Packet Loss Compensation Methods and Interleaving for Speech Recognition in Burst-like Packet Loss", Proc. ICSLP, pp. 485-488, 2004.
[109] A. James, B. Milner, "Soft Decoding of Temporal Derivatives for Robust Distributed Speech Recognition in Packet Loss", Proc. ICASSP, pp. 723-726, 2005.
[110] W. Jiang, H. Schulzrinne, "Modeling of Packet Loss and Delay and their Effect on Real Time Multimedia Service Quality", Proc. NOSSDAV, pp. 123-126, 2001.
[111] F. de Jong, J.L. Gauvain, J. deb Hartog, K. Netter, "OLIVE: Speech Based Video Retrieval," Proc. CBMI, pp. 88-91,1999.
[112] B.H. Juang, "On the hidden Markov Model and Dynamic Time Warping for Speech Recognition -- A Unified View," AT and T Technical Journal, Vol. 63, No. 7, pp. 1213-1243, 1984.
[113] S. Kanthak, K. Schutz, H. Ney, "Using SIMD Instructions for Fast Likelihood Calculation in LVCSR," Proc. ICASSP, Vol. 3, pp.1531-1534, 2000.
[114] S.M. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer," IEEE Trans. Acoustics, Speech & Signal Processing, Vol. 35(3), pp. 400-401, 1987.
[115] T. Kemp, A. Waibel, "Unsupervised Training of a Speech Recognizer: Recent Experiments," Proc. ESCA Eurospeech, Vol. 6, pp. 2725-2728, 1999.
[116] H. Kim, R. Cox, "Bitstream-Based Feature Extraction for Wireless Speech Recognition", Proc. ICASSP, pp 191-194, 2000.

[117] H. Kim, R. Cox, “A Bitstream-Based Front-End for Wireless Speech Recognition on IS-136 Communications System”, IEEE Transactions on Speech and Audio Processing, Vol. 9, No. 5, pp. 558–568, 2001.
[118] N.S. Kim, “Statistical Linear Approximation for Environment Compensation”, IEEE Signal Processing Letters, Vol. 5, pp. 8-10, 1998.
[119] K. Kinoshita, T. Nakatani, M. Miyoshi, “Spectral Subtraction Steered by Multi-Step Forward Linear Prediction for Single Channel Speech Dereverberation,” Proc. ICASSP, Vol. 1, pp. 817–820, 2006.
[120] R. Kneser, H. Ney, “Improved Backing-off for n-gram Language Modeling,” Proc. IEEE ICASSP, Vol. 1, pp. 181-184, 1995.
[121] R. Krishna, S. A. Mahlke, T. M. Austin, “Architectural Optimizations for Low-Power, Real-Time Speech Recognition”, CASES, pp.220-231, 2003.
[122] L.F. Lamel, J.L. Gauvain, “Continuous Speech Recognition at LIMSI,” Proc. ARPA Workshop on Continuous Speech Recognition, pp. 59-64, Stanford, CA, USA September 1992.
[123] C. Leggetter, C. Woodland, “Flexible Speaker Adaptation using Maximum Likelihood Linear Regression”, Proc. Eurospeech, pp. 1155–1158, 1995.
[124] S. Levinson, R. Rabiner, M. Sondhi, “An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition”, Bell Systems Technical Journal, 62:1035-1074, 1983.
[125] X. Li, Bilmes, “Feature Pruning for Low-Power ASR Systems in Clean and Noisy Environments”, IEEE Signal Processing Letters. Vol. 12, Issue 7. pp. 489-492.
[126] H. Liao, M. J. F. Gales, “Joint Uncertainty Decoding for Noise Robust Speech Recognition,” Proc. Interspeech, pp. 3129–3132, 2005.
[127] H. Liao, M. J. F. Gales, “Issues with Uncertainty Decoding for Noise Robust Speech Recognition,” Proc. ICSLP, pp. 311-314, 2006.
[128] B. Lilly, K. Paliwal, “Effect of Speech Coders on Speech Recognition Performance” Proc. ICSLP, pp. 131-134, 1996.
[129] E. Lin, K. Yu, R. Rutenbar, T. Chen, “Moving Speech Recognition from Software to Silicon: the In Silico Vox Project”, Interspeech, pp. 157-160, 2006.
[130] E. Lin, K. Yu, R. Rutenbar, T. Chen, “A 1000-word Vocabulary, Speaker-Independent, Continuous Live-Mode Speech Recognizer Implemented in a Single FPGA”, International Symposium on Field-Programmable Gate Arrays, pp. 321-324, 2007.
[131] R.P. Lippmann, B.A. Carlson, “Using Missing Feature Theory to actively select Features for Robust Speech Recognition with Interruptions, Filtering and Noise,” Proc. Eurospeech, pp. KN37–KN40, 1997.
[132] A. Ljolje, M.D. Riley, D.M. Hindle, F. Pereira, “The AT&T 60,000 Word Speech-To-Text System,” Proc. ARPA Spoken Language Systems Technology Workshop, pp. 162-165, Austin, TX, USA, January 1995.
[133] P. Lockwood, J. Boudy, “Experiments with a Nonlinear Spectral Subtractor (NSS), Hidden Markov Models and the Projection, for Robust Speech Recognition in Cars,” Speech Communication, Vol. 11, No. 2, pp. 215–228, 1992.
[134] J. Makhoul, “Spectral Linear Prediction, Properties and Applications,” IEEE Transactions Acoustics, Speech and Signal Processing, pp.283-296, 1975.
[135] J. Makhoul, L. Cosell, “LPCW: An LPC Vocoder with Linear Predictive Spectral Warping”, Proc. ICASSP, pp. 466-469, 1976.
[136] A. Mandal. "Transformation Sharing Strategies for MLLR Speaker Adaptation," Ph.D. thesis, University of Washington, 2007.
[137] J. Martin, D. Jurafsky, “Spoken Language Processing”, Prentice Hall, Third Edition, 2007.
[138] B. Mathew, A. Davis, Z. Fang, “A Low-Power Accelerator for the SPHINX 3 Speech Recognition System,” CASES, pp. 210–219, 2003.
[139] P. Mayorga, R. Lamy, L. Besacier, “Recovering of Packet Loss for Distributed Speech Recognition”, Proc. EUSIPCO, pp. 321-325, 2002.
[140] P. Mayorga, L. Besacier, R. Lamy, J. F. Serignat, “Audio Packet Loss over IP and Speech Recognition,” Proc. ASRU, pp. 607-612, 2003.
[141] S. McGlashan, et al., “Voice Extensible Markup Language (VoiceXML) 3.0”, World Wide Web Consortium, Working Draft WD-voicexml30-20101216, 2010.

[142] F. Metze, J. McDonough, H. Soltau, “Speech Recognition over NetMeeting Connections”, Proc. Eurospeech ISCA, pp. 483-486, 2001.
[143] B. Milner, S. Semnani, “Robust Speech Recognition over IP Networks”, Proc. ICASSP, pp. 373-376, 2000.
[144] B. Milner, A. James, “Analysis and Compensation of Packet Loss in Distributed Speech Recognition using Interleaving”, Proc. Eurospeech, pp. 2693–2696, 2003.
[145] B. Milner, A. James, “An Analysis of Packet Loss Models for Distributed Speech Recognition”, Proc. ICSLP, pp. 345-348, 2004.
[146] B. Milner, A. James, “An Analysis of Packet Loss Models for Distributed Speech Recognition”, Proc. ICSLP, pp. 191-194, 2004.
[147] Y. Minami, et al.,"A Large-Vocabulary Continuous Speech Recognition Algorithm and its Application to a Multi-Modal Telephone Directory Assistance System", Proc. HLT, 221-224, 1994.
[148] P.J. Moreno, “Speech Recognition in Noisy Environments”, PhD thesis, Carnegie Mellon University, 1996.
[149] D. Munteanu, “Contribuţii la elaborarea metodelor de recunoaştere a vorbirii în limba română”, Teză de doctorat, Academia Tehnică Militară, București, 2006.
[150] S. Nedevschi, R.K. Patra, E.A. Brewer, "Hardware Speech Recognition for User Interfaces in Low Cost, Low Power Devices", Proc. DAC, pp.684-689, 2005.
[151] E. Nicolau, I. Weber, S. Gavăt, “Recunoaşterea vocalelor în limba română”, Automatica şi electronica, Nr.5, pp.212-219, 1963.
[152] M. Novak, “Towards Large Vocabulary ASR on Embedded Platforms,” Proc. Interspeech, pp. 2309-2312, 2004.
[153] E. Oancea, C. Burileanu, D. Munteanu, “Continuous Speech Recognition System Improvement”, The 3rd Conference on Speech Technology and Human – Computer Dialog SpeD, pp. 81-91, 2005.
[154] K. Ohtsuki, S. Furui, N. Sakurai, A. Iwasaki, Z.P. Zeang, “Recent Advances in Japanese Broadcast News Transcription,” Proc. ESCA Eurospeech, Vol. 2, pp. 671-674, 1999.
[155] B.T. Oshika, V.W. Zue, R.V. Weeks, H. Neu, J. Aurbach, “The Role of Phonological Rules in Speech Understanding Research,” IEEE Trans. Acoustics, Speech, Signal Processing, ASSP-23, pp. 104-112, 1975.
[156] J. Padrell-Sendra, D. Martin-Iglesias, F. Diaz-de Maria, “Support Vector Machines for Continuous Speech Recognition,” Proc. EUSIPCO, 434-439, 2006.
[157] D.S. Pallett, et al., “1994 Benchmark Tests for the ARPA Spoken Language Program,” Proc. ARPA Spoken Language Systems Technology Workshop, pp. 5-36, Austin, TX, USA, January 1995.
[158] D.S. Pallett, J.G. Fiscus, M.A. Przybocki, “1996 Preliminary Broadcast News Benchmark Test,” Proc. DARPA Speech Recognition Workshop, pp. 22-46, Chantilly, VA, USA, February 1997.
[159] D.S. Pallett, J.G. Fiscus, A.F. Martin, M.A. Przybocki, “1997 Broadcast News Benchmark Test Results: English and Non-English,” Proc. DARPA Broadcast News Transcription & Understanding Workshop, pp. 5-11, Landsdowne,VA, USA, February 1998.
[160] D.S. Pallett, J.G. Fiscus, J.S. Garofolo, A.F. Martin, M.A. Przybocki, “1998 Broadcast News Benchmark Test Results: English and Non-English Word Error Rate Performance Measures,” Proc. DARPA Broadcast News Workshop, pp. 5-12, Herndon,VA, February 1999.
[161] A. Park, T. J. Hazen, "ASR Dependent Techniques for speaker Identification," Proc. ICSLP, pp. 345-348, 2002.
[162] A. Peinado, V. Sanchez, J. Perez, A. de la Torre, “HMM-based Channel error Mitigation and its Application to Distributed Speech Recognition”, Speech Communication, Vol. 41, pp. 549–561, 2003.
[163] A. Peinado, A. Gomez, V. Sanchez, J. Perez-Cordoba, A. Rubio, “Packet Loss Concealment Based on VQ Replicas and MMSE Estimation Applied to Distributed Speech Recognition”, Proc. ICASSP, pp. 283-286, 2005.
[164] A. Peinado, V. Sanchez, J. Perez, A. Rubio, “Efficient MMSE-based Channel Error Mitigation Techniques. Application to Distributed Speech Recognition over Wireless Channels”, IEEE Trans. on Wireless Communications, Vol. 4, No. 1, pp. 14–19, 2005.
[165] A. Peinado, J. Segura, “Speech Recognintion over Digital Channels, Robustness and Standards”, Wiley, 2006.
[166] C. Pelaez, A. Gallardo, F. Diaz, “Recognizing Voice over IP: a Robust Front-End for Speech Recognition on the World Wide Web”, IEEE Transactions on Multimedia, Vol. 3, No. 2, pp. 209–218, 2001.

[167] C. Petrea, D. Ghelmez-Hanes, V. Popescu, A. Buzo, C. Burileanu, “Spontaneous Speech Database for Romanian Language”, ECIT, Iasi, 2008, CD, 15p, Session X: Natural Language & Speech Technology.
[168] C. Petrea, D. Ghelmez-Hanes, A. Buzo, V. Popescu, C. Burileanu, “Spontaneous Speech Database for the Romanian Language with Medical Applicability”, Proc. BIOSTEC, pp. 78-86, 2009, ISBN 978-989-8111-78-4.
[169] C. Petrea, D. Ghelmez-Hanes, A. Buzo, C. Burileanu, “Statistical Results in the Context of Romanian Spontaneous Speech Recognition”, Proc. SPECOM, pp. 126-129, 2009, pp. 458-463, ISBN 978-5-8088-0442-5.
[170] C. Petrea, A. Buzo, H. Cucu, M. Paşca, C. Burileanu, “Speech Recognition Experimental Results for Romanian Language”, Proc. ECIT2010 – The 6th European Confe-rence on Intelligent Systems and Technologies, Iasi, Romania, 2010, CD ISSN: 2069-038X, Invited Plenary Session I.
[171] A. Potamianos,V. Weerackody, “Soft-Feature decoding for Speech Recognition over Wireless Channels”, Proc. ICASSP, 2001.
[172] B. Raj, R. Stern, “Missing Feature Approaches in Speech Recognition,” IEEE Signal Processing Magazine, Vol. 22, No. 5, pp. 101–116, 2005.
[173] F. Richardson, M. Ostendorf, J. R. Rohlicek, "Lattice-Based Search Strategies for Large Vocabulary Speech Recognition," Proc. ICASSP, Vol. 1, pp. 576-579, 1995.
[174] T. Robinson, J. Fransen, D. Pye, J. Foote, S. Renals, W.S.J. Camo, “A British English Speech Corpus for Large Vocabulary Continuous Speech Recognition”, Proc. ICASSP, Vol. 1, pp. 81-84, 1995.
[175] M.J. Russell, A.E. Cook, “Experimental Evaluation of Duration Modelling Techniques for Automatic Speech Recognition,” Proc. ICASSP, pp. 2376–2379, 1987.
[176] B. Sabac, Z. Valsan, I. Gavat, “Isolated Word Recognition Using the DTW Algorithm”, in Proc. International Conference COMMUNICATIONS, pp. 245-251, 1998.
[177] A. Sankar, et al., “Noise-Resistant Feature Extraction and Model Training for Robust Speech Recognition,” Proc. ARPA Speech Recognition Workshop, pp. 117-122, Harriman, NY, USA, February 1996.
[178] H. Sanneck, G. Carle G, “A Framework Model for Packet Loss Metrics Based on Loss Runlengths” Proc. IEEE Global Internet, pp. 554–557, 1996.
[179] G.Z. Saon, B. Kingsbury, L. Mangu, U. Chaudhari, “An Architecture for Rapid Decoding of Large Vocabulary Conversational Speech”, Proc. Eurospeech, pp. 161-164, 2003.
[180] M.R. Schroeder, “Recognition of Complex Acoustic Signal”, Life Sciences Research Report 5, edited by T. H. Bullock (Abakon Verlag, Berlin), p. 324, 1977.
[181] J. Schuster, K. Gupta, R. Hoare, A. Jones, “Speech Silicon: An FPGA Architecture for Real-Time, Hidden Markov Model Based Speech Recognition”, EURASIP Journal on Embedded Systems, pp. 33-42, 2006.
[182] R.M. Schwartz, S.C. Austin, "A Comparison of Several Aproximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses," Proc. ICASSP, Vol. 1, pp. 701-704, 1991.
[183] M. Seltzer, B. Raj, R. Stern, “A Bayesian Framework for Spectrographic Mask Estimation for Missing Feature Speech Recognition,” Speech Communication, Vol. 43, No. 4, pp. 379–393, 2004.
[184] S.D. Shenouda, F.W. Zaki, A. Goneid, “Hybrid Fuzzy HMM System for Arabic Connectionist Speech Recognition”, Robotics, pp. 64-69, 2006.
[185] M. Sima, “Contribuţii privind utilizarea reţelelor neuronale în procesarea semnalelor în comunicaţii. Recunoaşterea independentă de vorbitor a cuvintelor rostite izolat în limba română”, Teză de doctorat, Universitatea Politehnica Bucureşti, 2002.
[186] O. Siohan, C. Chesta, C.H. Lee, “Joint Maximum a Posteriori Estimation of Trasformation and Hidden Markov Model Parameters”, Proc. ICASSP, pp. 965-968, 2000.
[187] N. Smith, M. Gales, “Speech Recognition using SVMs,” Advances in Neural Information Processing Systems 14. MIT Press, 2002.
[188] F.K. Soong, E.F. Huang, "A Tree-Trellis Based Fast Search for Finding the N Best Sentence Hypotheses in Continuous Speech Recognition," Proc. ICASSP, Vol. 1, pp. 705-708, 1991.
[189] S.S. Stevens, “On the Psychophysical Law”, Psychological Review, Vol. 64, No. 3, pp. 153–181, 1957.
[190] A.D. Subramaniam, B. D. Rao, “PDF Optimized Parametric Vector Quantization of Speech Line Spectral Frequencies,” IEEE Trans. SAP, Issue 2, pp. 130-142, 2003.

[191] J. Suontausta, J. Hakkinen, V. Olli, ”Fast Decoding in Large Vocabulary Name Dialing”, Proc. ICASSP, pp. 1535–1538, 2000.
[192] S. Takahashi, S. Sagayama, “Four-level Tied Structure for Efficient Representation of Acoustic Modeling,” Proc. ICASSP, pp. 520-523, 1995.
[193] Z.H. Tan, P. Dalsgaard, “Channel Error Protection Scheme for Distributed Speech Recognition Proc. ICSLP, pp. 123-126, 2002.
[194] Z.H. Tan, B. Lindberg, P. Dalsgaard, “A Comparative Study of Feature-Domain Error Concealment Techniques for Distributed Speech Recognition”, Proc. Robust (Cost 278 and ITRW workshop), pp. 201-204, Norwich, UK, 2004.
[195] Z.H. Tan, P. Dalsgaard, B. Lindberg, ”Automatic Speech Recognition over Error-Probe Wireless Networks”, Speech Communication, Vol. 47, No. 1–2, pp. 220–242, 2005.
[196] Z.H. Tan, P. Dalsgaard, B. Lindberg, ”Exploiting Temporal Correlation of Speech for Error Robust and Bandwidth Flexible Distributed Speech Recognition”, IEEE Transactions on Audio Speech and Language Processing, Vol. 15, No. 4, pp. 1391–1403, 2007.
[197] H.N. Teodorescu, L. Pistol, M. Feraru, M. Zbancioc, D. Trandabăţ, “Sounds of the Romanian Language Corpus”, © 2010, http://www.etc.tuiasi.ro/sibm/romanian_spoken_language/index.htm.
[198] H.N. Teodorescu, “A Proposed Theory in Prosody Generation and Perception: th Multi-dimnsional Contextual Integration Principle of Prosody”, SpeD 2005 - 3th Conference on Speech Technology and Human Dialogue, Eds. C. Burileanu, Trend in Speech Technology, Editura Academiei Române, ISBN 973-27-1178-7, pp. 109-118, 2005.
[199] B. H. Tran, F. Seide, V. Steinbiss, "A Word Graph Based N-Best Search in Continuous Speech Recognition," Proc. ICSLP, Vol. 4, pp. 2127-2130, 1996.
[200] Z. Valsan, B. Sabac, I. Gavat, D. Zamfirescu, “Combining Self-Organizing Map and Multi-layer Perceptron in a Neuronal System for Improved Isolated Word Recognition”, in Proc. International Conference COMMUNICATIONS, pp. 245-251, 1998.
[201] A.P. Varga, R.K. Moore, “Hidden Markov Model Decomposition of Speech and Noise,” Proc. ICASSP, pp. 845–848, 1990.
[202] M. Vasilache, “Speech Recognition Using HMMs with Quantized Parameters,” Proc. ICSLP, pp. 441–444, 2000.
[203] L. Villarrubia, L. Hernandez, “Reconocimiento de voz en el entorno de las nuevas redes de comunicacion UMTS e Internet”, Comunicaciones de Telefonica I+D, Vol. 23, pp. 99–112, 2001.
[204] J.Q. Walker, J.T. Hicks, “Taking Charge of your VoIP Project: Strategies and Solutions for Successful VoIP Deployments. Cisco Press, Indianapolis, IN, USA, 2004.
[205] V. Wan, “Speaker Verification using Support Vector Machines”, Ph.D. thesis, University of Heffield, 2003.
[206] X. Wang, D. O’Shaughnessy, “Environmental Independent ASR Model Adaptation/ Compensation by Bayesian Parametric Representation”, IEEE Transactions on Audio Speech and Language Processing, Vol. 15, No. 4, 1204-1217, 2007.
[207] C.J. Waters, B.A. MacDonald, "Efficient Word-Graph Parsing and Search With a Stochastic Context-Free Grammar," Proc. IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 311-318, IEEE Press Piscataway, NJ, USA, 1997, ISBN:0780336984.
[208] S. Wegmann, et al., “Dragon Systems’ 1998 Broadcast News Transcription System”, Proc. DARPA Broadcast News Workshop. NIST, pp. 117-120, 1999.
[209] D. Willett, ”Context-Dependent Duration Modeling”, Proc. ICASSP, 387-390, 2005.
[210] P.C. Woodland, C.J. Leggetter, J.J. Odell, V. Valtchev, S.J. Young, “The Development of the 1994 HTK Large Vocabulary Speech Recognition System,” Proc. ARPA Spoken Language Systems Technology Workshop, pp. 104-109, 1995.
[211] M. Yajnik, S. Moon, J. Kurose, D. Towsley, “Measurement and modeling of the temporal dependence in packet loss” Proc. IEEE Infocom, pp. 345–352, 1999.
[212] S.J. Young, N.H. Russell, J.H.S. Thornton, “Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems”, Technical Report, Cambridge University: Engineering Department, 1989.
[213] S.J. Young, “The General Use of Tying in Phoneme-Based HMM Speech Recognisers,” Proc. ICASSP, pp. 721-724, 1992.
[214] S.J. Young, P.C. Woodland, “The Use of State Tying in Continuous Speech Recognition,” Proc. ESCA Eurospeech, Vol. 3, pp. 2203-2206, 1993.
[215] V. Zue, et al., “A telephone-Based Conversational Interface for Weather Information”, 2000.

[216] G. Zweig, M. Picheny, “Advances in Large Vocabulary Continuous Speech Recognition”, Advances in Computers , Vol. 60, pp. 249-291, 2004.