

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

Prelucrarea de limbaj natural folosind inteligență artificială:
căutarea de cuvinte de interes și creșterea inteligibilității
transcrierilor de vorbire

TEZĂ DE DIZERTAȚIE

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

Conducător lucrare:

Conf. Dr. Ing. Horia CUCU

Student:

Cristian MANOLACHE

București
2019

Universitatea "Politehnica" din București
Facultatea de Electronică, Telecomunicații și Tehnologia Informației
Programul de masterat **Tehnologii multimedia în aplicații de biometrie și securitatea informației**

Anexa 2

TEMA LUCRĂRII DE DISERTAȚIE
a masterandului **MANOLACHE R. Cristian , 421-BIOSINF**

1. Titlul temei: Identificarea cuvintelor de interes în transcrieri de vorbire și creșterea inteligibilității transcrierilor de vorbire

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

Lucrarea de față își propune proiectarea, dezvoltarea și testarea a două componente software de procesare de limbaj natural bazate pe algoritmi de inteligență artificială ce au ca scop (i) identificarea cuvintelor de interes în documente audio, respectiv (ii) creșterea inteligibilității transcrierilor de vorbire în text.

Algoritmii de identificare de cuvinte de interes ce vor fi dezvoltați pornesc de la idei simple de tipul comparație directă de șiruri de caractere. Se au în vedere și algoritmi mai complecși de aliniere imperfectă a cuvintelor căutate cu transcrierile de vorbire, respectiv de lematizare a textelor de comparat înaintea realizării comparației propriu-zise.

Creșterea inteligibilității transcrierilor are în vedere operații de formatare a numerelor (transformarea numerelor scrise cu litere în numere formate cu cifre), inserare de litere mari, subliniere de URL-uri și inserare de semne de punctuație. Operațiile vor fi aplicate asupra unor transcrieri (neformate) de fișiere audio. Operațiile vor fi realizate folosind algoritmi bazați pe rețele neurale profunde.

Cele două componente vor fi integrate într-un serviciu web (REST) de transcriere de fișiere audio și identificare de cuvinte cheie și vor putea fi demonstrate printr-o aplicație web corespunzătoare.

3. Resurse folosite la dezvoltarea proiectului:

Limbaje de programare: Python, Java; Medii de dezvoltare: Netbeans, PyCharm; Cunoștințe de bază: Machine Learning.

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

ICVSI, TACAI, IVOM

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

6. Data înregistrării temei: 2018-11-29 14:56:44

Conducător(i) lucrare,

Conf. dr. ing. Horia CUCU

semnătura:

Student,

semnătura:

Responsabil program master,

Prof. dr. ing. Dragoș BURILEANU

semnătura:

Decan,

Prof. dr. ing. Cristian NEGRESCU

semnătura:

Cod Validare: **0f06cf33e0**

Declarație de onestitate academică

Prin prezenta declar că lucrarea cu titlul “ *Prelucrarea de limbaj natural folosind inteligență artificială: căutarea de cuvinte de interes și creșterea inteligibilității transcrierilor de vorbire*”, prezentată în cadrul Facultății de Electronică, Telecomunicații și Tehnologia Informației a Universității “Politehnica” din București ca cerință parțială pentru obținerea titlului de *Master* în domeniul *Inginerie Electronică și Telecomunicații*, programul de studii *BIOSINF* este scrisă de mine și nu a mai fost prezentată niciodată la o facultate sau instituție de învățământ superior din țară sau străinătate.

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

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

București, 18.06.2019

Manolache Cristian



(semnătura în original)

Cuprins

Cuprins	7
Lista de figuri	9
Lista de tabele	11
Lista de acronime	13
1 Introducere.....	15
1.1 Motivație	15
1.2 Scopul lucrării	15
1.3 Structura lucrării.....	15
2 Detecția de cuvinte de interes	17
2.1 Introducere	17
2.2 Metode de detecție de cuvinte de interes	17
2.2.1 Căutare literală de cuvinte	17
2.2.2 Căutare bazată pe DTW	17
2.2.3 Căutare bazată pe lematizare	18
2.3 Baza de date de cuvinte de interes	19
2.4 Rezultate experimentale	20
2.4.1 Metrice de performanță	20
2.4.2 Rezultate căutare literală.....	21
2.4.3 Rezultate căutare bazată pe DTW	21
2.4.4 Rezultate căutare bazată pe lematizare	23
2.4.5 Rezultate transcrieri sistem RAV.....	26
3 Creșterea inteligibilității transcrierilor de vorbire	31
3.1 Introducere	31
3.2 Generalități despre rețele neurale recurente	32
3.3 Metode de creștere a inteligibilității.....	32
3.4 Metoda propusa	34
3.5 Rezultate experimentale	36
3.5.1 Setul de date utilizat și preprocesari	36
3.5.2 Rezultate	36
4 Baze de date de vorbire pentru limba română	37
4.1 Motivație	37

4.2	Modalitate de achiziție	37
4.3	Aplicații software și procesări de date	37
4.4	Organizarea bazei de date	40
4.5	Baze de date pentru limba română	41
5	Infrastructura web de transcriere de fișiere audio și căutare de cuvinte de interes	43
5.1	Arhitectura sistemului integrat	43
5.2	Funcționalitate	44
5.3	Infrastructura serviciului web	44
5.3.1	Baza de date și sistemul de partajare de date în rețea	44
5.3.2	Cozi de mesaje și sarcini	47
5.4	Componente de procesare de date	48
5.5	Interfețe cu utilizatorul	51
6	Concluzii	57
6.1	Concluzii generale	57
6.2	Contribuții personale	57
6.3	Dezvoltări viitoare	57
7	Bibliografie	58

Lista de figuri

Figura 3.1 Rețea neurală recurentă desfășurată [4]	32
Figura 3.2 Arhitectura rețelei neurale pentru restaurare de litere mari	35
Figura 5.1 Arhitectura “Speed Web Services”	43
Figura 5.2 Arhitectura bazei de date “Speed Web Services”	47
Figura 5.3 Interfață vizuală pentru serviciul de transcriere, pagina “Transcriptions”	52
Figura 5.4 Interfață vizuală pentru serviciul de transcriere, pagina “View Transcription”	53
Figura 5.5 Interfață vizuală pentru serviciul de transcriere, pagina “My Account”	54
Figura 5.6 Interfață vizuală pentru serviciul de ICI, pagina “Keyword Spotting”	55
Figura 5.7 Interfață vizuală pentru serviciul de ICI, pagina “View KWS”	55
Figura 5.8 Interfață vizuală pentru serviciul de restaurare de diacritice, pagina “Home”	56

Lista de tabele

Tabel 2.1 Listă de cuvinte de interes	20
Tabel 2.2 Rezultate experimentale CLF și CLI (transcrieri de referință)	21
Tabel 2.3 Rezultate experimentale căutare bazată pe DTW (transcrieri de referință)	22
Tabel 2.4 Rezultate experimentale CLM (transcrieri de referință)	23
Tabel 2.5 Rezultate experimentale CLM+DTW (transcrieri de referință)	25
Tabel 2.6 Rezultate experimentale CLF și CLI (transcrieri RAV)	26
Tabel 2.7 Rezultate experimentale căutare bazată pe DTW (transcrieri de RAV)	27
Tabel 2.8 Rezultate experimentale CLM (transcrieri RAV)	28
Tabel 2.9 Rezultate experimentale CLM+DTW (transcrieri de RAV)	29
Tabel 3.1 Exemplu 1 post-procesare	31
Tabel 3.2 Exemplu 2 post-procesare	31
Tabel 3.3 Acuratețe restaurare de litere mari	36
Tabel 3.4 Probabilitate de restaurare eșuată pentru restaurare de litere mari	36
Tabel 3.5 Probabilitate de alarmă falsă pentru restaurare de litere mari	36
Tabel 4.1 Seturi de date de vorbire în limba română [13]	41
Tabel 5.1 Pași de urmat pentru transcrierea în text a unui fișier audio	49
Tabel 5.2 Pași de urmat pentru identificare unor cuvinte de interes într-un fișier audio	50
Tabel 5.3 Pași de urmat pentru restaurarea de diacritice într-un fișier text	51

Lista de acronime

API – Application Programming Interface

CL – Căutare literară

CLF – Căutare literară de tip fix

CLI – Căutare literară de tip incluziv

CLM – Căutare bazată pe lematizare

DTW - Dynamic Time Warping

HTML - HyperText Markup Language

HTTP - Hypertext Transfer Protocol

IA – Inteligență artificială

ICI – Identificare de cuvinte de interes

JSON - JavaScript Object Notation

KWS – Keyword Spotting

LSTM - Long Short Term Memory

NLP – Natural Language Processing

PLN - Procesare a limbajului natural

RAV – Recunoaștere automată a vorbirii

REST - Representational State Transfer

RNR - Rețelele neurale recurente

SQL - Structured Query Language

URL - Uniform Resource Locator

1 Introducere

1.1 Motivație

Cea mai naturală formă de comunicare între oameni este vorbirea și datorită acestui fapt, în ultima perioadă s-a încercat implementarea acestei forme în tehnologiile pe care le folosim zi de zi. Aceste tehnologii includ mașini inteligente, case inteligente, telefoane mobile și altele. Cele mai notabile sisteme de recunoaștere automată a vorbirii (RAV) sunt Siri (Apple), Alexa (Amazon), Cortana (Microsoft) și Google Assistant. Aceste sisteme au capacitatea de a detecta cuvinte cheie ce au rolul de a atenționa sistemul că urmează să primească o comandă. În ultima vreme, sistemele de RAV s-au dezvoltat folosind inteligență artificială. Un alt exemplu de folosință a sistemelor RAV sunt sistemele Speech-to-Text în care se pot obține documente scrise prin dictare. Aceste documente scrise, vin de regulă într-un format “brut”, în care nu există semne de punctuație, litere mari etc. Pentru a crește inteligibilitatea acestor transcrieri, se folosesc componente software de formatare a textului. Ideile menționate mai sus stau la baza motivației acestei lucrări.

1.2 Scopul lucrării

Scopul acestei lucrări de dizertație constă în dezvoltarea a două componente software de procesare de limbaj natural (PLN), precum și în achiziția unei baze de date de vorbire:

- Componentă software de post-procesare pentru transcrierea vorbirii utilizând algoritmi bazați pe rețele neurale profunde
- Componentă software de căutare de cuvinte de interes
- Dezvoltarea serviciului “Speed Web Services”
- Bază de date de vorbire în limba română formată din fișiere audio și fișiere text ce conțin transcrierile fișierelor audio

Componenta de post-procesare de text va înlocui componenta actuală de post-procesare din serviciul “Speed Web Services” a laboratorului de cercetare “Speech and Dialogue (Speed)”.

Componenta software de post-procesare va efectua operații de formatare de numere (trecere din litere în cifre), inserție de litere mari, subliniere de URL-uri și inserare de semne de punctuație asupra unor transcrieri neformatate de fișiere audio, folosind rețele neurale.

Componenta software de căutare de cuvinte de interes are ca scop identificarea de cuvinte în transcrierile audio folosind un algoritm de căutare bazat pe efectuarea unei comparații între cuvinte și luarea deciziei pe baza unui prag, precum și un algoritm de căutare bazat pe lematizare.

Serviciul web “Speed Web Services” de transcriere de fișiere audio și căutare de cuvinte cheie înglobează componentele de procesare menționate mai sus, dar și alte componente precum o componentă de diarizare și o componentă de transcriere, care nu fac obiectul acestei lucrări. Serviciul web oferă funcționalitățile menționate prin intermediul unui protocol standard (REST - HTTP).

Baza de date de vorbire va fi folosită pentru antrenarea componentei software de post-procesare, precum și pentru alte aplicații viitoare de inteligență artificială (IA).

1.3 Structura lucrării

Această lucrare este organizată în următorul fel. În capitolul 2 este prezentată componenta de detecție de cuvinte de interes ce face parte din aplicația de căutare de cuvinte cheie din “Speed

Web Services”; în acest capitol sunt prezentate metodele folosite, baza de date de testare, precum și rezultatele experimentale în diferite cazuri. Capitolul 3 prezintă componenta de creștere a inteligibilității transcrierilor, descriind generalități de IA, metoda folosită și rezultate experimentale. În capitolul 4 este descrisă metoda de achiziție a bazei de date de vorbire împreună cu diversele problemele întâlnite și o comparație cu baze de date existente. Capitolul 5 arată infrastructura serviciilor web ce înglobează cele 2 componente de procesare de date. În final, capitolele 6 și 7 sunt rezervate pentru concluzii și bibliografie.

2 Detecția de cuvinte de interes

2.1 Introducere

Detecția cuvintelor de interes (Keyword Spotting- KWS) se ocupă, în contextul procesării vorbirii, cu identificarea de cuvinte de interes (ICI) în vorbire. Detecția de cuvinte de interes se poate realiza atât direct pe fișiere audio, cât și pe transcrierea fișierelor audio. Această lucrare tratează cazul detecției cuvintelor în transcrierile fișierelor audio. Cuvintele de interes reprezintă fie cuvinte individuale (ex. tren, ambulanță, București) sau secvențe de cuvinte (ex. Președintele României, Porțile de Fier) pe care dorim să le identificăm într-un document audio. Identificarea cuvintelor de interes în transcriere presupune, în principiu, căutare de text în text. Acest lucru nu este banal întrucât cuvintele de interes căutate apar de cele mai multe ori sub diverse forme în transcrierea text a vorbirii. De exemplu, utilizatorul ar putea căuta cuvântul “tren” într-un fișier audio a cărui transcriere este “trenul a ajuns în gară”. În consecință, am experimentat cu mai mulți algoritmi de căutare cu scopul găsirii variantei optime care să identifice corect, conceptual, cuvintele de interes căutate. Aplicația ICI este o componentă integrată a serviciului “Speed Web Services” dezvoltată în limbajul de programare Java.

2.2 Metode de detecție de cuvinte de interes

Algoritmii ICI au fost dezvoltați în mai multe etape. Pentru detecția cuvintelor de interes s-au folosit trei algoritmi. Mai jos sunt descriși acești algoritmi împreună cu avantajele și dezavantajele acestora.

2.2.1 Căutare literală de cuvinte

În primă fază, s-a folosit o căutare literală (CL) de cuvinte ce presupune preluarea transcrierilor segmentelor audio și realizarea unei comparații între cuvântul căutat și cuvintele din transcrieri, numită și CL de tip fix (CLF). CLF caută cuvântul sau cuvintele din transcrieri care sunt identice cu cuvântul căutat. Deși această metodă poate fi considerată o opțiune rezonabilă pentru limbi precum engleza, aceasta nu se pretează la limbi bogate din punct de vedere morfologic precum româna, franceza sau germana. De exemplu, CLF nu va putea identifica cuvintele de interes “băiat” sau “mașină” în propoziția “i-am dat băiatului trei mașini”. Astfel, acest tip de căutare ne permite să identificăm cuvântul căutat, dar nu și forme flexionate ale acestuia.

O variantă improvizată a CLF constă în folosirea CL de tip incluziv (CLI). În această variantă, se caută cuvinte ce conțin cuvântul de interes. CLI va putea identifica cuvântul “băiat” în exemplul de mai sus, deoarece este inclus în cuvântul “băiatului”, dar nu va putea identifica cuvântul “mașină” ca fiind forma singulară a cuvântului “mașini”. De asemenea, CLI este predispus să producă alarme false, în special în cazul cuvintelor scurte ce pot fi incluse în cuvintele din transcrieri.

Este important de notat că metoda folosită nu este capabilă să identifice secvențe de cuvinte de interes.

2.2.2 Căutare bazată pe DTW

Al doilea algoritm implementat are la bază o formă a algoritmului DTW (Dynamic Time Warping), un algoritm standard de măsurare a similarității a două secvențe temporale ce ar putea avea dimensiuni diferite. În acest caz, algoritmul DTW a fost folosit pentru a compara 2 cuvinte între ele, anume cuvântul de interes și cuvintele din transcrieri, urmând să ia o decizie în privința identificării cuvântului de interes pe baza unui prag. Măsurarea similarității se realizează prin alinierea optimă a caracterelor termenului căutat cu caracterele cuvintelor ce formează fereastra din transcrierea RAV. Dacă secvență de cuvinte de interes are N cuvinte, atunci aceasta este

comparată cu N cuvinte consecutive din transcriere. O altă abordare ar fi în următorul fel: dacă secvența de cuvinte de interes are N caractere, se selectează din transcriere M cuvinte consecutive (conținând P caractere) astfel încât N are o valoare apropiată de P. Dar pentru acest tip de comparație nu s-au efectuat experimente.

Regulile după care se realizează alinierea sunt următoarele:

- toate caracterele din termenul căutat sunt puse în corespondență cu unul din caracterele din fereastra de transcriere RAV (și vice-versa);
- primul și ultimul caracter din termenul căutat trebuie să fie puse în corespondență cu primul, respectiv ultimul caracter din fereastra de transcriere RAV ;
- corespondența trebuie să respecte ordinea cronologică.

La ieșirea algoritmului se obține alinierea ce minimizează o funcție de cost. Această funcție de cost însumează toate diferențele la nivel de caracter și normalizează suma la lungimea termenului cel mai lung dintre cele două cuvinte. Decizia privind identificarea sau nu a cuvântului de interes se ia pe baza unui prag estimat empiric; acesta luând valori între 0 și 1. Dacă valoarea este sub pragul stabilit, s-a realizat o identificare. De exemplu, în cazul cuvintelor “ligament” vs. “ligamentele”, aveam o diferență de 3 caractere. Funcția de cost va avea valoarea $3/11 = 0.27$. Dacă avem un prag de 0.3 sau mai mare, algoritmul va semnaliza o identificare. Pentru praguri de 0.2 sau mai mici, algoritmul va fi mult mai exigent și în consecință, în cazul exemplului anterior, va susține ca nu a găsit cuvântul de interes. Pragul de valoare 0 înseamnă că nu trebuie să existe nici-o diferență între cuvintele comparate, acesta fiind echivalent cu CLF. În anumite cazuri precum “om” vs. “omul” și “ac” vs. “acul”, unde ne-am aștepta la o identificare a cuvântului de interes, funcția de cost are valoarea $2/4 = 0.5$. Pentru a obține răspunsul așteptat de la algoritm, ar fi necesar un prag de 0.5 sau mai mare, dar un astfel de prag ar ridica multe alarme false (de exemplu: “bugetar” vs. “bucata”, “utopie” vs. “datorie”). Algoritmul implementat va putea identifica de asemenea și secvențe de cuvinte de interes. O variantă simplificată pentru algoritmul DTW este prezentată mai jos:

```
found = false
for word in words
    cost = dtw(keyword, word)
    if(cost < threshold)
        found = true
    endif
endfor
```

2.2.3 Căutare bazată pe lematizare

În final, am folosit un algoritm de lematizare a cuvintelor. În această situație fiecare cuvânt (atât cuvântul cheie cât și cele din transcriere) este adus la forma lui de bază (lemă), printr-un algoritm de PLN (natural language processing - NLP) ce realizează o analiză morfologică a cuvintelor, după care se va realiza comparația dintre leme. Pentru exemplificare, presupunem că ne dorim să căutăm cuvântul cheie “ac” în transcrierea “caut acul în carul cu fân”. Cuvântul cheie fiind “ac”, forma lui de bază (lematizată) este tot “ac”. Forma lematizată a transcrierii este “căuta ac în car cu fân”. După obținerea formelor lematizate pentru cuvântul cheie și transcriere, printr-o simplă comparație între cuvinte se poate identifica cu ușurință cuvântul “ac”. Procesul de lematizare al cuvintelor a fost realizat cu ajutorul framework-ului open-source NLP-Cube [1]. Deoarece framework-ul NLP-Cube este scris în Python, iar aplicația ICI este scrisă în Java, a fost

neesecară o metodă de comunicare între cele 2 componente. Astfel, cuvintele de interes și transcrierile sunt transmise între cele 2 componente în format JSON cu ajutorul framework-ului RabbitMQ [2].

2.3 Baza de date de cuvinte de interes

Pentru a testa și evalua sistemul ICI a fost necesară o bază de date de cuvinte de interes. La ora actuală nu există o astfel de bază de date pentru limba română, în consecință, a fost necesar să se creeze una pornind de la baza de date RSC [3] ce conține transcrieri din știri, interviuri și literatură. În această bază de date există 2 seturi de transcrieri, un set de transcrieri de referință și un set de transcrieri obținut la ieșirea unui sistem de RAV ce este utilizat și în serviciul “Speed Web Services”. Fiecare set are un număr total de 2504 transcrieri. Analizând manual transcrierile s-au extras 100 cuvinte de interes având în vedere următoarele criterii:

- cuvântul a fost transcris corect de sistemul RAV și are aceeași formă în transcriere
- cuvântul a fost transcris corect de sistemul RAV, dar sistemul ICI nu l-a identificat
- cuvântul nu apare în transcrierea obținută din sistemul RAV, dar sistemul ICI spune că l-a găsit
- cuvântul nu a fost transcris corect de sistemul RAV, în consecință sistemul ICI nu avea cum să-l identifice
- cuvântul nu există în fișierul audio, dar a fost scos la ieșirea sistemului RAV, în consecință sistemul ICI l-a identificat în mod eronat

Lista cuvintelor de interes este prezentată în Tabel 2.1:

Tabel 2.1 Listă de cuvinte de interes

obez	bugetar	poster	occident	competiție
polițist	tren	forțat	mogul	farfurie
fâlfâie	solicitare	favorit	natură	campanie
inutil	radio	ilie năstase	sincer	școală
vinovat	străin	reforma	profit	atacator
fotografie	prim ministru	ac	utopie	sacrificiu
veteran	umor	sănătos	titlu	jurnalist
mașină	lege	glob	capitalism	solidar
bucată	oficial	hotel	entitate	autocar
consumat	reformă	idee	curajos	justiție
garnitură	raport	câștig	număr	umăr
românia	veteran	liberal	persecutor	sediu
economie	europa	protest	specialist	tăiere
sentiment	douăzeci	comunism	disperare	dragomir
expoziție	sud-american	organism	vot	cuvânt
cancer	diabet	frig	fereastră	cultură
râu	strigăt	unanimitate	întrebare	săpun
majoritate	știință	zonă	decizie	împotrivire
țară	mihai răzvan	click	colectiv	imobil
bancă	zilnic	lumină	atitudine	român

Împreună cu cuvintele de interes au fost notate și id-urile transcrierilor în care acestea apar.

2.4 Rezultate experimentale

Experimentele efectuate în vederea evaluării sistemului ICI au constatat în testarea sistemului folosind lista de cuvinte de interes menționată anterior și transcrierile din baza de date RSC. Mai jos sunt descrise metricile de performanță folosite precum și rezultatele experimentale pentru transcrierile de referință (caz ideal de transcrieri) și transcrierile obținute la ieșirea sistemului RAV (caz real de transcrieri).

2.4.1 Metrici de performanță

Pentru evaluarea sistemului de căutare de cuvinte de interes, s-au folosit următoarele metrici:

- Probabilitate de neidentificare; această metrică verifică de câte ori cuvântul de interes ar fi trebuit să fie găsit, dar nu a fost. Pentru fiecare cuvânt avem o listă de transcrieri de referință, unde ar trebui să fie găsit cuvântul și o listă de transcrieri în care a fost găsit de sistemul ICI. Numărul de transcrieri ce sunt în lista de referință dar nu se regăsesc în lista de transcrieri generată de sistemul ICI reprezintă numărul de *false negatives*. Probabilitatea de rată este calculată astfel:

$$\text{Miss} = \text{False Negatives} / \text{size}(\text{Reference List}) * 100;$$

Pentru a obține probabilitatea globală, se calculează media aritmetică a probabilităților tuturor cuvintelor de interes.

- Probabilitate de alarmă falsă; similar cu probabilitatea de neidentificare, aceasta verifică de câte ori cuvântul de interes a fost identificat în mod eronat. Numărul de transcrieri ce este găsit în lista obținută la ieșirea sistemului ICI dar nu și în lista de transcrieri de referință reprezintă numărul de *false positives*. Probabilitatea de alarmă falsă este calculată astfel:

$$\text{False Alarm} = \text{False Positives} / \text{size}(\text{Found List}) * 100;$$
- F-measure; cunoscut și ca F-score sau F1-score este o măsură a acurateții. Această metrică folosește Precision și Recall pentru a obține scorul final și este calculată astfel:

$$\text{Precision} = 1 - \text{False Alarm};$$

$$\text{Recall} = 1 - \text{Miss};$$

$$\text{F-measure} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$$

2.4.2 Rezultate căutare literală

În această secțiune este evaluat algoritmul CL pe transcrierile de referință ale rostirilor din setul de date RSC. Rezultatele pentru algoritmi CLF și CLI sunt prezentate în tabelul de mai jos. Ipoteza experimentală a fost că CLF nu va putea identifica toate cuvintele de interes din cauza faptului că acestea sunt găsite în transcrieri în diverse forme morfologice, în timp ce CLI va reuși să identifice mai multe cuvinte în schimbul unor alarme false. Rezultatele se afla în Tabel 2.2.

Tabel 2.2 Rezultate experimentale CLF și CLI (transcrieri de referință)

Nr	Sistem RAV	Algoritm ICI	Tip	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
1.1	ideal (REF trans)	CLF	EM	57.0	0.0	43.0	100.0	60.2
1.2	ideal (REF trans)	CLI	IM	27.5	3.0	72.5	97.0	83.0

Rezultatele experimentale ne validează ipoteza inițială. Într-adevăr, CLF are foarte multe neidentificări (57%), dar nu produce nici-o alarmă falsă. CLI prezintă un procentaj de neidentificare mai mic (27.5%) și o rată de alarme false mica (3%). Comparând cei 2 algoritmi după scorul F-measure, putem trage concluzia că algoritmul CLI este semnificativ mai bun decât CLF. Mai jos sunt prezentate câteva exemple în care algoritmul CLF a eșuat, dar CLI a realizat o identificare corectă.

Termen: “**occident**” în transcrierea: “acum suntem **occidentali**”

Termen: “**tren**” în transcrierea: “are școală la propriu cât clasele **trenului** iar când a câștigat alegerile ...”

Termen: “**străin**” în transcrierea: “... într-o mașină în care se aflau jurnaliști **străini**”

Exemple când CLI a dat alarme false:

Termen: “**tren**” în transcrierea: “noul **antrenor** al stelei este în pericol să fie demis ...”

Termen: “**lege**” în transcrierea: “... iar când a câștigat **alegerile** ...”

Termen: “**imobil**” în transcrierea: “... doi polițiști l-au **imobilizat** și l-au trântit”

De asemenea, cuvântul “ac” a fost de multe ori identificat eronat în interiorul altor cuvinte, cazuri care în mod ideal nu ar fi trebuit să se întâmple.

2.4.3 Rezultate căutare bazată pe DTW

Mai jos sunt prezentate rezultatele pentru algoritmul DTW testat pe același set de date ca și algoritmul CL din subcapitolul anterior. Algoritmul a fost testat pe un set de praguri între 0 și 0.9.

Ne așteptăm ca cele mai bune rezultate să se găsească la un prag cu o valoare situată între 0.2 și 0.4, iar pentru valori mai mari de 0.5 să primim multe alarme false.

Tabel 2.3 Rezultate experimentale căutare bazată pe DTW (transcrieri de referință)

Nr	Sistem RAV	Algoritm ICI	Prag	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
2.1	ideal (REF trans)	DTW	0.0	55.0	0.0	45.0	100.0	62.1
2.2	ideal (REF trans)	DTW	0.1	51.7	0.0	48.3	100.0	65.1
2.3	ideal (REF trans)	DTW	0.2	37.5	1.7	62.5	98.3	76.4
2.3.1	ideal (REF trans)	DTW	0.25	26.0	2.1	74.0	97.9	84.3
2.3.2	ideal (REF trans)	DTW	0.26	20.0	3.7	80.0	96.3	87.4
2.3.3	ideal (REF trans)	DTW	0.27	20.0	3.7	80.0	96.3	87.4
2.3.4	ideal (REF trans)	DTW	0.28	18.6	3.7	81.4	96.3	88.2
2.3.5	ideal (REF trans)	DTW	0.29	14.9	6.8	85.1	93.2	88.9
2.4	ideal (REF trans)	DTW	0.3	14.9	6.8	85.1	93.2	88.9
2.4.1	ideal (REF trans)	DTW	0.31	14.5	8.4	85.5	91.6	88.4
2.4.2	ideal (REF trans)	DTW	0.32	14.5	8.4	85.5	91.6	88.4
2.4.3	ideal (REF trans)	DTW	0.33	14.5	8.4	85.5	91.6	88.4
2.4.4	ideal (REF trans)	DTW	0.34	11.9	15.8	88.1	84.2	86.1
2.4.5	ideal (REF trans)	DTW	0.35	11.9	15.8	88.1	84.2	86.1
2.5	ideal (REF trans)	DTW	0.4	9.4	23.0	90.6	77.0	83.3
2.6	ideal (REF trans)	DTW	0.5	4.0	45.9	96.0	54.1	69.2
2.7	ideal (REF trans)	DTW	0.6	0.1	87.4	99.9	12.6	22.3
2.8	ideal (REF trans)	DTW	0.7	0.0	95.6	100.0	4.4	8.4
2.9	ideal (REF trans)	DTW	0.8	0.0	97.8	100.0	2.2	4.3
2.10	ideal (REF trans)	DTW	0.9	0.0	98.3	100.0	1.7	3.4

În Tabel 2.3 se poate observa că cele mai bune rezultate sunt găsite pentru o valoare a pragului de 0.3. S-a încercat de asemenea și o incrementare/decrementare a pragului cu 0.01 în jurul valorii 0.3 în speranța de a găsi rezultate mai bune. În schimb, alarmele false au un procentaj ridicat (23%) chiar de la o valoare a pragului de 0.4 și procentaje semnificativ mai ridicate pentru valori mai mari de acest prag. Este de notat faptul că algoritmul DTW obține rezultate puțin mai bune (62.1%) la o valoare a pragului de 0 față de algoritmul CLF (60.2%). Teoretic, acești 2 algoritmi ar fi trebuit să obțină un scor identic; această lucră este datorată faptului că algoritmul DTW este capabil să identifice secvențe de cuvinte precum “prim ministru”, pe când algoritmul CL nu. În concluzie, algoritmul DTW obține rezultate mai bune (88.9%) față de algoritmul CLI (83%). Mai jos sunt prezentate câteva exemple unde căutarea bazată DTW cu prag de 0.3 a reușit să identifice cu succes cuvintele de interes dar CLI nu:

Termen: “**competiție**” în transcrierea: “la o dată care va fi stabilită de departamentul de **competiții** al lpf”

Termen: “**polițist**” în transcrierea: “n-a apucat să facă asta pentru că doi **polițiști** l-au imobilizat și l-au trântit”

Termen: “**campanie**” în transcrierea: “... pentru că dă bine în **campania** electorală”

Exemplu de neidentificare a algoritmului DTW:

Termen: “**polișist**” în transcrierea: “**polișistii** care se ocupau de atacatorul nostru lucrau în tăcere”

Cazuri în care algoritmul DTW a dat greș dar algoritmul CLI a identificat corect :

Termen: “**tren**” în transcrierea: “are școală la propriu cât clasele **trenului**”

Având în vedere alarmele false, algoritmul DTW are mai multe (6.8%) decât algoritmul CLI (3%). Exemple:

Termen: “**tren**” în transcrierea: “... articolul o sută douăzeci și **trei** din regulamentul disciplinar ...”

Termen: “**mogul**” în transcrierea: “**modul** în care poate fi făcut acest lucru ține de fiecare guvern”

Termen: “**inutil**” în transcrierea: “... meciul universitatea cluj cfr cluj întrerupt marți în **minutul** douăzeci și șapte”

2.4.4 Rezultate căutare bazată pe lematizare

Rezultatele experimentale pentru căutarea bazată pe lematizare (CLM) sunt prezentate în Tabel 2.4. În primă fază s-a încercat aplicarea unei CLF după aplicarea procesului de lematizare a cuvintelor (atât cuvintele de interes cât și cuvintele din transcrieri au fost lematizate). S-a implementat de asemenea și o actualizare a procesului de lematizare în privința procesării numelor proprii. Aceasta constă în înlocuirea terminației “ei” cu “a” (din “României” în “România”) și eliminarea terminației “ului” (din “Brașovului” în “Brașov”) în cazul numelor proprii. Această implementare a fost necesară deoarece framework-ul NLP-Cube nu o efectua.

Tabel 2.4 Rezultate experimentale CLM (transcrieri de referință)

Nr	Sistem RAV	Algoritm ICI	Procesare nume proprii	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
3.1	ideal (REF trans)	CLM	da	25.6	0.0	74.4	100.0	85.3
3.2	ideal (REF trans)	CLM	nu	27.1	0.0	72.9	100.0	84.3

Având în vedere datele din tabelul de mai sus, se poate observa că algoritmul de lematizare cu procesare de nume proprii are un scor puțin mai bun decât varianta standard, dar, surprinzător, acest scor (85.3%) este mai mic decât cel al algoritmului DTW cu prag de 0.3 (88.9%).

Este de menționat faptul că procesul de creare a bazei de date de cuvinte de interes a fost subiectiv, de exemplu, în cazul căutării cuvântului de interes “**economie**” în transcrierea “pentru că a asigurat că politicile **economice** ale româniei vor continua în cadrul acordului cu fmi și ce”, ne-am fi așteptat ca “**economie**” să fie identificat în termenul “**economice**” din transcriere. După lematizare, cuvântul de interes a rămas “**economie**”, iar transcrierea a devenit “pentru că avea asigura că politică **economic** al românia vrea continua în cadru acord cu fmi și ce”; deși cuvintele au fost aduse la forma lor de bază, după aplicarea unei căutări de tip CLF nu s-a putut identifica cuvântul de interes. Au existat de asemenea și erori ale componentei de lematizare precum în cazul căutării cuvântului de interes “**mogul**” în transcrierea “cu credibilitatea minus infinit pe care o are împreună cu alți câțiva **moguli** ai capitalismului de partid și de stat omul e în stare să îngroape orice chiar o idee banală doar pronunțându-se pentru”; după lematizare, cuvântul de interes a

devenit “**mog**”, iar transcrierea a devenit “cu credibilitate minus infinit pe care el avea împreună cu alți câțiva **mogul** al capitalism de partid și de stat om fi în stare să îngropa orice chiar un idee banal doar pronunța sine pentru”. Astfel, având în vedere exemple date mai sus, se poate înțelege scorul mai mic al căutării bazate pe lematizare față de căutarea bazată pe DTW. Câteva exemple în care căutarea bazată DTW cu prag de 0.3 a identificat cu succes cuvintele de interes, dar CLM nu:

Termen: “**occident**” (lematizat: “**occident**”) în transcrierea: “acum suntem **occidentali**” (lematizată: “acum fi **occidental**”)

Termen: “**fotografie**” (lematizat: “**fotografie**”) în transcrierea: “am coborât din autocar val vârtej și am început să **fotografiez**” (lematizată: “avea coborî din autocar val vârtej și avea începe să **fotografia**”)

Cazuri în care căutarea bazată pe DTW a dat greș dar CLM a identificat corect:

Termen: “**reforma**” (lematizat: “**reformă**”) în transcrierea: “acordul cu fmi indică ceea ce românia are de făcut atât în privința consolidării fiscale și **reformelor** structurale” (lematizată: “acord cu fmi indica acela ce românia avea de face atât în privințe consolidare fiscal și al **reformă** structural”)

Termen: “**lege**” (lematizat: “**lege**”) în transcrierea: “este adoptarea în unanimitate de către parlamentul româniei a **legii** care implementează noul cod de procedură civilă” (lematizată: “fi adoptare în unanimitate de către parlament românia al **lege** care implementa nou cod de procedură civil”)

În Tabel 2.4 putem observa faptul că nu există alarme false pentru CLM, căutarea bazată pe DTW cu prag de 0.3 având în schimb un procentaj de 6.8%. Astfel, există cazuri în care ambele căutări au identificat corect, dar căutarea DTW cu prag de 0.3 a dat și alarme false:

Termen: “**inutil**” confundat cu “**minutul**” în transcrierea: “să se rejoyce începând din minutul unu la o dată ce urmează să fie stabilită ulterior”

Termen: “**umor**” confundat cu “**unor**” în transcrierea: “nu întâmplător chemarea la sacrificiu a luat forma unui îndemn civic după ce unor generații succesive li s-a cerut să strângă cureaua disponibilitatea lor pentru sacrificiu este aproape zero”

Termen: “**tren**” confundat cu “**trei**” în transcrierea: “în cazul în care steaua nu va lua cele trei puncte becali e hotărât să îl dea afară”

Cazuri în care ambele tipuri de căutări au dat greș:

Termen: “**glob**” (lematizat: “**glob**”) în transcrierea: “acea datorie pe care a ratat-o sistematic în ultimul an ignorând din incompetență sau interes electoral prăbușirea economiilor **globale**” (lematizată: “acel datorie pe care avea ratael sistematic în ultimul an ignora din incompetență sau interes electoral prăbușire economie **global**”)

Termen: “**decizie**” (lematizat: “**decizie**”) în transcrierea: “pdl a **decis** depunerea unei moțiuni simple în cazul ministrului educației ioan mang acuzat de plagiat” (lematizată: “pdl avea **decide** depunere un moțiune simplu în caz ministru educație ioan mang acuza de plagiat”)

În consecință s-a încercat o combinație a procesului de căutare bazată pe lematizare cu procesul de căutare bazată pe DTW (CLM+DTW). Pragurile au fost alese similar cu cele din experimentele pentru căutarea bazată pe DTW. În cazul acesta, ajustarea pragului

incrementând/decrementând cu 0.01 în jurul valorii cu scorul cel mai bun (prag 0.2 – 91.1%) ne-a permis să găsim un scor puțin mai bun (91.8%) pentru valorile 0.23, 0.24 și 0.25 ale pragului.

Tabel 2.5 Rezultate experimentale CLM+DTW (transcrieri de referință)

Nr	Sistem RAV	Algoritm ICI	Prag	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
4.1	ideal (REF trans)	CLM + DTW	0	25.6	0.0	74.4	100.0	85.3
4.2	ideal (REF trans)	CLM + DTW	0.1	25.6	0.0	74.4	100.0	85.3
4.2.1	ideal (REF trans)	CLM + DTW	0.15	19.0	0.3	81.0	99.7	89.4
4.2.2	ideal (REF trans)	CLM + DTW	0.16	19.0	0.3	81.0	99.7	89.4
4.2.3	ideal (REF trans)	CLM + DTW	0.17	17.0	0.6	83.0	99.4	90.5
4.2.4	ideal (REF trans)	CLM + DTW	0.18	17.0	0.6	83.0	99.4	90.5
4.2.5	ideal (REF trans)	CLM + DTW	0.19	16.0	0.6	84.0	99.4	91.1
4.3	ideal (REF trans)	CLM + DTW	0.2	16.0	0.6	84.0	99.4	91.1
4.3.1	ideal (REF trans)	CLM + DTW	0.21	14.7	2.3	85.3	97.7	91.1
4.3.2	ideal (REF trans)	CLM + DTW	0.22	14.7	2.3	85.3	97.7	91.1
4.3.3	ideal (REF trans)	CLM + DTW	0.23	13.3	2.6	86.7	97.4	91.8
4.3.4	ideal (REF trans)	CLM + DTW	0.24	13.3	2.6	86.7	97.4	91.8
4.3.5	ideal (REF trans)	CLM + DTW	0.25	13.3	2.6	86.7	97.4	91.8
4.3.6	ideal (REF trans)	CLM + DTW	0.26	11.0	5.8	89.0	94.2	91.5
4.4	ideal (REF trans)	CLM + DTW	0.3	8.7	9.7	91.3	90.3	90.8
4.5	ideal (REF trans)	CLM + DTW	0.4	6.5	28.5	93.5	71.5	81.1
4.6	ideal (REF trans)	CLM + DTW	0.5	2.0	51.6	98.0	48.4	64.8

Având în vedere rezultatele din Tabel 2.5, se poate observa că metoda CLM+DTW are cel mai bun scor (91.8%) de indentificare de cuvinte de interes dintre metodele aplicate, deși acesta produce un procentaj de 2.6% de alarme false față de CLM care are 0%. Mai jos sunt câteva exemple în care CLM+DTW a identificat corect, dar căutarea bazată pe DTW cu prag de 0.3 nu:

Termen: “**economie**”(lematizat: “**economie**”) în transcrierea: “acea datorie pe care a ratat-o sistematic în ultimul an ignorând din incompetență sau interes electoral prăbușirea **economiilor** globale” (lematizată: “acel datorie pe care avea rata el sistematic în ultimul an ignora din incompetență sau interes electoral prăbușire **economie** global”)

Termen: “**comunism**”(lematizat: “**comunism**”) în transcrierea: “românilor li s-a impus să se sacrifice pentru utopia societății socialiste multilateral dezvoltate iar după căderea

comunismului pentru construirea capitalismului dezirabil și tangibil” (lematizată: “român el sine avea impune să sine sacrifice pentru utopie societate socialist multilateral dezvoltat iar după cădere **comunism** pentru construire capitalism”)

Cazuri în care CLM+DTW a identificat corect față de CLM:

Termen: “**economie**”(lematizat: “**economie**”) în transcrierea: “pentru că a asigurat că politicile **economice** ale româniei vor continua în cadrul acordului cu fmi și ce” (lematizată: “pentru că avea asigura că politică **economic** al românia vrea continua în cadru acord cu fmi și ce”)

Termen: “**disperare**”(lematizat: “**disperare**”) în transcrierea: “acel om era un om **disperat**” (lematizată: “acel om fi un om **disperat**”)

2.4.5 Rezultate transcrieri sistem RAV

În subcapitolele de mai sus au fost prezentate rezultatele experimentale ale testelor efectuate pe transcrierile de referință ale setului de date RSC. În această secțiune sunt prezentate rezultatele experimentale ale testelor efectuate pe transcrierile obținute la ieșirea sistemului RAV. Comparativ cu rezultatele obținute pe transcrierile ideale, scorurile obținute în acest caz sunt mai mici; lucru care este de așteptat, având în vedere că în acest caz există și erori ale sistemului RAV pe lângă cele ale sistemului ICI (cuvântul nu a fost transcris corect de sistemul RAV, în consecință sistemul ICI nu avea cum să-l identifice; cuvântul nu există în fișierul audio, dar a fost scos la ieșirea sistemului RAV, în consecință sistemul ICI l-a identificat în mod eronat) ceea ce duce la procentaje mai ridicate de neidentificare și de alarme false.

Tabel 2.6 Rezultate experimentale CLF și CLI (transcrieri RAV)

Nr	Sistem RAV	Algoritm ICI	Tip	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
5.1	real (HYP trans)	CLF	EM	60.2	6.2	39.8	93.8	55.9
5.2	real (HYP trans)	CLI	IM	30.3	13.3	69.7	86.7	77.3

În Tabel 2.6 avem rezultatele experimentale pentru CLF și CLI efectuate pe transcrierile obținute la ieșirea sistemului RAV. Cel mai bun scor îl regăsim tot în cazul CLI (77.3%), dar acesta este mai mic față de scorul în cazul transcrierilor de referință (83%). Exemple de cazuri de neidentificare:

Termen: “**obez**” în transcrierea: “un stat deopotrivă salvator și persecutor și iubesc pe deasupra este doar în avantajul politicianilor care vor să acceadă sau au ajuns la putere” (transcriere de referință: “un stat deopotrivă salvator și persecutor și obez pe deasupra este doar în avantajul politicianilor care vor să acceadă sau au ajuns la putere”); în acest caz, termenul “obez” a fost confundat cu termenul “iubesc” de către sistemul RAV.

Termen: “**străin**” în transcrierea: “un om care a vrut să strige disperarea și ale să facă acest lucru aruncând cu o cărămidă într-o mașină în care se aflau jurnaliști trei” (transcriere de referință: “un om care a vrut să-și strige disperarea și a ales să facă acest lucru aruncând cu o cărămidă într-o mașină în care se aflau jurnaliști străini”); în acest caz, termenul “străini” nu a fost identificat de către sistemul RAV.

Exemple de cazuri de alarme false:

Termen: “**poster**” în transcrierea: “invitat la un poster at ea și ilie năstase i-a transmis un mesaj amenințător sau între traco mie după ce acesta l-a reclamat la dna în luna februarie” (transcriere de referință: “invitat la un post de radio ilie năstase i-a transmis un mesaj amenințător ruxandrei dragomir după ce aceasta l-a reclamat la dna în luna februarie”); în acest caz, termenul “post” a fost confundat cu termenul ”poster” de către sistemul RAV.

Termen: “**autocar**” în transcrierea: “polițiștii care se ocupau de autocarul nostru lucrau în tăcere” (transcriere de referință: “polițiștii care se ocupau de atacatorul nostru lucrau în tăcere”); în acest caz, termenul “atacatorul” a fost confundat cu termenul ” autocarul” de către sistemul RAV.

Tabel 2.7 Rezultate experimentale căutare bazată pe DTW (transcrieri de RAV)

Nr	Sistem RAV	Algoritm ICI	Prag	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
6.1	real (HYP trans)	DTW	0	57.2	6.2	42.8	93.8	58.8
6.2	real (HYP trans)	DTW	0.1	52.0	6.2	48.0	93.8	63.5
6.3	real (HYP trans)	DTW	0.2	33.9	10.1	66.1	89.9	76.2
6.3.1	real (HYP trans)	DTW	0.25	20.6	12.0	79.4	88.0	83.5
6.3.2	real (HYP trans)	DTW	0.26	20.6	12.0	79.4	88.0	83.5
6.3.3	real (HYP trans)	DTW	0.27	20.6	12.0	79.4	88.0	83.5
6.3.4	real (HYP trans)	DTW	0.28	19.4	12.1	80.6	87.9	84.1
6.3.5	real (HYP trans)	DTW	0.29	15.7	15.3	84.3	84.7	84.5
6.4	real (HYP trans)	DTW	0.3	15.2	16.9	84.8	83.1	84.0
6.4.1	real (HYP trans)	DTW	0.31	15.2	16.9	84.8	83.1	84.0
6.4.2	real (HYP trans)	DTW	0.32	15.2	16.9	84.8	83.1	84.0
6.4.3	real (HYP trans)	DTW	0.33	15.2	16.9	84.8	83.1	84.0
6.4.4	real (HYP trans)	DTW	0.34	12.9	25.5	87.1	74.5	80.3
6.4.5	real (HYP trans)	DTW	0.35	12.9	25.5	87.1	74.5	80.3
6.5	real (HYP trans)	DTW	0.4	9.3	38.8	90.7	61.2	73.1
6.6	real (HYP trans)	DTW	0.5	1.9	76.4	98.1	23.6	38.0
6.7	real (HYP trans)	DTW	0.6	0.4	90.6	99.6	9.4	17.2
6.8	real (HYP trans)	DTW	0.7	0.1	96.4	99.9	3.6	7.0
6.9	real (HYP trans)	DTW	0.8	0.0	98.0	100.0	2.0	3.9
6.10	real (HYP trans)	DTW	0.9	0.0	98.3	100.0	1.7	3.4

În Tabel 2.7 avem rezultatele experimentale pentru căutarea bazată pe DTW efectuate pe transcrierile obținute la ieșirea sistemului RAV. Cel mai bun scor îl găsim la pragul de 0.29 cu un scor de 84.5% față de cazul experimentelor realizate pe transcrierile de referință, unde cel mai bun scor îl găsim la pragul de 0.3 cu un scor de 88.9%. Exemple de cazuri de neidentificare:

Termen: “**titlu**” în transcrierea: “adică încontinuu de campion în clasamentul neoficial” (transcriere de referință: “adică încă un titlu de campion în clasamentul neoficial”); în acest caz, termenii “încă un titlu” au fost confundați cu termenul ” încontinuu” de către sistemul RAV.

Termen: “**oficial**” în transcrierea: “eram la capătul unei zile în care bătusem la pas expoziția mondială cu câteva ore înainte de deschiderea” (transcriere de referință: “eram la capătul unei zile în care bătusem la pas expoziția mondială cu câteva ore înainte de deschiderea oficială”); în acest caz, termenul “oficială” nu a fost identificat de către sistemul RAV.

Tabel 2.8 Rezultate experimentale CLM (transcrieri RAV)

Nr	Sistem RAV	Algoritm ICI	Procesare nume proprii	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
7.1	real (HYP trans)	CLM	da	28.0	10.4	72.0	89.6	79.9
7.2	real (HYP trans)	CLM	nu	29.5	10.4	70.5	89.6	78.9

În Tabel 2.8 sunt prezentate rezultatele experimentale pentru CLM efectuate pe transcrierile obținute la ieșirea sistemului RAV. Și în acest caz se poate observa o scădere a scorului (79.9%) față de varianta în care experimentele s-au realizat pe transcrierile de referință (85.3%). Exemplu de neidentificare:

Termen: “**râu**” (lematizat: “**râu**”) în transcrierea “dacă mâine va zice că apa curge la vale toată țara se va aștepta careurile soia la deal” (lematizată: “dacă mâine vrea zice că apă curge la vale tot țară sine vrea aștepta careu soi la deal”) ce are ca transcriere de referință: “dacă mâine va zice că apa curge la vale toată țara se va aștepta ca râurile s-o ia la deal”.

Tabel 2.9 Rezultate experimentale CLM+DTW (transcrieri de RAV)

Nr	Sistem RAV	Algoritm ICI	Prag	Miss [%]	False Alarm [%]	Recall (1 - Miss)	Precision (1 - FA)	F-measure
8.1	real (HYP trans)	CLM + DTW	0	28.0	10.4	72.0	89.6	79.9
8.2	real (HYP trans)	CLM + DTW	0.1	25.6	10.4	74.4	89.6	81.3
8.3	real (HYP trans)	CLM + DTW	0.2	16.3	11.6	83.7	88.4	86.0
8.3.1	real (HYP trans)	CLM + DTW	0.25	12.1	14.3	87.9	85.7	86.8
8.3.2	real (HYP trans)	CLM + DTW	0.26	12.1	14.3	87.9	85.7	86.8
8.3.3	real (HYP trans)	CLM + DTW	0.27	12.1	14.3	87.9	85.7	86.8
8.3.4	real (HYP trans)	CLM + DTW	0.28	11.7	14.5	88.3	85.5	86.9
8.3.5	real (HYP trans)	CLM + DTW	0.29	9.7	17.2	90.3	82.8	86.4
8.4	real (HYP trans)	CLM + DTW	0.3	9.6	17.7	90.4	82.3	86.2
8.4.1	real (HYP trans)	CLM + DTW	0.31	9.6	18.2	90.4	81.8	85.9
8.4.2	real (HYP trans)	CLM + DTW	0.32	9.6	18.2	90.4	81.8	85.9
8.4.3	real (HYP trans)	CLM + DTW	0.33	9.6	18.2	90.4	81.8	85.9
8.4.4	real (HYP trans)	CLM + DTW	0.34	7.1	29.9	92.9	70.1	79.9
8.4.5	real (HYP trans)	CLM + DTW	0.35	7.1	29.9	92.9	70.1	79.9
8.5	real (HYP trans)	CLM + DTW	0.4	4.2	43.4	95.8	56.6	71.2
8.6	real (HYP trans)	CLM + DTW	0.5	0.9	77.6	99.1	22.4	36.5

În Tabel 2.9 sunt prezentate rezultatele experimentale pentru CLM efectuate pe transcrierile obținute la ieșirea sistemului RAV. În acest caz, cel mai bun scor se găsește la pragul de 0.28 (86.9%) față de pragul de 0.24 (91.8%) în cazul experimentelor realizate pe transcrierile de referință. Exemplu de neidentificare:

Termen: “**întrebare**” (lematizat: “**întrebare**”) în transcrierea “un soi de război fratricid bugetari particulari a înghițit tot acest peisaj mutând miza reală a dezbaterii pentru baza căreia nimeni nu-i va putea oferi un răspuns coordonat ce ar trebui să plătească efectele recesiunii” (lematizată: “un soi de război fratricid bugetar particular avea înghiți tot acest peisaj muta miză real al dezbaterie pentru bază care nimeni nuel vrea putea oferi un răspuns coordonat ce avea trebui să plăti efect recesiune”) ce are ca transcriere de referință: “un soi de război fratricid bugetari-particulari a înghițit tot acest peisaj mutând miza reală a dezbaterii pe întrebarea căreia nimeni nu- i va putea oferi un răspuns coordonat cine ar trebui să plătească efectele recesiunii”.

3 Creșterea inteligibilității transcrierilor de vorbire

3.1 Introducere

Ieșirea unui sistem RAV constă într-un text “brut”, ce conține de obicei numai litere mici și fără semne de punctuație.

Creșterea inteligibilității transcrierilor de vorbire se referă la preluarea transcrierilor neformate ale fișierelor audio și efectuarea unor operații asupra acestora precum:

- Transformarea numerelor și datelor scrise cu litere în numere și date scrise cu cifre (de exemplu, expresia “douăzeci și doi” este transformată în “22”)
- Inserarea de semne de punctuație (virgule și puncte)
- Înlocuirea literelor mici cu litere mari acolo unde este cazul: început de propoziție, nume proprii, acronime etc.

Pentru a pune în evidență diferența dintre textul neformatat și cel formatat, prezentăm mai jos 2 exemple.

Tabel 3.1 Exemplu 1 post-procesare

Neformatat	după injecția de lichidități și efectele nu se văd și nu se resimt asupra roborului după cum puteți vedea din nou robor a crescut cu zero virgulă zero unu la sută față de săptămâna trecută iar mâine banca centrală va avea o decizie grea de luat dacă va interveni sau nu cu noi lichidități pe piață
Formatat	După injecția de lichidități și efectele nu se văd și nu se resimt asupra ROBOR-ului, după cum puteți vedea din nou ROBOR a crescut cu 0 , 1% față de săptămâna trecută , iar mâine banca centrală va avea o decizie grea de luat , dacă va interveni sau nu cu noi , lichidități pe piață.

Tabel 3.2 Exemplu 2 post-procesare

Neformatat	nu orice fel de emoție care le ți această misiune nu se cunosc foarte bine pilotul cu care a zburat și j nu a ajuns în această poziție întâmplător aeronava cu care a devenit instructor iar e nouăzeci și nouă este un avion subsonic de antrenament avansat construit în românia capabil de atac la sol și misiuni de recunoaștere ani continuăm cum proiect inedit în an centenar o sută de chipuri ale unirii de la o mie nouă sute optsprezece realizate de elevii liceului de arte din alba iulia
Formatat	Speaker S0, gender M: Nu orice fel de emoție care le ți această misiune nu se cunosc foarte bine , pilotul , cu care a zburat și J nu a ajuns în această poziție . Întâmplător. Speaker S1, gender F: Aeronava cu care a devenit instructor iar e 99 . Este un avion subsonic de antrenament avansat construit în România capabil de atac la sol și misiuni de recunoaștere ani. Speaker S2, gender F: Continuăm cum proiect inedit în an centenar 100 de chipuri ale Unirii de la 1918 , realizate de elevii liceului de Arte din Alba Iulia.

3.2 Generalități despre rețele neurale recurente

Rețelele neurale reprezintă un set de algoritmi, modelați în mare parte după creierul uman, proiectate pentru a recunoaște un anumit tipar. Aceste tipare reprezintă date din lumea reală precum imagini, date audio sau text ce sunt translatate în vectori de numere. Rețelele neurale profunde (Deep neural networks - DNN) diferă față de rețelele neurale simple cu un singur strat ascuns prin numărul de straturi prin care trec datele într-un proces de recunoaștere a tiparului din mai multe etape. O rețea “fully connected” se referă la faptul că fiecare neuron dintr-un strat este conectat la fiecare neuron din stratul următor (fiecare neuron de la ieșire este influențat de o combinație liniară a neuronilor de la intrare).

Rețelele neurale recurente (RNR) diferă față de rețelele standard feed-forward prin faptul că acestea au o noțiune a ordinii temporale, luând în considerare nu doar exemplul de la momentul curent ci și ieșirea rețelei pentru exemplul de la momentul anterior. Rețelele recurente pot avea dimensiunea vectorilor de intrare și de ieșire variabile. RNR conțin bucle ce permit ca informația să persiste [4]. În Figura 3.1 este prezentată în primă fază o parte dintr-o rețea neurală, **A**, ce primește la intrare x_t și scoate la ieșire o valoare h_t . Bucla permite ca informația să fie trecută dintr-o etapă a rețelei în alta. O rețea neurală recurentă poate fi văzută ca și copii multiple ale aceleiași rețele, fiecare trimițând un mesaj către un succesor. Tot în Figura 3.1 se poate observa și versiunea desfășurată a rețelei.

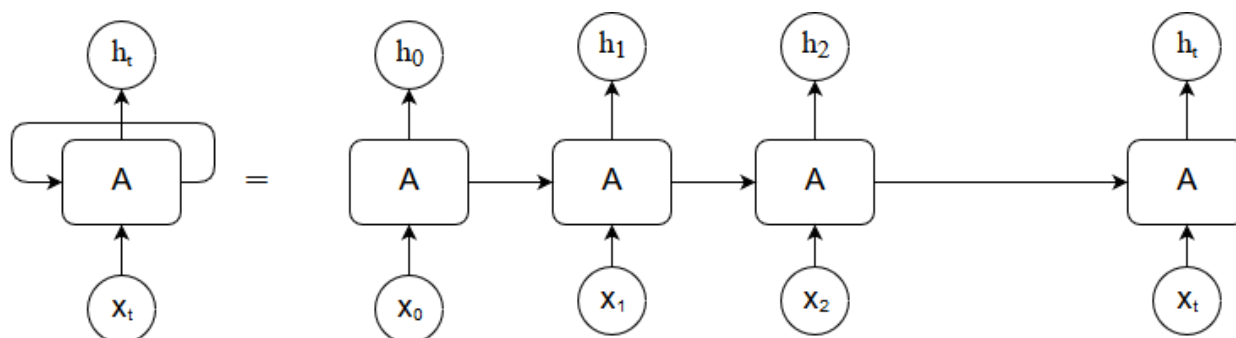


Figura 3.1 Rețea neurală recurentă desfășurată [4]

De asemenea, se mai poate adăuga un strat de rețea recurentă care să aibă în vedere informațiile legate de momentele ulterioare celui curent. Un astfel de model, ce conține informații atât din trecut cât și din viitor se numește rețea recurentă bidirecțională.

O problemă a RNR este că deși poate să învețe să folosească informații din trecut, există situații în care distanța între informațiile relevante și momentul curent, unde sunt necesare, să fie foarte mare, practic, se pierde informații pe măsură ce se avansează temporal. O soluție pentru această problemă o reprezintă rețelele LSTM (Long Short Term Memory). Aceste rețele sunt o variație a RNR capabile să învețe dependențe pe termen lung. O celulă LSTM poate decide să păstreze sau să șteargă o informație în funcție de importanța acesteia.

3.3 Metode de creștere a inteligibilității

În această secțiune sunt prezentate diverse metode de creștere a inteligibilității transcrierilor de vorbire.

În articolul “**On Formatting Transcriptions of Romanian Speech**” [5], se pune accentul pe :

- Separarea paragrafelor
- Formatarea de numere și date (conversia din litere în cifre)

- Restaurarea semnelor de punctuație
- Inserarea de litere mari

Utilizând un framework de post-procesare a transcrierilor, ce folosește la rândul său informația vorbitorului și momentele de timp asociate cu transcrierea neformatată, brută, se realizează organizarea transcrierilor în paragrafe, împreună cu inserarea de semne de punctuație și litere mari. Restaurarea semnelor de punctuație și inserarea de litere mari se efectuează folosind informația statistică lingvistică [6] și informația acustică. De asemenea, framework-ul de post-procesare formează numere și date (conversia numerelor din litere în cifre) creând astfel transcrieri mai inteligibile.

Transcrierea neformatată, împreună cu momentele de timp și informația vorbitorului sunt formate secvențial de către patru blocuri de procesare.

În primul rând, entitățile numerice sunt identificate și formate. Apoi transcrierea este segmentată în paragrafe. În final, sunt restaurate semnele de punctuație și literele mari sunt inserate printr-un proces în doi pași: o abordare (data-driven) bazată pe informația statistică lingvistică și o abordare (knowledge-based) bazată pe informația acustică.

Detecția propozițiilor și inserarea de puncte nu funcționează foarte bine: punctul este de obicei inserat când vorbitorul face pauza în vorbire iar acest lucru corespunde rareori cu sfârșitul propoziției, în special în vorbirea spontană. Totuși, chiar dacă nu sunt inserate la sfârșitul propozițiilor, pauzele în vorbire marcate cu punct cresc inteligibilitatea transcrierii.

Un alt articol, “**Joint Learning of Correlated Sequence Labeling Tasks Using Bidirectional Recurrent Neural Networks**” [7], propune o tehnică ce abordează mai multe sarcini corelate, precum inserarea semnelor de punctuație și a literelor mari folosind rețele neuronale recurente bidirecționale (BiRNN). În această lucrare, se tratează sarcina unei etichetări multiple de secvență, ce are ca scop atribuirea de etichete multiple fiecărui element din secvență, cum ar fi prezicerea semnelor de punctuație și inserarea de litere mari în transcrieri RAV. Etichetarea secvențelor implică atribuirea unei etichete fiecărui element din secvență. Ca și exemple, s-au inclus prezicerea semnelor de punctuație pentru transcrieri RAV, restaurarea literelor mari și recunoașterea numelelor proprii. De asemenea, se prezintă exemple de corelație a acestor sarcini, de exemplu, după inserarea unui punct urmează întotdeauna literă mare (it hasn’t been refined enough yet. It needs to be worked). Contribuțiile principale ale acestui articol sunt:

- Un framework BiRNN pentru etichetarea multiplă a secvențelor din sarcinile corelate.
- Îmbunătățirea prezicerii punctuației în transcrieri de vorbire prin antrenarea acesteia împreună cu sarcina de inserare de litere mari fără a folosi caracteristici prozodice.

Ca și set de date s-au folosit transcrierile emisiunii de televiziune “Intelligence Squared”, care sunt publice. S-au preluat 45 de dezbateri, fiecare conținând patru vorbitori din care s-a creat un set de antrenare-validare-test împărțit în 60:10:30. De asemenea s-au folosit transcrieri în engleză din transcrierile engleză-franceză din IWSLT 2012.

Fiecare secvență de antrenare constă într-un număr aleator de elemente/secvențe (între 40 și 70 în acest caz), cu constrângerea faptului că aceste secvențe trebuie să înceapă cu propoziție nouă. Propozițiile neîncheiate formează începutul următoarei secvențe de antrenare. Această schemă de generare de secvențe de antrenare previne modelul să insereze întotdeauna punct sau semn de întrebare la finalul fiecărei secvențe.

Modelul a fost antrenat folosind propagare inversă standard în TensorFlow. În experimentele din această lucrare, s-au antrenat două tipuri modele: model BiRNN, ce a fost antrenat la comun

pentru sarcini de inserare de semne de punctuație și inserare de litere mari (Corr-BiRNN) și modele pentru sarcini specifice (Single-BiRNN), ce au fost antrenate separat pentru fiecare sarcină în parte. În final, modelul BiRNN comun, antrenat numai pe caracteristici lexicale, a depășit ca performanță mai multe modele complexe, ceea ce demonstrează robustețea și abilitatea de generalizare.

“**Formatting Time-Aligned ASR Transcripts for Readability**” [8] folosește o metodă ce combină gramatici create de mână cu modele de limbă orientate pe clase antrenate pe text scris și care se bazează pe automate ponderate cu stări finite (Weighted Finite State Transducers - WFSTs) pentru a reține timpul de început și de sfârșit al fiecărui cuvânt.

În “**LSTM for Punctuation Restoration in Speech Transcripts**” [9], este prezentat un model bazat pe o rețea neuronală recurentă în două etape ce folosește LSTM pentru resturarea punctuației în transcrierile de vorbire. În prima etapă, caracteristicile textuale sunt învățate dintr-un corpus mare de text. A doua etapă combină caracteristicile textuale cu duratele pauzelor și adaptează modelul la domeniul de vorbire. Această abordare reduce numărul de erori în punctuație cu 16.9% față de un arbore de decizie ce combină un model de limbă (hidden-event language model posteriors) cu informația de pauză dintre cuvinte, cea mai bună îmbunătățire făcându-se în partea de restaurare de punct. Această lucrare descrie un sistem de restaurare de punctuație pentru vorbire difuzată în estoniană transcrisă automat ce folosește LSTM. LSTM a fost folosită pe o varietate de sarcini de etichetare de secvențe supervizate, printre care clasificarea fonemelor și înțelegerea limbii vorbite. Folosind arhitectura flexibilă a rețelelor neuronale, se antrenează un model de restaurare de punctuație în două etape. În primul rând, se folosește un corpus mare de text pentru antrenarea modelului, folosind numai cuvintele ca și caracteristici. Apoi un nou model este antrenat pe un corpus mai mic cu pauze adnotate, folosind durata pauzelor și ieșirea ultimului strat ascuns din primul model ca și caracteristici.

“**Restoring Punctuation and Capitalization in Transcribed Speech**” [10] prezintă o abordare ce realizează atât inserarea semnelor de punctuație, cât și inserarea de litere mari folosind un model de limbă n-gram bazat numai pe text. Se studiază impactul asupra performanței folosind diverse ordine de n-gram (de la $n=3$ până la $n=6$) și date de antrenare (de la 58 de milioane până la 55 de miliarde de elemente). Datele de antrenare au fost luate dintr-o colecție de articole de știri adunate de-a lungul anilor până în iunie 2007. Ca și date de evaluare s-au folosit două corpusuri: unul conținând articole de știri scrise, pentru a testa modelele în aceleași condiții ca și cele de antrenare și un corpus de transcrieri de știri difuzate, pentru a evalua abilitatea de generalizare a modelului pe vorbire transcrisă. Rezultatele arată că folosirea unui set mai mare de date de antrenare crește performanța, în timp ce creșterea ordinului n-gram nu ajută semnificativ.

3.4 Metoda propusă

Aplicația implementată are ca scop preluarea unui text ce conține numai litere mici și semne de punctuație și restaurarea literelor mari acolo unde este cazul. Arhitectura acestei aplicații este preluată de la aplicația de restaurare de diacritice descrisă în lucrarea de diplomă “Restaurare de diacritice utilizând tehnici bazate pe rețele neurale”, Iordache, 2019 [11].

Metoda folosită implică utilizarea unui corpus mare de text folosit pentru antrenarea unei RNR. Pentru fiecare propoziție din text se face o copie în care toate literele mari sunt înlocuite cu litere mici. Astfel, se obține un nou corpus de propoziții numai cu litere mici, care, împreună cu corpusul inițial va fi folosit pentru antrenarea RNR.

S-au realizat experimente pe 4 tipuri de modele:

- Model LSTM cu 1 strat unidirecțional

- Model LSTM cu 1 strat bidirecțional
- Model LSTM cu 2 straturi unidirecționale
- Model LSTM cu 2 straturi bidirecționale

Propozițiile sunt grupate în bucăți de 256 de caractere (propozițiile mai mari de 256 sunt împărțite în 2 sau mai multe părți în funcție de lungime, iar cele mai mici sunt grupate astfel încât împreună să formeze 256 de caractere) astfel încât să se obțină o matrice de lungime 256 (și lățime egală cu numărul propoziții obținute în urma grupării). În cazul în care nu s-a putut forma o linie de 256 caractere, s-a efectuat o completare cu 0 (aceasta etapă se efectuează după codificare). Această lungime fixă este folositoare pentru antrenarea pe loturi, fiind mai eficient.

După grupare, primele 5000 de exemple sunt folosite pentru validare. După fiecare epocă, aceste exemple sunt trecute prin rețea în scopul obținerii unei metrici: eroarea de validare.

Fiecare caracter va fi codificat printr-un vector de 128 de valori. Stratul LSTM primește vectorul de 128 de valori și scoate la ieșire un vector de 512 de valori. Se aplică un dropout de 0.12 pe ieșirea stratului LSTM. Stratul de fully-connected transformă cele 512 de valori într-un vector de 105 valori (egal cu dimensiunea vectorului de caractere de ieșire) nenormalizat, adică cu valori între minus infinit și plus infinit. Stratul de softmax normalizează aceste valori aducându-le în intervalul [0,1]; aceste valori însumate dau împreună valoarea 1. Hiperparametrii folosiți au fost de asemenea împrumutați de la aplicația de restaurare de diacritice.

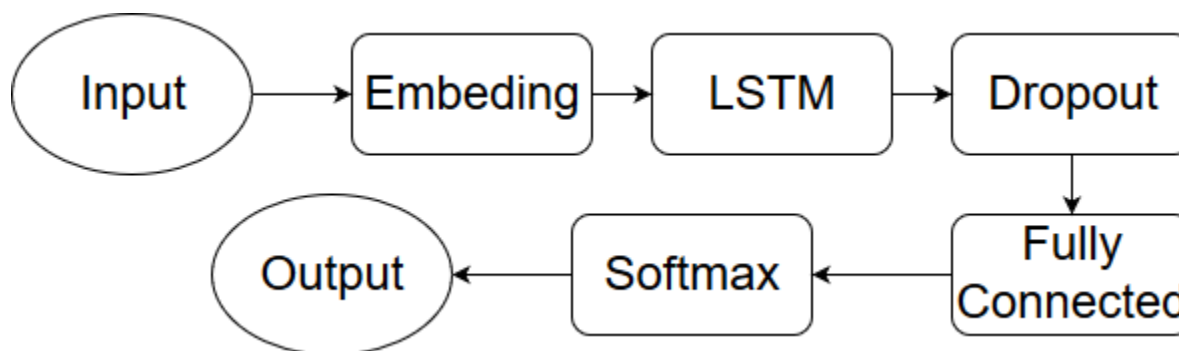


Figura 3.2 Arhitectura rețelei neuronale pentru restaurare de litere mari

La output avem probabilitățile pentru fiecare caracter din setul de propoziții de la intrare. Aceste probabilități vor indica în ce litere vor fi transformate caracterele. Caracterul cu probabilitatea maximă va înlocui caracterul de la intrare. Astfel, vom obține în final propozițiile restaurate.

Rețelele neuronale mari antrenate pe seturi de date relativ mici au tendința de a pierde din puterea de generalizare. Dropout este o tehnică de regularizare ce ajută la evitarea de supraînvățare (overfitting). În timpul antrenării, ieșirile unor straturi sunt ignorate. Astfel, un nou hiperparametru este introdus ce indică probabilitatea cu care ieșirile unui strat sunt ignorate sau reținute. Aceasta poate să fie implementată pe oricare strat ascuns, precum și pe stratul de intrare, dar nu și pe stratul de ieșire.

3.5 Rezultate experimentale

3.5.1 Setul de date utilizat și preprocesari

Setul de date utilizat este format din transcrieri preluate din baza de date CDEP. Aceste transcrieri au fost împărțite într-un set de antrenare și unul de evaluare. Setul de evaluare având 9975 de transcrieri iar cel de antrenare 219119 de transcrieri.

3.5.2 Rezultate

Ca și metrici de performanță s-au utilizat acuratețea, probabilitatea de restaurare eșuată și probabilitatea de alarmă falsă, fișierele text obținute după testarea celor 4 modele fiind comparate cu un fișier text de referință (groundtruth) . Acuratețea este calculată în felul următor: câte cuvinte din fișierul text restaurat sunt la fel cu cele din fișierul text de referință, împărțite apoi la numărul total de cuvinte. Probabilitatea de restaurare eșuată reprezintă câte cuvinte ce conțin literă mare în fișierul text de referință nu au fost restaurate corect de către model, împărțite apoi la numărul total de cuvinte ce conțin literă mare din fișierul text de referință. Probabilitatea de alarmă falsă este calculată într-un mod similar cu probabilitatea de restaurare eșuată, împărțind numărul de cuvinte ce au literă mare în fișierul text obținut la ieșirea modelului dar nu conțin literă mare în fișierul de referință, la numărul total de cuvinte ce conțin literă mare din în fișierul text obținut la ieșirea modelului.

Tabel 3.3 Acuratețe restaurare de litere mari

Acuratețe [%]	Număr de straturi	
Direcționalitate	1	2
unidirecțional	96.75	96.9
bidirecțional	98.91	99.1

Tabel 3.4 Probabilitate de restaurare eșuată pentru restaurare de litere mari

Probabilitate de restaurare eșuată [%]	Număr de straturi	
Direcționalitate	1	2
unidirecțional	17.15	16.28
bidirecțional	5.26	3.88

Tabel 3.5 Probabilitate de alarmă falsă pentru restaurare de litere mari

Probabilitate de alarmă falsă	Număr de straturi	
Direcționalitate	1	2
unidirecțional	6.08	5.84
bidirecțional	2.38	2.41

În concluzie, analizând datele din Tabel 3.3, Tabel 3.4 și Tabel 3.5 putem observa că obținem cele mai bune rezultate pentru modelul cu 2 straturi bidirecțional (cel mai complex).

4 Baze de date de vorbire pentru limba română

4.1 Motivație

În ultimii ani, algoritmi de procesare de semnal audio au devenit din ce în ce mai buni, mai ales în cazul recunoașterii automate a vorbirii. Însă, una din problemele cele mai mari pentru aceste sisteme rămâne lipsa unor baze de date de vorbire, în special în cazul limbii române. Având în vedere aceste lucruri, a fost necesară achiziția unei baze de date de vorbire, urmând a fi folosită în antrenarea modelelor pentru creșterea inteligibilității transcrierilor de vorbire, cât și pentru aplicații de IA viitoare.

4.2 Modalitate de achiziție

Baza de date CDEP (Camera Deputaților), a fost achiziționată folosind un set de aplicații dezvoltate în limbajul de programare Java cu ajutorul mediului de dezvoltare NetBeans, ce extrag și procesează date de pe site-ul camerei deputaților, www.cdep.ro. S-au extras de pe site înregistrările video și stenogramele ședințelor, între ianuarie 2003 și februarie 2019. De asemenea s-au extras și date înainte de 2003 (începând cu 1990), ce conțin stenograme dar nu și captări audio/video.

4.3 Aplicații software și procesări de date

Procesul de extragere a datelor a fost realizat folosind 4 aplicații:

1. Download Cdep Original
2. Wav Extractor
3. HTML Processor
4. Cdep Transcriptions Cleaner

Prima aplicație, Download Cdep Original, are rolul de a extrage fișierele video și paginile HTML de pe site, având la intrare un id de început și un id de final între care trebuie să descarce fișierele. Aplicația va parcurge HTML-uri de forma:

```
"http://www.cameradeputatilor.ro/pls/steno/steno2015.stenogram  
a?ids=[id]"
```

urmând apoi să caute dacă există sau nu o secvență HTML ce dovedește existența unui fișier video de forma:

```
"<a href="javascript:  
videoDivShow('http://www.cdep.ro/pls/steno/http_jwplayer?stream=p  
len/2005/20051128dep.mp4');"></a>"
```

În cazul în care nu există o astfel de secvență, se va descărca doar fișierul HTML. Dacă, în schimb, este găsită și secvența HTML, aplicația va continua cu extragerea link-ului către fișierul video și cu descărcarea fișierului video în sine.

Cea de-a 2-a aplicație, Wav Extractor, convertește fișierele video (mp4/f4v) descărcate de aplicația 1 în fișiere wav. Aceasta primește la intrare o dată de început și o dată de sfârșit între care procesează fișierele video. Numai fișierele video ce au data între datele de intrare vor fi procesate. De asemenea, ca date de intrare avem și attribute pentru fișierele audio obținute la ieșirea aplicației,

precum numărul de biți per secundă (bit rate = 256000), codec (codec = pcm_s16le) și frecvența de eșantionare (sample rate = 16000).

Aplicația a 3-a HTML Processor, procesează fișierele HTML descărcate de aplicația 1 cât și fișierele wav generate de aplicația 2. Procesarea fișierelor constă în următoarele operații:

- Extragerea transcrierilor fiecărui vorbitor în parte și stocarea lor în fișiere HTML
- Tăierea fișierelor wav generate de aplicația 2 în segmente de fișiere audio corespunzătoare transcrierilor vorbitorilor.
- Crearea/actualizarea unui fișier text ce conține lista vorbitorilor

Aplicația HTML Processor parcurge fiecare fișier HTML în parte și extrage numele vorbitorilor, URL-ul acestora (link către pagina lor de pe site-ul camerei deputaților), text-ul vorbit în format HTML și link-ul către porțiunea din fișierul video al ședinței corespunzătoare textului. Numele vorbitorilor urmează apoi să fie supus unui proces de normalizare ce presupune eliminare diverselor elemente adiționale găsite în nume, precum “(din sală)”, “(secretar de stat, Ministerul Sănătății)”, “Domnul”, “Doamna”; astfel sunt eliminate elementele din nume ce sunt în paranteză, se găsesc după caracterele “,”, “-”, “:” sau sunt prefixe la începutul numelui. După această etapă, numele vor fi comparate cu o listă de nume de normalizare. Această listă a fost necesară deoarece existau multe situații în care același vorbitor este întâlnit cu nume diferite, în unele situații fiind scris cu numele incomplet (Sorin Mihai Grindeanu vs Sorin Grindeanu), în timp ce în alte cazuri numele era diferit din cauza unor greșeli de scriere (Titus Nicolae Gheorghiof vs Titu Nicolae Gheorghiof). Această listă de normalizare a fost creată în urma unor comparații între numele vorbitorilor precum și a URL-urilor acestora. Comparația între nume a fost realizată folosind algoritmul DTW prezentat anterior în capitolul 2. Astfel perechile de nume cu cele mai mici scoruri (similaritate mare) au fost supuse unor analize manuale, iar perechile de nume ce indicau spre același vorbitor au fost adăugate în lista de excepții. Dacă un nume este găsit în lista de excepții, acesta va fi înlocuit cu numele corespunzător (Carol Emil Kovacs=Carol Emil Covacs), astfel păstrându-se un singur nume pentru vorbitor, evitând duplicate. După extragerea tuturor vorbitorilor, se efectuează o comparație atât între numele vorbitorilor cât și între URL-urile acestora, obținându-se o listă a vorbitorilor neduplicați. Aplicația continuă apoi să compare lista de vorbitori nou obținută cu lista vorbitorilor existenți (lista vorbitorilor din fișierele HTML procesate anterior) ce se află în fișierul cdepSpk.txt împreună cu un id unic pentru fiecare vorbitor, precum și URL-ul acestora. Vorbitorii sunt comparați atât din punct de vedere al numelui cât și din punct de vedere al URL-ului. Vorbitorii ce nu sunt găsiți în lista vorbitorilor existenți sunt supuși unei operații de comparație a numelor cu vorbitorii existenți în fișierul cdepSpk.txt folosind algoritmul DTW. În cazul în care se găsește o pereche de nume cu scor mai mic sau egal cu 2 (diferență de maxim 2 caractere) și această pereche nu se află într-o listă de excepții (perechi de nume procesate anterior manual) se va ridica un semnal de alarmă ce necesită o intervenție manuală. După analiza perechii de nume, aceasta va fi adăugată la lista de excepții, iar numele vor fi puse în lista de normalizare dacă acestea indica același vorbitor. În final, se efectuează etapa de scriere pe hard disk a transcrierilor HTML și a segmentelor audio corespunzătoare transcrierilor. Segmentele audio sunt obținute extrăgând din linkurile video timpii de început și de sfârșit. Exemplu link:

http://www.cdep.ro/pls/steno/htp_jwplayer?stream=plen/2019/20190220dep.mp4&offend=254&offset=66

În exemplul de mai sus, timpul de început este offset=66 iar timpul de sfârșit este offend=254 exprimat în secunde. Tăierea fișierului wav al întregii ședințe în segmente se face pe baza utilitarului sox. După aceste procese, vorbitorii noi vor fi adăugați la lista de vorbitori existenți din cdepSpk.txt.

Aplicația a 4-a Cdep Transcriptions Cleaner, preluând fișierele transcrierilor HTML obținute din aplicația 3, va scoate la ieșire pentru fiecare fișier HTML în parte un fișier txt ce conține transcrierea nemodificată în format text și un fișier text ce conține transcrierea procesată a transcrierilor. Transcrierea procesată este obținută folosind javaNLP2 [12], efectuând următoarele operații:

- Uniformizarea caracterelor, ce constă în înlocuirea caracterelor de forma: ã, ä -> ă; Ä -> Ă; â -> â; Ã, Â -> Â; Î -> Î; î -> î; Ş -> Ș; ş -> ș; Ţ -> Ț; ț -> ț;
- Înlocuirea numerelor cu forma lor scrisă: 1300 -> o mie trei sute;
- Înlocuirea caracterelor speciale: % -> la sută, < -> mai mic, = -> egal, + -> plus etc.
- Înlocuirea abrevierilor: art -> articolul, aprox -> aproximativ, nr -> numărul etc.
- Transformarea literelor mari în litere mici

Pentru a înțelege mai bine aceste procese, un exemplu concret este prezentat mai jos:

Fișier HTML (00010_20031211dep_part_006):

```
<td width="100%">
  <!-- START=557206,23848689,23878059 --> <p align="justify"><b><a
href="/pls/parlam/structura2015.mp?idm=86&cam=2&leg=2000"
target="PARLAMENTARI"><font color="#0000FF">Domnul Gheorghe
Dinu</font></a></b>: </p><p align="justify">La art.267, există aici un tabel care nu sună
deloc românește. Uitați-vă și dumneavoastră: "peste 150 de metri pătrați, dar nu peste 250
de metri pătrați". Ce înțelege un om care mai știe puțin și limba română?! Corect era: "între
250 de metri pătrați și 500 de metri pătrați, inclusiv, pentru categoria respectivă" și, atunci,
lucrurile erau clare. Domnul Bolcaș spunea că s-a tradus. De multe ori, s-a tradus chiar
prost.
  <!-- END --> </p></td>
```

Fișier txt nemodificat:

La art.267, există aici un tabel care nu sună deloc românește. Uitați-vă și dumneavoastră: "peste 150 de metri pătrați, dar nu peste 250 de metri pătrați". Ce înțelege un om care mai știe puțin și limba română?! Corect era: "între 250 de metri pătrați și 500 de metri pătrați, inclusiv, pentru categoria respectivă" și, atunci, lucrurile erau clare. Domnul Bolcaș spunea că s-a tradus. De multe ori, s-a tradus chiar prost.

Fișier txt procesat:

la articolul două sute șazeci și șapte există aici un tabel care nu sună deloc românește uitați-vă și dumneavoastră peste o sută cincizeci de metri pătrați dar nu peste două sute cincizeci de metri pătrați ce înțelege un om care mai știe puțin și limba română corect era între două sute cincizeci de metri pătrați și cinci sute de metri pătrați inclusiv pentru categoria respectivă și atunci lucrurile erau clare domnul bolcaș spunea că s-a tradus de multe ori s-a tradus chiar prost

4.4 Organizarea bazei de date

Baza de date este organizată în 2 foldere principale: “cdep” și “cdep-original”.

În “cdep-original” se găsesc folderele “video”, “html-w-video”, “html-wo-video” și “wav”. Folderele “video” și “html-w-video” sunt populate de aplicația Download Cdep Original ce descarcă atât paginile HTML cât și fișierele video (unde există). Aceste fișiere au de regulă următoare formă: [an][lună][zi].html/mp4/f4v. Exemple: 20180212.html, 20180212.mp4, 20180212dep.html, 20180212dep.mp4. Deoarece pot exista mai multe ședințe într-o zi, unele fișiere au terminația “dep”, pentru a semnaliza în cazul acesta că aceasta este o înregistrare/stenogramă a ședinței camerei deputaților și nu ședința comună a camerei deputaților și senatului. Există fișiere ce conțin și alte terminații în funcție de caz (20161222dep_a, 20161222dep_b, 20170207_partea1 etc.). Folderul “html-wo-video” este de asemenea populat de aplicația 1; acesta conține stenogramele ședințelor ce nu au și înregistrări video. În acest folder se vor regăsi fișiere HTML ce au ca denumire id-ul lor de pe site-ul camerei deputaților (de exemplu: 990.html). Fișierele din folderul “wav” sunt create de aplicația Wav Extractor și păstrează denumirea fișierelor din folderul “video” (de exemplu: 20180212.wav).

În folderul “cdep” avem folderele “wav”, “html”, “txt”, “trn” precum și fișierul cdepSpk.txt. Fișierul cdepSpk.txt este creat și actualizat de aplicația HTML Processor și conține lista vorbitorilor întâlniți în fișierle HTML, un id unic, precum și URL-ul către pagina lor de pe site-ul camerei deputaților. În folderele “wav” și “html” vor fi stocate fișierele wav tăiate, respectiv fișierele HTML ale transcrierilor sub următoare formă: [id-vorbitor_dată_parte]: 0001_20180208_001.wav/html, unde id-vorbitor este id-ul vorbitorului din cdepSPk.txt, dată este aceeași cu numele fișierului HTML de unde a fost extrasă transcrierea, iar parte reprezintă a câta oară vorbitorul respectiv a luat cuvântul la ședința respectivă. Aplicația HTML Processor crează foldere pentru fiecare vorbitor numite după id-ul vorbitorului din cdepSpk.txt; în aceste foldere vor fi stocate toate fișierle wav/html ale vorbitorilor. De exemplu, fișierele wav și html ce conțin a 2-a luare de cuvânt a vorbitorului cu id-ul 0003 la data de 08.02.2018 se vor găsi în “cdep/wav/0003/003_20180208_001.wav”, respectiv “cdep/html/0003/003_20180208_001.html” (numerotarea părților, cât și a id-urilor începe de la 0). În final, aplicația Cdep Transcription Cleaner va completa folderele “txt” și “trn” cu transcrierile neprocesate, respectiv procesate ale vorbitorilor sub aceeași formă ca și aplicația HTML Processor. Astfel, atât în folderul “txt” cât și în folderul “trn” se vor găsi foldere cu id-ul vorbitorilor ce conțin fișiere txt cu aceeași denumire ca în folderul “html”. Mai jos este prezentată o schemă a bazei de date:

- cdep-original
 - wav
 - [date].wav (i.e. 20180208.wav)
 - Video
 - [date].mp4/f4v (i.e. 20180208.wav)
 - html-w-video
 - [date].html (i.e. 20180208.html)
 - html-wo-video
 - [id].html (i.e. 81.html)
- cdep
 - wav
 - speakerID
 - [speakerID_date_part].wav (i.e. 0001_20180208_001.wav)

- html
 - speakerID
 - [speakerID_date_part].html (i.e. 0001_20180208_001.html)
- txt
 - speakerID
 - [speakerID_date_part].txt (i.e. 0001_20180208_001.txt)
- trn
 - speakerID
 - [speakerID_date_part].txt (i.e. 0001_20180208_001.txt)
- cdepSpk.txt

Din păcate, datele de pe site-ul camerei deputaților nu sunt întotdeauna consecvente, astfel, la achiziția bazei de date s-au întâlnit probleme precum:

- Același vorbitor sa fie scris cu nume diferite: Mircea Costache vs. Costache Mircea, Sorin Mihai Grindeanu vs. Sorin Grindeanu;
- Vorbitori care conțin în numele lor "din sală" sau "de la prezidiu"
- Existența vorbitorilor “din sală”, “voci din sală”
- Fișiere audio ce conțin voci din transcrieri anterioare (ultimele cuvinte ale vorbitorului precedent se aud în începutul fișierului audio al vorbitorului respectiv)
- Vorbitori ce nu au etichete de timp pentru porțiunea pe care o vorbesc
- Etichete de timp incomplete în link-ul video al vorbitorului (doar timpul de început)
- URL-uri atribuite greșit unor vorbitori
- Transcrieri ce conțin text în plus față de vorbire și invers

Este important de știut faptul că această bază de date nu este ideală și că în ciuda eforturilor de îmbunătățire a bazei de date (de exemplu: eliminarea de vorbitori duplicați sau a vorbitorilor ce conțin în numele lor în paranteză “din sală”), aceasta încă prezintă anumite probleme și poate prezenta și în viitor la achiziția de noi corpusuri de date.

4.5 Baze de date pentru limba română

În Tabel 4.1 sunt prezentate cele mai importante corpusuri de vorbire în limba română, împreună cu baza de date CDEP.

Tabel 4.1 Seturi de date de vorbire în limba română [13]

Nume	Tip vorbire	Domeniu	Nr. de rostiri	Nr. de ore	Nr. de vorbitori	Disponibilitate
RASC [14]	Citită	Articole Wikipedia	3k	4,8	N/A	public
RO-GRID [15]	Citită	General	4,8k	6,6	12	public
IIT [16]	Citită	Literatură	N/A	0,8	3	non-public
n/a [17]	Citită	Traduceri adaptate Eurom-1	4k	10,0	100	non-public

n/a [18]	Spontană	Internet, Emisiuni TV	N/A	4,0	12	non-public
RSS [19]	Citită	Știri, Literatură	4k	4,0	1	public
SWARA [20]	Citită	Ziare	19k	21,0	17	public
n/a [21]	Spontană	Știri	N/A	31,0	N/A	non-public
n/a [22]	Spontană	Bancar	N/A	40,0	30	non-public
RSC [3]	Citită	Știri, Interviuri, Literatură	147k	105	157	non-public
SSC [3]	Spontană	Radio și Emisiuni TV	227k	135	N/A	non-public
CDEP	Spontană	Politică	226.791	3510.2	2553	non-public

Corpusul SWARA [20] conține 21 de ore de vorbire de înaltă calitate de la 17 vorbitori, adnotate manual la nivel de rostire și semi-automat la nivel de fonem. Scopul principal al acestui corpus este să dea persoanelor cu dizabilități în vorbire posibilitatea de a folosi o voce sintetizată cât mai apropiată de a lor.

Un corpus de 40 de ore de conversații înregistrate de la 30 de vorbitori bazate pe 25 de scenarii de apeluri bancare este prezentat în [22].

În [3] sunt prezentate corpusurile RSC și SSC. RSC este un corpus de vorbire citită înregistrată într-un mediu ferit de zgomot conținând aproximativ 100 de ore, în timp ce SSC reprezintă un corpus de vorbire spontană conținând aproximativ 135 de ore din emisiuni radio și TV, unele înregistrări fiind afectate de zgomot de fundal.

Baza de date CDEP conține 226.791 de rostiri de la 2553 de vorbitori, formând un total de 3510.2 ore de vorbire. Aceasta conține vorbire din ședințele camerei deputaților și senatului român. Momentan, baza de date CDEP nu este publică, aceasta urmând să fie publicată în viitor.

5 Infrastructura web de transcriere de fișiere audio și căutare de cuvinte de interes

5.1 Arhitectura sistemului integrat

Serviciul „Speed Web Services” are 3 scopuri principale și anume:

1. Transcrierea de fișiere audio și post-procesarea transcrierilor respective;
2. Căutarea de cuvinte de interes în transcrierile fișierelor audio;
3. Restaurarea de diacritice în fișiere txt, doc și docx;

Pentru realizarea acestor sarcini, fișierul audio va trebui să treacă prin mai multe componente ce vor fi descrise în subcapitolele ce urmează. În figura de mai jos este prezentată arhitectura “Speed Web Services”.

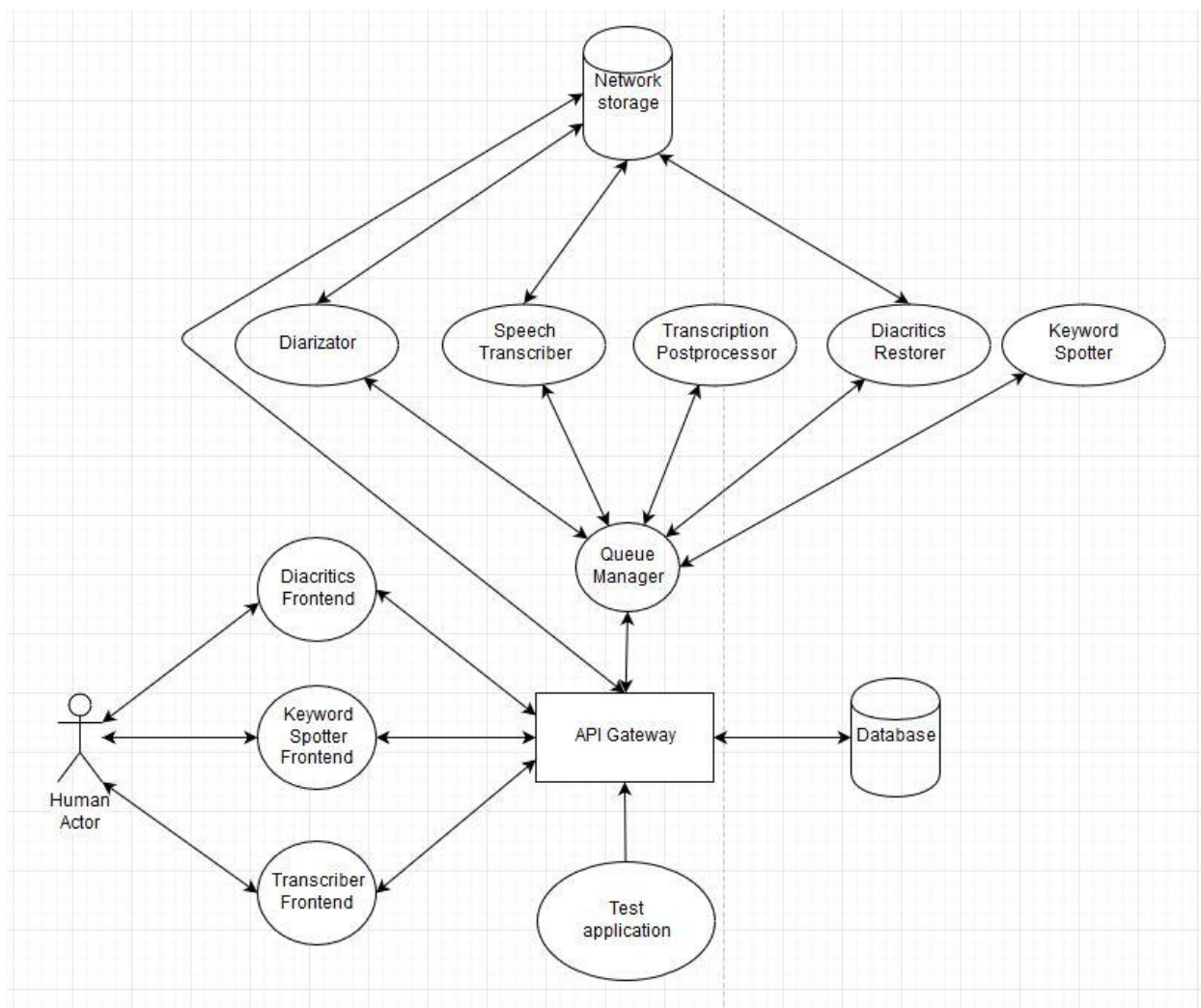


Figura 5.1 Arhitectura “Speed Web Services”

Acest serviciu este format din mai multe părți componente:

- Interfețe cu utilizatorul sub forma unor aplicații web;
- API Gateway;
- Componente de procesare de date;
- Baza de date.
- Manager de cozi
- Aplicație de test

5.2 Funcționalitate

Prin intermediul aplicațiilor web, utilizatorul poate accesa serviciul într-un browser web și poate să trimită cereri HTTP către API Gateway. API Gateway-ul este o componentă software reprezentând punctul unic de acces la serviciul web. Prin intermediul acestei componente cererile HTTP venite de la utilizator sunt procesate și transformate în cereri mai specifice către componentele de procesare de date. De exemplu, o cerere HTTP de transcriere pentru un fișier audio se transformă în cereri succesive de diarizare a fișierului audio (adresate componentei de diarizare), de transcriere a segmentelor audio (adresate componentei de transcriere) și de post-procesare a transcrierii (adresate componentei de post-procesare). API Gateway-ul și componentele de procesare de date comunică între ele prin intermediul unui manager de cozi. Fișierele audio și text ce sunt încărcate de utilizatori pentru procesare sunt stocate în Network Storage, ce reprezintă spațiul unde sunt ținute toate fișierele. În baza de date sunt ținute informații legate de utilizatori și fișierele audio ale acestora. Există de asemenea și o aplicație de test ce testează toate cererile HTTP pe care le poate primi API Gateway-ul în diverse situații și verifică dacă rezultatele obținute sunt cele așteptate. Această aplicație a fost creată cu scopul de a verifica buna funcționare a API Gateway-ului, precum și a componentelor de procesare de date și a bazei de date în cazul unor actualizări a serviciului sau în cazul unei instalări a serviciului de la zero.

5.3 Infrastructura serviciului web

Toate componentele ce formează împreună serviciul “Speed Web Services” sunt instalate pe mașini virtuale. Componentele de procesare de date sunt așezate împreună pe o mașină virtuală, în timp ce API Gateway-ul împreună cu interfețele vizuale și baza de date împreună cu managerul de cozi pe alte 2 mașini virtuale. Modul de organizare a fost subiectiv, aceste componente putând să funcționeze și de pe mașini virtuale separate. Network Storage-ul se află pe o partiție specială ce poate fi accesată atât de API Gateway cât și de componentele de procesare de date. API Gateway-ul împreună cu interfețele vizuale sunt ținute pe un server web de Apache Tomcat.

5.3.1 Baza de date și sistemul de partajare de date în rețea

Baza de date este creată folosind pgAdmin ce este un utilitar de gestionare a bazei de date pentru PostgreSQL. API Gateway-ul comunică cu baza de date prin intermediul API-ului JDBC, apelând procedurile stocate din baza de date. Aceasta bază de date este formată din 11 tabele:

1. users : conține informații legate de utilizatori
2. audio_files: conține informații legate de fișierele audio
3. text_files: conține informații legate de fișierele text
4. asr_domains: conține lista de domenii RAV
5. audio_jobs: conține lista de sarcini pentru fișiere audio trimise de utilizatori pentru procesare
6. text_jobs: conține lista de sarcini de restaurare de diacritice pentru fișiere text trimise de utilizatori pentru procesare
7. keywords: conține cuvintele de interes trimise de utilizator pentru o sarcină de ICI

8. keyword_positions: conține pozițiile în timp ale cuvintelor de interes identificate în fișierele audio
9. audio_segments: conține segmentele audio ale fișierelor audio
10. transcriptions: conține transcrierile segmentelor audio
11. pp_transcriptions: conține transcrierile post-procesate ale segmentelor audio

Câmpurile pentru fiecare tabel sunt prezentate mai jos:

- users
 - user_id: id-ul utilizatorului
 - first_name: prenumele utilizatorului
 - last_name: numele de familie al utilizatorului
 - user_email: adresa de email a utilizatorului
 - password_hash: parola utilizatorului criptată MD5
 - diarization_sec: numărul de secunde limită pe care utilizatorul le are pentru sarcini de diarizare. Valoare implicită: 7200 secunde.
 - speech-to-text_sec: numărul de secunde limită pe care utilizatorul le are pentru sarcini de transcriere. Valoare implicită: 7200 secunde.
 - keywords_sec: numărul de secunde limită pe care utilizatorul le are pentru sarcini de ICI. Valoare implicită: 7200 secunde.
 - keywords_count: numărul de cuvinte limită pe care utilizatorul le are pentru sarcini de ICI. Valoare implicită: 20 de cuvinte.
 - diacritics-rest_words: numărul de cuvinte limită pe care utilizatorul le are pentru sarcini de restaurare de diacritice. Valoare implicită: 10000 de cuvinte.
 - allow_login: câmp de tip boolean ce reprezintă abilitatea utilizatorului de a efectua operația de login sau nu
 - created_date: data de înregistrare a utilizatorului
 - is_admin: câmp de tip boolean ce reprezintă dacă utilizatorul are drepturi de administrator sau nu

De fiecare dată când utilizatorul cere o sarcină de transcriere/ICI/restaurare de diacritice, câmpurile corespunzătoare sarcinilor (diarization_sec, speech-to-text_sec pentru transcriere; diarization_sec, speech-to-text_sec, keywords_sec, keywords_count pentru ICI; diacritics-rest_words pentru restaurare de diacritice) sunt decrementate cu numărul de secunde ale fișierelor audio/ numărul de secunde ale fișierelor audio și numărul de cuvinte de interes căutate / numărul de cuvinte din fișierul text.

- audio_files
 - audio_id: id-ul fișierului audio
 - user_id: id-ul utilizatorului de care aparține fișierul audio
 - file_name: numele fișierului audio la care este adăugat eticheta de timp cu momentul încărcării acestuia
 - created_date: data de încărcare a fișierului audio
 - is_deleted: câmp de tip boolean ce reprezintă faptul dacă utilizatorul a șters sau nu fișierul audio
 - length: durata fișierului audio
 - sampling_rate: frecvența de eșantionare a fișierului audio
- text_files
 - text_file_id: id-ul fișierului text

- user_id: id-ul utilizatorului de care aparține fișierul text
- file_name: numele fișierului text la care este adăugat eticheta de timp cu momentul încărcării acestuia
- num_words: numărul de cuvinte din fișierul text
- diacritics_rest_file_name: numele noului fișier text după efectuarea procesului de restaurare de diacritice
- created_date: data de încărcare a fișierului text
- is_deleted: câmp de tip boolean ce reprezintă faptul dacă utilizatorul a șters sau nu fișierul text
- asr_domains
 - asr_domain_id: id-ul domeniului RAV
 - name: numele domeniului RAV. De exemplu: știri, sport
 - band: frecvența de eșantionare (8 sau 16 kHz)
 - created_date: data de inserare a domeniului RAV în baza de date
- audio_jobs
 - job_id: id-ul sarcinii audio
 - audio_id: id-ul fișierului audio ce trebuie procesat
 - job_type: tipul de sarcină (transcriere, transcriere cu post-procesare sau ICI)
 - asr_domain_id: id-ul domeniului RAV
 - status: starea în care se află sarcina (în progres sau încheiată)
 - created_date: data de creare a sarcinii audio
 - is_deleted: câmp de tip boolean ce reprezintă faptul dacă utilizatorul a șters sau nu fișierul audio pentru care s-a cerut sarcina audio
- text_jobs
 - text_job_id: id-ul sarcinii text (restaurare diacritice)
 - text_file_id: id fișierului text ce trebuie procesat
 - status: starea în care se află sarcina (în progres sau încheiată)
 - created_date: data de creare a sarcinii text
 - is_deleted: câmp de tip boolean ce reprezintă faptul dacă utilizatorul a șters sau nu fișierul text pentru care s-a cerut sarcina text
- keywords
 - keyword_id: id-ul cuvântului de interes
 - job_id: id-ul sarcinii ICI de care aparține cuvântul de interes
 - keyword: cuvântul de interes în sine
 - created_date: data de inserare a cuvântului de interes
- keywords_positions
 - keyword_positions_id: id-ul poziției cuvântului de interes
 - keyword_id: id-ul cuvântului de interes
 - start_time: timpul la care se începe pronunția cuvântului de interes în fișierul audio
 - length: durata de pronunție a cuvântului de interes în fișierul audio
 - created_date: data de creare a informațiilor legate de poziția cuvântului de interes
- audio_segments
 - segment_id: id-ul segmentului audio
 - audio_id: id-ul fișierului audio de care aparține segmentul audio
 - segment_type: tipul de segment (vorbitură, muzică, vorbitură cu muzică în fundal)
 - segment_band: frecvența de eșantionare a segmentului audio (8 sau 16 kHz)

- speaker_gender: sexul persoanei din segmentul audio
- spaker_name: numele persoanei din segmentul audio
- start_time: timpul de început al segmentului audio în fișierul audio
- length: durata segmentului audio
- file_name: numele fișierului segmentului audio
- created_date: data de creare a segmentului audio
- transcriptions
 - transcription_id: id-ul transcrierii
 - segment_id: id-ul segmentului audio de care aparține transcrierea
 - transcript_json: textul transcris din segmentul audio în format JSON
 - likelihood: scorul de încredere al transcrierii
 - created_date: data de creare a transcrierii segmentului audio
- pp_transcriptions
 - pp_transcription_id: id-ul transcrierii post-procesate
 - transcription_id: id-ul transcrierii asociate
 - pp_transcript_json: textul transcris și post-procesat în format JSON
 - created_date: data de creare a transcrierii post-procesate a segmentului audio

Relația dintre aceste tabele este prezentată în Figura 5.2:

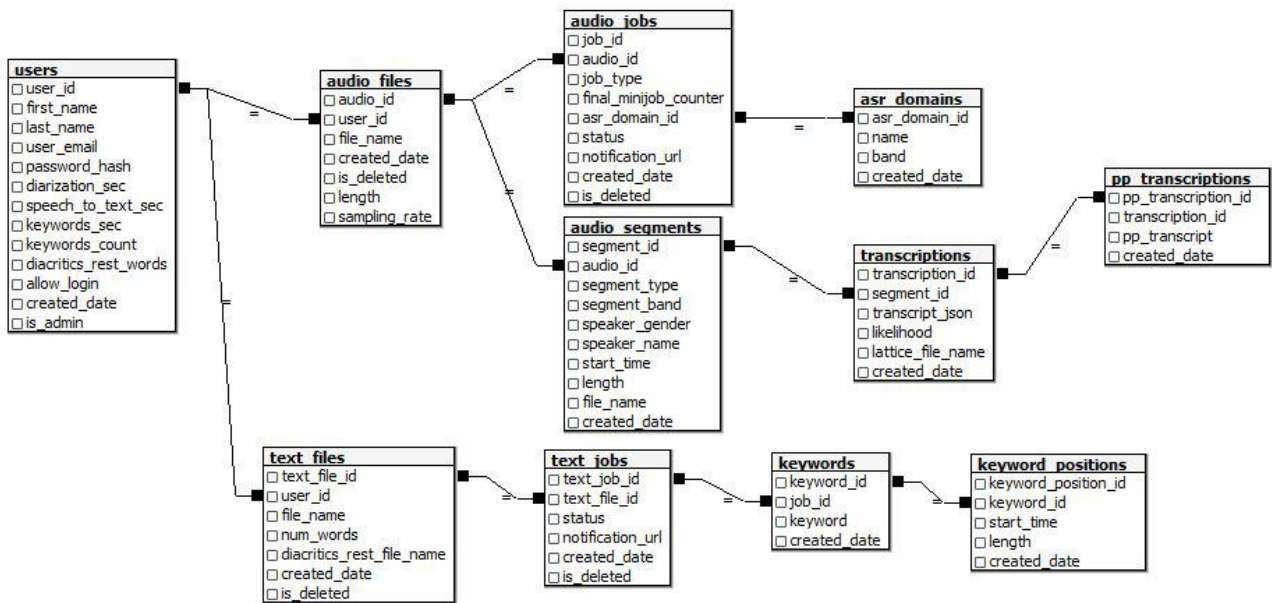


Figura 5.2 Arhitectura bazei de date “Speed Web Services”

Analizând Figura 5.2, se poate observa că există anumite câmpuri ce nu au fost menționate în listarea de mai sus, acestea fiind create în faza incipientă dezvoltării serviciilor “Speed Web Services” și nefiind folosite din motive de redundanță sau urmând a fi implementate pe viitor în versiuni ulterioare.

5.3.2 Cozi de mesaje și sarcini

Comunicarea între API Gateway și componentele de procesare de date se realizează prin intermediul managerului de cozi RabbitMQ. Aceasta este o componentă software ce gestionează

cozile pentru cele 5 componente de procesare de date, precum și pentru API Gateway. Aceste cozi reprezintă intermediul prin care API Gateway comunică cu componentele de procesare de date. În momentul primirii unei cereri HTTP de către API Gateway, aceasta va trimite un mesaj în coada componentei de procesare de date responsabilă pentru sarcina respectivă. După ce datele din mesaj au fost procesate de către componenta respectivă, aceasta va transmite un răspuns în coada API Gateway-ului. În general, fluxul de date pentru toate cazurile de utilizare este următorul:

- API Gateway-ul trimite o cerere de procesare în coada unei componente
- Componenta de procesare de date primește cererea, efectuează procesarea de date, apoi trimite răspunsul în coada API Gateway-ului
- API Gateway-ul primește răspunsul și acționează în funcție de acesta

Aceste cozi au de asemenea proprietatea de a fi “durabile”, în sensul că în cazul unei erori sau întreruperi în componenta de procesare de date, mesajul ce conține cererea de procesare de date nu va fi pierdut. Acesta va rămâne în coada și va fi retrimis la repornirea componentei. Mesajul va fi șters din coadă numai atunci când componenta a rezolvat sarcina respectivă și va trimite o confirmare către componenta de gestionare de cozi.

5.4 Componente de procesare de date

Componentele de procesare de date sunt:

1. Componenta de diarizare - segmentează fișierul audio în mai multe părți, în funcție de vorbitori și pauzele din vorbire;
2. Componenta de transcriere - realizează transcrierile segmentelor audio;
3. Componenta de post-procesare a transcrierii - îmbunătățește inteligibilitatea transcrierilor, prin formatarea numerelor, datelor, inserția semnelor de punctuație și a literelor mari;
4. Componenta ICI - identifica cuvinte de interes în transcrieri.

Pașii de urmat pentru transcrierea în text a unui fișier audio sunt descriși în Tabel 5.1.

Tabel 5.1 Pași de urmat pentru transcrierea în text a unui fișier audio

Pasul	Punct de acces HTTP	Descriere
Creare cont	POST users/register	Prin acest pas se creează un utilizator nou pentru serviciile de transcriere, restaurare de diacritice și identificare de cuvinte. Este necesară introducerea numelui și prenumelui, unei adrese de email valide și alegerea unei parole.
Login	POST users/login	În etapa de login, utilizatorul își va introduce adresa de email și parola pentru a se autentifica și a avea acces la serviciile de transcriere, restaurare de diacritice și identificare de cuvinte de interes.
Încărcare fișier audio	POST audio	Pentru a realiza o transcriere sau o identificare de cuvinte de interes, trebuie încărcat mai întâi fișierul audio pe server.
Cerere transcriere	POST transcriptions/{audio_id}	Utilizatorul va face o cerere de transcriere, selectând fișierul audio pe care dorește să-l transcrie și bifând sau nu opțiunea de post-procesare. În cazul unei cereri de transcriere fără post-procesare, în pagina de rezultate, utilizatorul va vizualiza transcrierea neformatată.
Interogare lista job-uri	GET transcriptions/jobs	După operația de cerere de transcriere, se poate observa progresul sarcinii în tabela de sarcini. În această tabelă sunt listate toate sarcinile de transcriere ale utilizatorului, conținând detalii precum fișierul audio pentru care s-a efectuat sarcina, dimensiunea fișierului audio, tipul de sarcină (transcriere cu sau fără post-procesare) și statusul sarcinii (în progres sau încheiată). În cazul unei sarcini încheiate, opțiunea de vizualizare de rezultate va fi disponibilă pentru sarcina respectivă.
Afișare rezultate job	GET transcriptions/{job_id}	Prin accesarea opțiunii de vizualizare de rezultate, utilizatorul va fi redirecționat către pagina de rezultate. Pe aceasta pagina sunt afișate forma de undă a fișierului audio (cu posibilitatea de a asculta fișierul respectiv) și transcrierea formatată sau neformatată în funcție de opțiunea de post-procesare aleasă la pasul de cerere de transcriere.

Pașii de urmat pentru identificarea unor cuvinte de interes într-un fișier audio sunt descriși în Tabel 5.2.

Tabel 5.2 Pași de urmat pentru identificare unor cuvinte de interes într-un fișier audio

Pasul	Punct de acces HTTP	Descriere
Creare cont	POST users/register	Prin acest pas se creează un utilizator nou pentru serviciile de transcriere, restaurare de diacritice și identificare de cuvinte. Este necesară introducerea numelui și prenumelui, unei adrese de email valide și alegerea unei parole.
Login	POST users/login	În etapa de login, utilizatorul își va introduce adresa de email și parola pentru a se autentifica și a avea acces la serviciile de transcriere, restaurare de diacritice și identificare de cuvinte de interes.
Încărcare fișier audio	POST audio	Pentru a realiza o transcriere sau o identificare de cuvinte de interes, trebuie încărcat mai întâi fișierul audio pe server.
Cerere identificare cuvinte cheie	POST keywords/{audio_id}/{asr_domain_id}	Utilizatorul va face o cerere de identificare de cuvinte de interes, selectând fișierul audio în care dorește să efectueze identificarea și introducând cuvintele de interes de căutat, separate prin virgulă.
Interogare lista job-uri	GET keywords/jobs	După operația de cerere de identificare de cuvinte de interes, se poate observa progresul sarcinii în tabela de sarcini. În această tabelă sunt listate toate sarcinile de identificare de cuvinte de interes ale utilizatorului, conținând detalii precum fișierul audio pentru care s-a efectuat sarcina, dimensiunea fișierului audio și statusul sarcinii (în progres sau încheiată). În cazul unei sarcini încheiate, opțiunea de vizualizare de rezultate va fi disponibilă pentru sarcina respectivă.
Afișare rezultate job	GET keywords/{job_id}	Prin accesarea opțiunii de vizualizare de rezultate, utilizatorul va fi redirecționat către pagina de rezultate. Pe aceasta pagina sunt afișate cuvintele de interes căutate și forma de undă a fișierului audio (cu posibilitatea de a asculta fișierul respectiv). Pe forma de undă a fișierului audio vor fi puse în evidență într-un chenar cuvintele de interes identificate. Aceste chenare vor avea o culoare diferită de la cuvânt la cuvânt și se vor regăsi pe forma de undă la momentul de timp la care au fost identificate cuvintele de interes; de asemenea, chenarul va conține cuvântul de interes identificat în colțul din stânga sus.

Pașii de urmat pentru restaurarea de diacritice într-un fișier text sunt descriși în Tabel 5.3.

Tabel 5.3 Pași de urmat pentru restaurarea de diacritice într-un fișier text

Pasul	Punct de acces HTTP	Descriere
Creare cont	POST users/register	Prin acest pas se creează un utilizator nou pentru serviciile de transcriere, restaurare de diacritice și identificare de cuvinte. Este necesară introducerea numelui și prenumelui, unei adrese de email valide și alegerea unei parole.
Login	POST users/login	În etapa de login, utilizatorul își va introduce adresa de email și parola pentru a se autentifica și a avea acces la serviciile de transcriere, restaurare de diacritice și identificare de cuvinte de interes.
Încărcare fișier text	POST text	Pentru a realiza o sarcină de restaurare de diacritice, trebuie mai întâi încărcat fișierul text pe server.
Cerere restaurare diacritice	POST diacritics/{text_id}	Utilizatorul va face o cerere de restaurare de diacritice, selectând fișierul text pentru care se dorește să se efectueze restaurarea de diacritice.
Interogare listă fișiere	GET text	După operația de cerere de restaurare de diacritice, se poate verifica dacă sarcina s-a încheiat sau nu în tab-ul "Home" unde sunt listate toate fișierele text ale utilizatorului. În cazul în care sarcina s-a încheiat, butonul de cerere de restaurare de diacritice o să fie înlocuit cu butonul de descărcare în dreptul fișierului text pentru care s-a solicitat restaurarea de diacritice.
Descărcare fișier restaurat	GET diacritics/{text_id}	Prin această opțiune, utilizatorul va putea să descarce fișierul text cu diacriticele restaurate.

5.5 Interfețe cu utilizatorul

Există 3 interfețe cu utilizatorul după cum se poate vedea și în arhitectura "Speed Web Services" prezentată la începutul capitoului în Figura 5.1. În cele ce urmează vor fi prezentate aceste 3 interfețe și opțiunile pe care le are utilizatorul.

Interfața cu utilizatorul pentru serviciul de transcriere de text conține următoarele pagini:

- Login: de unde se poate autentifica utilizatorul cu email și parolă
- Register: de unde un utilizator nou se poate înregistra completând următoarele câmpuri dintr-un formular: nume, prenume, email și parolă
- Transcriptions: pe această pagină, utilizatorul are mai multe opțiuni:
 - Înregistrarea unui fișier direct din interfață
 - Încarcarea unui fișier
 - Ascultarea unui fișier deja încărcat sau înregistrat
 - Solicitarea unei sarcini de transcriere, menționând fișierul audio pentru care se dorește transcrierea, domeniul RAV, precum și faptul dacă se dorește post-procesarea transcrierii

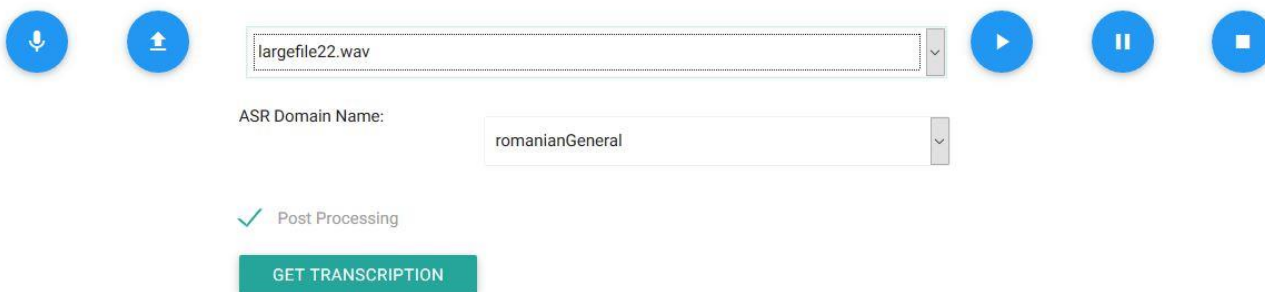
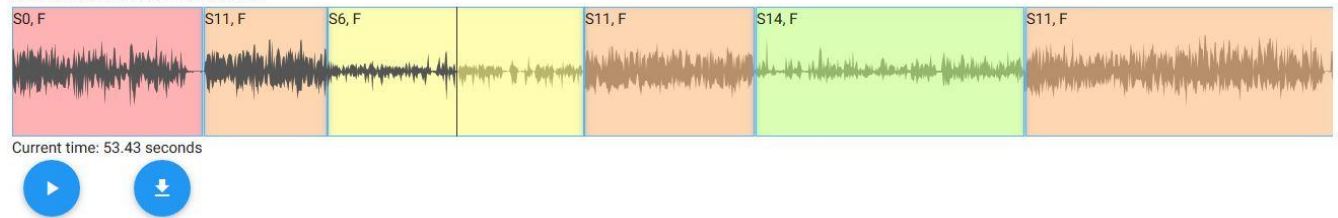


Figura 5.3 Interfață vizuală pentru serviciul de transcriere, pagina “Transcriptions”

- **Audio Files:** pe această pagină, utilizatorul poate să încarce fișiere audio noi și să le vizualizeze într-un tabel în care sunt listate numele fișierului, durata fișierului audio, data încărcării, precum și un set de acțiuni pentru fiecare fișier în parte: solicitare de transcriere (ce va redirecționa utilizatorul către pagina Transcriptions), ștergere și descărcare.
- **Jobs:** pe această pagină sunt listate sarcinile de transcriere solicitate într-o manieră similară cu cea a paginii Audio Files. Sunt listate numele fișierului audio pentru care a fost solicitată sarcina de transcriere, durata fișierului audio, data solicitării de transcriere, domeniul RAV, tipul de job (cu sau fără post-procesare), statusul sarcinii (în progres sau încheiată) și un set de acțiuni pentru fiecare sarcină: vizualizare rezultate (ce va redirecționa utilizatorul către pagina View Transcription) și descărcare transcriere (permite descărcarea transcrierii în format JSON într-un fișier txt). Acțiunile sunt disponibile doar pentru sarcini încheiate.
- **View Transcription:** pagina pe care se pot vizualiza rezultatele pentru o sarcină de transcriere finalizată. În partea superioară a paginii sunt notate numele fișierului audio pentru care s-a efectuat sarcina și domeniul RAV, urmat apoi de un player audio ce permite ascultarea fișierului audio, vizualizând de asemenea și forma de undă precum și zonele de vorbire pentru fiecare vorbitor. Opțiunea de descărcare a fișierului în format JSON se regăsește și pe această pagină lângă butonul de play. În partea de jos a paginii, poate fi vizualizată transcrierea fișierului audio împărțită pe vorbitori. Pe măsură ce fișierul audio se derulează, cuvântul vorbit la momentul respectiv este scos în evidență prin setarea fundalului acestuia cu galben. De asemenea se poate derula fișierul audio făcând click pe cuvântul din transcriere.

File Name: largefile22.wav

ASR Domain Name: romanianGeneral



S0, F: Autoritățile cer anularea în instanță a acordului de mediu pentru proiectul aurifer de la certe și ministrul mediului, Rovana Plumb, sunt experții au constatat că acordul nu respecta legislația și că cei responsabili de acest lucru, vor fi sancționați Rovana Plumb a subliniat că resursele naturale ale României, nu vor fi obiect de tranzacție pe sub mână. Pentru nici o funcție.

S11, F: O comisie de control formate din reprezentanții Agenției Naționale pentru Protecția Mediului Gărzii Naționale de mediu și Corpului de Control al ministrului, a constatat o serie de încălcări ale legislației române și europene privind acordul de mediu pentru proiectul de exploatare a aurului era certe și.

S6, F: Acordul de mediu, în cazul proiectului certe și eliberat de către Agenția Regională pentru Protecția Mediului Timișoara este ilegal și iresponsabil este ilegal pentru că încălca legislația românească și de asemenea, legislația europeană, în primul rând, se calcă o convenție internațională privind impactul de mediu transfrontalier.

S11, F: Eliberarea Acordului de mediu pentru proiectul minier de la certe și reprezintă prelungirea unui comportament iresponsabil al celor de la pdl care considerau la guvernare, că pot dispune cum doresc de resursele minerale și naturale ale României. A mai spus Rovana Plumb în cadrul consultărilor asupra impactului transfrontalier. Ungaria a fost cea care s-a opus Programului de la certe J.

S14, F: Privind în fi poziția și ungari de ce se spune așa. că în ciuda volumului său imens documentația ei. A, deci și valoarea de impact asupra mediului pregătită și prezentată de către investitor este inadecvată pentru evaluare obiectivă, pe baza evaluării experților se poate afirma faptul că documentația ei a nu include secțiuni care ar fi importante pentru evaluare reală.

S11, F: Ministrul a mai precizat că odată cu anularea acestui acord de mediu, va fi re inițiată procedura de autorizare și orice companie va avea dreptul să solicite autorizarea pentru un proiect de exploatare la certe și compania canadiană eldorado Gold corp Royce, n-a obținut în această vară avizul de mediu pentru exploatarea auriferă din comuna certe jude sus, mai multe organizații de mediu contest acest proiect, aveți încă Tehnologia de prelucrare în care se folosesc cianurile este riscantă pentru mediul înconjurător. Reprezentanții companiei au explicat însă că folosirea cianurilor se va limita la concentratul aurifer, care reprezintă 10% din totalul minereului exploatat în cariera de la certe și cu.

Figura 5.4 Interfață vizuală pentru serviciul de transcriere, pagina “View Transcription”

- **My Account:** în această pagină, utilizatorul poate să își schimbe adresa de email și parola și să vizualizeze câmpurile de secunde și cuvinte limită pe care le mai are.

Cristi@yahoo.com	Email
.....	Password
UPDATE	
Manolache	First Name
Cristian	Last Name
1 hour 53 minutes 15 seconds	Diarization seconds
1 hour 53 minutes 15 seconds	Speech to Text seconds
2 hours	Keywords seconds
1	Keywords Count
10000	Diacritics Restoration words

Figura 5.5 Interfață vizuală pentru serviciul de transcriere, pagina “My Account”

- Logout: opțiunea de ieșire din aplicație

Interfața cu utilizatorul pentru serviciul de ICI conține următoarele pagini:

- Login: de unde se poate autentifica utilizatorul cu email și parolă
- Register: de unde un utilizator nou se poate înregistra completând următoarele câmpuri dintr-un formular: nume, prenume, email și parolă
- Keyword Spotting: pe această pagină, utilizatorul are mai multe opțiuni:
 - Înregistrarea unui fișier direct din interfață
 - Încarcarea unui fișier
 - Ascultarea unui fișier deja încărcat sau înregistrat
 - Solicitarea unei sarcini ICI, menționând fișierul audio pentru care se dorește identificarea de cuvinte, domeniul RAV și lista de cuvinte de interes separate prin virgulă ce se dorește a fi identificate.



Figura 5.6 Interfață vizuală pentru serviciul de ICI, pagina “Keyword Spotting”

- **Audio Files:** pe această pagină, utilizatorul poate să încarce fișiere audio noi și să le vizualizeze într-un tabel în care sunt listate numele fișierului, durata fișierului audio, data încărcării, precum și un set de acțiuni pentru fiecare fișier în parte: solicitare ICI (ce va redirecționa utilizatorul către pagina Keyword Spotting), ștergere și descărcare.
- **Jobs:** pe această pagină sunt listate sarcinile ICI solicitate. Sunt listate numele fișierului audio pentru care a fost solicitată sarcina ICI, durata fișierului audio, data solicitării ICI, domeniul RAV, cuvintele de interes căutate, statusul sarcinii (în progres sau încheiată) și opțiunea de a vizualiza rezultatele. Opțiunea de vizualizare a rezultatelor este disponibilă doar pentru sarcini încheiate.
- **View KWS:** pe această pagină se pot vizualiza rezultatele pentru o sarcină ICI finalizată. În partea superioară a paginii sunt notate numele fișierului audio pentru care s-a efectuat sarcina, domeniul RAV și cuvintele de interes ce au fost căutate. Mai jos se găsește un player audio ce permite ascultarea fișierului audio, vizualizând de asemenea și forma de undă precum și zonele unde cuvintele de interes au fost identificate în fișierul audio. Făcând click pe regiunile cuvintelor de interes se pot asculta doar acele regiuni.

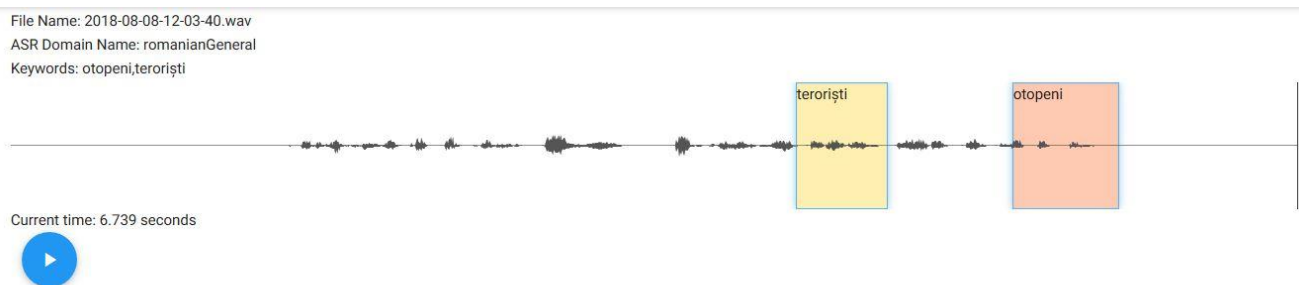


Figura 5.7 Interfață vizuală pentru serviciul de ICI, pagina “View KWS”

- **My Account:** în această pagină, utilizatorul poate să își schimbe adresa de email și parola și să vizualizeze câmpurile de secunde și cuvinte limită pe care le mai are.
- **Logout:** opțiunea de ieșire din aplicație

Interfața cu utilizatorul pentru serviciul de restaurare de diacritice conține următoarele pagini:

- **Login:** de unde se poate autentifica utilizatorul cu email și parolă

- Register: de unde un utilizator nou se poate înregistra completând următoarele câmpuri dintr-un formular: nume, prenume, email și parolă
- Home: pe această pagină, utilizatorul poate să încarce un fișier text nou, să solicite o sarcină de restaurare de diacritice și să descarce fișierul restaurat pentru o sarcină încheiată.



Figura 5.8 Interfață vizuală pentru serviciul de restaurare de diacritice, pagina “Home”

- My Account: în această pagină, utilizatorul poate să își schimbe adresa de email și parola și să vizualizeze câmpurile de nume și prenume.
- Logout: opțiunea de ieșire din aplicație

6 Concluzii

6.1 Concluzii generale

Serviciul “Speed Web Services” se pot găsi online pe site-ul laboratorului Speed. Componenta ICI este integrată deja în acest serviciu, iar componenta de post-procesare, aflându-se momentan în fază incipientă, urmează să fie integrată după dezvoltări ulterioare. Baza de date CDEP a fost achiziționată cu succes, efectuându-se deja experimente de antrenare și testare folosind transcrierile de vorbire.

Această lucrare prezintă pașii efectuați de autor în vederea realizării componentei ICI, a realizării parțiale a componentei de creștere a inteligibilității, precum și a achiziției bazei de date CDEP. După descrierea motivației și a obiectivelor în capitolul 1, capitolele 2, 3 și 4 prezintă în detaliu cele 2 componente de procesare de date realizate și programele folosite pentru achiziția bazei de date.

6.2 Contribuții personale

Acest proiect a fost dezvoltat în colaborare cu colegii mei de la laboratorul Speed și a reprezentat un efort de echipă. Contribuțiile personale ale autorului acestei lucrări sunt următoarele:

- Dezvoltarea, implementarea și testarea componentei ICI, precum și crearea bazei de date de cuvinte de interes
- Dezvoltarea parțială a componentei de creștere a inteligibilității transcrierilor, fiind dezvoltată după o aplicație de restaurare de diacritice ce nu face obiectul acestei lucrări. Momentan, această componentă se află în stadiul de restaurare de litere mari.
- Achiziția bazei de date CDEP
- Crearea serviciului de back-end (API Gateway) bazat pe arhitectura REST
- Crearea bazei de date pentru serviciile “Speed Web Services” și a procedurilor stocate
- Crearea aplicației de testare pentru API Gateway
- Preluarea unor versiuni incipiente, existente de interfețe vizuale pentru serviciile “Speed Web Services” și dezvoltarea lor ulterioară

6.3 Dezvoltări viitoare

Ca și dezvoltări viitoare, urmează dezvoltarea componentei de creștere a inteligibilității transcrierilor de vorbire în vederea obținerii unei aplicații ce realizează toate operațiile propuse în capitolul 3. Baza de date CDEP va fi actualizată prin alinierea transcrierilor fișierelor audio ce vor fi obținute folosind sistemul de transcriere din serviciile “Speed Web Services” cu transcrierile extrase din stenogramele ședințelor, pentru a obține secvențe audio-text cât mai precise pentru antrenarea viitoarelor aplicații de IA, rezolvând astfel problema transcrierilor ce conțin text adițional față de vorbirea din fișierul audio și a fișierelor audio ce conțin vorbire adițională față de transcriere.

7 Bibliografie

- [1] „<https://github.com/adobe/NLP-Cube>”.
- [2] „<https://www.rabbitmq.com/features.html>”.
- [3] „A. L. Georgescu, H. Cucu and C. Burileanu, Speed's DNN approach to Romanian speech recognition, In Speech Technology and Human-Computer Dialogue (SpeD), 2017 International Conference on (pp. 1-8). IEEE.”.
- [4] „<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>”.
- [5] „Cucu, H., Buzo, A., Caranica, A., & Burileanu, C. (2015). On Formatting Transcriptions of Romanian Speech. Rev. TJFE, ISSN, 0254-0770.”.
- [6] „Caranica, A., Cucu, H., Buzo, A., & Burileanu, C. (2015). Capitalization and Punctuation Restoration For Romanian Language. University “Politehnica” of Bucharest Scientific Bulletin.”.
- [7] „Pahuja, V., Laha, A., Mirkin, S., Raykar, V., Kotlerman, L., & Lev, G. (2017). Joint learning of correlated sequence labelling tasks using bidirectional recurrent neural networks. arXiv preprint arXiv:1703.04650.”.
- [8] „Shugrina, M. (2010, June). Formatting time-aligned ASR transcripts for readability. In Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics (pp. 198-206). Association for Co”.
- [9] „Tilk, O., & Alumăe, T. (2015). LSTM for punctuation restoration in speech transcripts. In Sixteenth annual conference of the international speech communication association.”.
- [10] „Gravano, A., Jansche, M., & Bacchiani, M. (2009, April). Restoring punctuation and capitalization in transcribed speech. In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4741-4744). IEEE.”.
- [11] „[Iordache, 2019] Florin Iordache, „Restaurare de diacritice utilizând tehnici bazate pe rețele neurale”, Lucrare de diploma, Universitatea Politehnica București, Iulie 2019.”.
- [12] „Horia Cucu, “Towards a speaker-independent, large-vocabulary continuous speech recognition system for Romanian”, PhD Thesis, University “Politehnica” of Bucharest, Oct 2011”.
- [13] „Georgescu, A. L., Caranica, A., Cucu, H., & Burileanu, C. (2018). RoDigits-a Romanian Connected-Digits Speech Corpus for Automatic Speech and Speaker Recognition. University Politehnica of Bucharest Scientific Bulletin”.
- [14] „Ș. D. Dumitrescu, T. Boroș and R. Ion, Crowd-Sourced, Automatic Speech-Corpora Collection–Building the Romanian Anonymous Speech Corpus. CCURL 2014:

Collaboration and Computing for Under-Resourced Languages in the Linked Open Data Era (2014): 90-94.”.

- [15] „A. Kabir and M. Giurgiu, A romanian corpus for speech perception and automatic speech recognition, The 10th International Conference on Signal Processing, Robotics and Automation. 2011”.
- [16] „A. D. Bibiri, D. Cristea, L. Pistol, L. A. Scutelnicu and A. Turculeț, Romanian Corpus For Speech-To-Text Alignment, In Proc. of the 9th International Conference on Linguistic Resources And Tools For Processing The Romanian Language (pp. 151-162).”.
- [17] „M. Boldea, C. Munteanu and A. Doroga, Design, Collection and Annotation of a Romanian Speech Database, In Proceedings of the First LREC-Workshop on Speech Database Development for Central and Eastern European Languages.”.
- [18] „V. Popescu, C. Petrea, D. Haneș, A. Buzo and C. Burileanu, Spontaneous Speech Database for Romanian Language, 2008.”.
- [19] „A. Stan, J. Yamagishi, S. King and M. Aylett, The Romanian speech synthesis (RSS) corpus: Building a high quality HMM-beased speech synthesis system using a high sampling rate, In Speech Communication 53.3 (2011): 442-450.”.
- [20] „A. Stan, F. Dinescu, C. Țiple, Ș. Meza, B. Orza, M. Chirilă and M. Giurgiu, The SWARA speech corpus: A large parallel Romanian read speech dataset, In Speech Technology and Human-Computer Dialogue (SpeD), 2017 International Conference on (pp. 1-6). IEEE.”.
- [21] „B. Tarjan, T. Mozsolics, A. Balog, D. Halmos, T. Fegyo and P. Mihajlik, Broadcast news transcription in Central-East European languages, In Cognitive Infocommunications (CogInfoCom), 2012 IEEE 3rd International Conference on (pp. 59-64). IEEE.”.
- [22] „G. Suci, Ș. A. Toma and R. Cheveresan, Towards a continuous speech corpus for banking domain automatic speech recognition, In Speech Technology and Human-Computer Dialogue (SpeD), 2017 International Conference on (pp. 1-6). IEEE.”.
- [23] „<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>”.
- [24] „<https://skymind.ai/wiki/lstm#recurrent>”.
- [25] „<https://skymind.ai/wiki/recurrent-network-rnn>”.
- [26] „<https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>”.
- [27] „Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780”.