Raport de analiză a impactului utilizării de RAV complementare pentru generarea de adnotări în contextul îmbunătăţirii sistemelor de RAV

Lucian Georgescu, Alexandru Caranica,

Horia Cucu, Dragoş Burileanu, Corneliu Burileanu

1. Introducere

După o primă discuție sumară despre seturile de date utilizate în cadrul acestui raport, detaliem în secțiunile următoare două subiecte principale. În primul rând prezentăm dezvoltarea suplimentară a metodei de generare automată de adnotări folosind sisteme RAV complementare. Astfel am încercat adaptarea şi aplicarea metodei proiectate în activitățile anterioare folosind un sistem de RAV complet nou, bazat pe platforma ESPnet, ca fiind unul dintre cele două sisteme RAV complementare. Secțiunea 2 prezintă paşii urmați pentru dezvoltarea acestui sistem.

În al doilea rând am efectuat un studiu comparativ asupra performanțelor sistemului de RAV inițial (la începutul proiectului ReTeRom) şi a sistemelor de RAV rezultate în cadrul activităților A1.13/2018 şi A2.13/2019, ca urmare a proiectării şi aplicării metodei de generare automată de adnotări folosind sisteme RAV complementare. Metoda se bazează pe folosirea a două sisteme RAV cât mai diferite, ce produc transcrieri pentru un corpus neadnotat, considerând ulterior părțile identice din transcriere ca fiind corecte. Această presupunere este certificată prin verificarea complementarității sistemelor; cele două sisteme diferă din punct de vedere constructiv astfel încât ele produc erori diferite şi necorelate. În final, setul de date adnotat este utilizat pentru reantrenarea celui mai performant sistem inițial. Secțiunea 3 prezintă acest studiu comparativ.

Acest raport de analiză face referire în multiple rânduri la raportul ştiințific şi tehnic aferent proiectului component TADARAV [Georgescu, 2020b] disponibil online pe pagina web a acestui proiect de cercetare[1].

[1] Proiectul de cercetare TADARAV: https://tadarav.speed.pub.ro

1.1 Seturi de date

Seturile de date la care se face referire în cadrul acestui raport sunt prezentate în Tabelul 1 şi Tabelul 2. Mai multe detalii despre aceste seturi de date puteți găsi în raportul etapei 2019 a proiectului TADARAV [Georgescu, 2020b], secțiunea 2.1. Este de menționat faptul că seturile de date de evaluare RSC-eval [Georgescu, 2020a] şi SSC-eval1 au fost actualizate în cursul anului 2020. Toate comparațiile rezultatelor de recunoaaştere automată a vorbirii (RAV) din acest raport cu rezultate publicate anterior anului 2020 trebuie să țină cont de îmbunătățirea artificială de performanță provenită din corectarea acestor seturi de date de evaluare. Mai multe detalii în [Georgescu, 2020b], secțiunea 2.1.

Tabelul 1. *Seturile de vorbire adnotată folosite pentru antrenarea şi evaluarea sistemelor de RAV şi seturile de vorbire adnotată obținute în etapa anterioară (2/2019)*

Setul de date	Subset	Durată
Spontaneous Speech Corpus (SSC)	SSC-train1+2	130h, 44m
	SSC-train3-trans-v4	41h, 00m
	SSC-train4-trans-v4	250h, 10m
	SSC-eval1	3h, 29m
	SSC-eval2	1h, 31m
Read Speech Corpus (RSC)	RSC-train	94h, 46m
	RSC-eval	5h, 29m
Contemporary Romanian Language (CoRoLa)	n/a	85h, 11m
Camera Deputaților (CDep)	CDep-eval	5h, 00m

Tabelul 2. *Seturi de date de vorbire neadnotată (+ transcrieri aproximative) utilizate ca date de intrare pentru cele trei metode de adnotare automată.*

Setul de date	Sursa	Durată	Transcrieri aproximative	Număr de vorbitori
CDep-raw	Parlamentul României	3,510h, 13m	25.1M cuvinte	~2,500
CoBiLiRo-raw	Alma[2]	19h, 29m	127.5k cuvinte	n/a
	VBarbu[3]	7h, 30m	80.8k cuvinte	
	GCVC[4]	16h, 49m	324.5k cuvinte	
	Ro100[5]	25h, 35m	181.8k cuvinte	

[2] Seria de emisiuni Alma Mater Iassiensis
[3] Interviu cu prof. Viorel Barbu
[4] Seria de emisiuni Ghici cine mai vine la cină
[5] Seria de emisiuni România 100: Iaşul în arcul timpului

2. Dezvoltarea sistemului de transcriere de vorbire ESPnet pentru limba română

Pentru dezvoltarea noului sistem de RAV pentru limba română, am optat să folosim ESPnet [Watanabe, 2018], un utilitar de învățare profundă modern ce permite realizarea unui sistem de RAV complet sub forma unei rețele neurale unice (arhitectură de tip end-to-end). Biblioteca de învățare profundă ESPnet poate fi folosită pentru o gamă întreagă de aplicații de inteligență artificială pornind de la recunoaşterea automată a vorbirii şi sinteză de vorbire pornind de la text şi mergând, ceva mai recent, şi spre traducerea automată vorbirii (en: speech translation) între multiple limbi de circulație internațională, dar şi conversia vorbirii (en: voice conversion).

Utilitarul include exemple de rețete (directoare de proiect) pentru limba engleză, de la care se poate porni dezvoltarea şi cercetarea în domeniu, cele mai multe fiind bazate pe resurse de antrenare audio şi text disponibile în licențe de tip sursă deschisă (*e.g.* LibriSpeech [Panayotov, 2015], CommonVoice [Ardila, 2020]), astfel încât orice cercetător să poată uşor replica rezultatele unor articole state-of-the-art.

ESPnet foloseşte, la rândul ei, bibliotecile Chainer [Tokui, 2019] şi PyTorch [Paszke, 2019] pentru implementarea funcționalităților legate de învățarea profundă (en: deep learning), lucru care simplifică extinderea codului şi arhitecturii de procesare. Ambele biblioteci sunt folosite de cercetătorii din domeniul învățării automate, pentru aplicații în procesarea imaginilor şi a limbajului natural.

De asemenea, un alt mare avantaj al ESPnet constă în procesarea datelor audio de intrare în stil Kaldi (cel mai popular utilitar de învățare profundă pentru RAV [Povey, 2011]) astfel încât extragerea parametrilor vocali şi formatul fişierelor caracteristicilor extrase se păstrează, iar portarea directoarele ce conțin resurse audio şi text între cele două sisteme fiind astfel extrem de facilă.

2.1 Arhitectura sistemului RAV propus

Sistemul RAV propus este bazat pe arhitectura end-to-end din ESPnet, folosind atât clasificarea temporală conexionistă (CTC) cât şi rețeaua codor-decodor bazată pe Transformer (en: self-attention encoder-decoder) [Kim, 2017; Watanabe, 2017]. Această a doua metodă utilizează un mecanism de auto-atenție, pentru a efectua alinierea între cadrele acustice şi simbolurile recunoscute, în timp ce CTC foloseşte ipotezele Markov pentru a rezolva eficient problemele secvențiale prin intermediul programării dinamice.

Astfel, am adoptat pentru RAV o rețea hibridă de tip CTC/Transformer end-to-end (Figura 1), care utilizează în mod eficient avantajele ambelor arhitecturi în antrenare şi decodare. În timpul antrenării, folosim framework-ul de învățare multi-obiectiv pentru a îmbunătăți robustețea alinierilor problematice cât şi pentru a obține o convergență mai rapidă.

În timpul decodării, efectuăm inferența atât prin combinarea scorurilor bazate pe atenție, cât şi a scorurilor CTC (Figura 2), într-un algoritm de căutare a fasciculului (en: beam-search) cu o singură trecere, pentru a elimina în continuare alinierile problematice. În plus față de arhitectura end-to-end prezentată mai sus realizată doar pe fişierele audio, am realizat o serie de experimente şi cu modele de limbă, de tip Transformer.

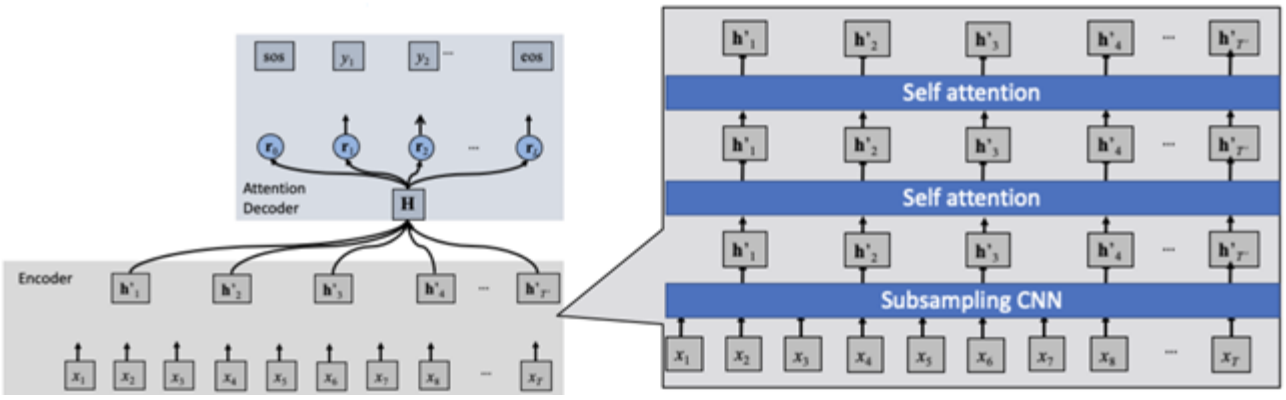

Figura 1. *Arhitectura rețelei CTC-Transformer utilizată, unde toate conexiunile recurente din codor-decodor-ul bazat pe atenție sunt înlocuite cu un bloc de auto-atenție (poate capta interdependențe pe distanțe foarte lungi). [Hori, 2019]*

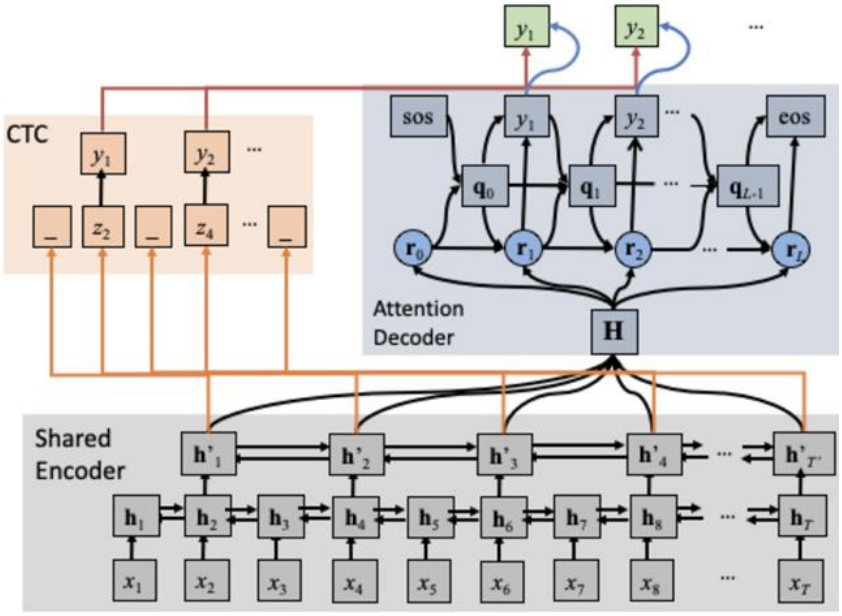

Figura 2. *Arhitectura rețelei CTC/attention, antrenarea fiind bazată pe învățarea multi-obiectiv iar inferența bazată pe combinarea scorurilor.*

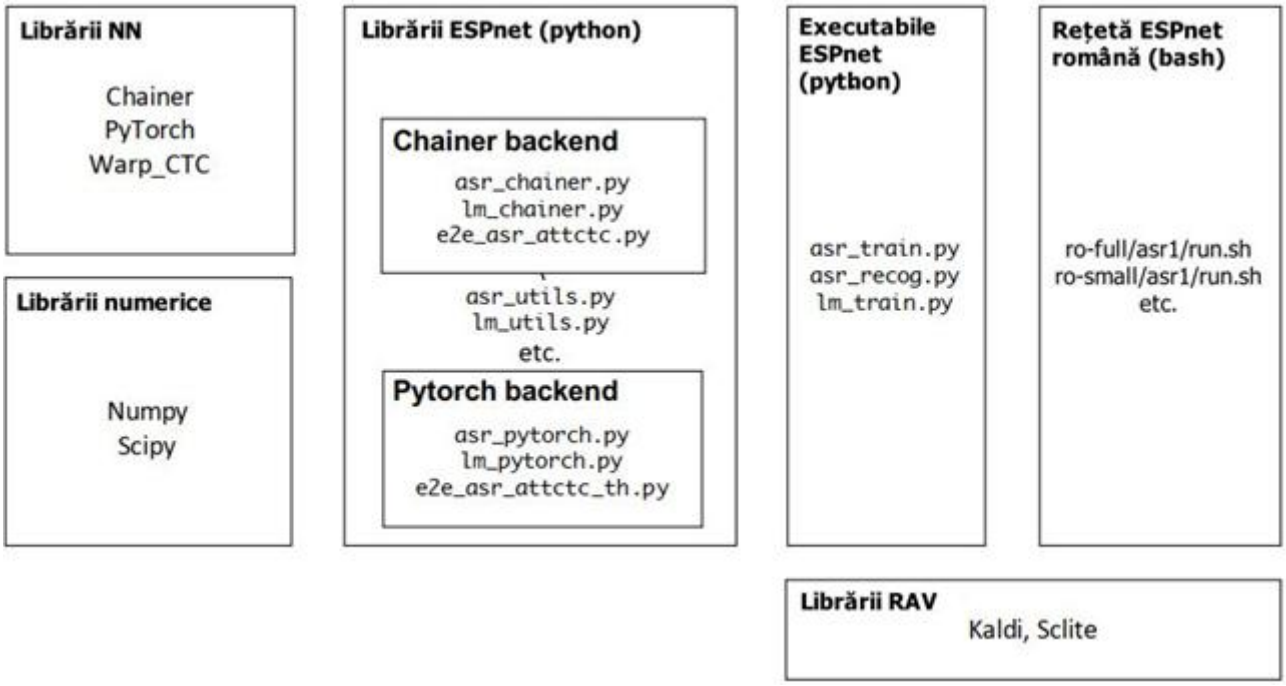

Figura 3. *Arhitectura software a sistemului RAV în limba română.*

2.2 Arhitectura software a sistemului ESPnet

Figura 3 prezintă arhitectura software a sistemului de recunoaştere în limba română, construit în ESPnet. Componentele principale pentru antrenarea şi inferența unei rețele neuronale sunt scrise în Python, care apelează SDK-urile Chainer şi PyTorch, prin comutarea backend-ului în funcție de opțiuni. Structura de directoare în care se preprocesează datele, după cum am menționat anterior, urmează filozofia Kaldi, utilizând scripturi bash pentru lansarea diferitelor binare necesare preprocesării datelor şi a augmentării lor.

În continuare, Figura 4 prezintă fluxul unei rețete (en: recipe) şi stagiile necesare executării cu succes a unei antrenări, apoi a unei decodări, pentru sistemul în limba română.

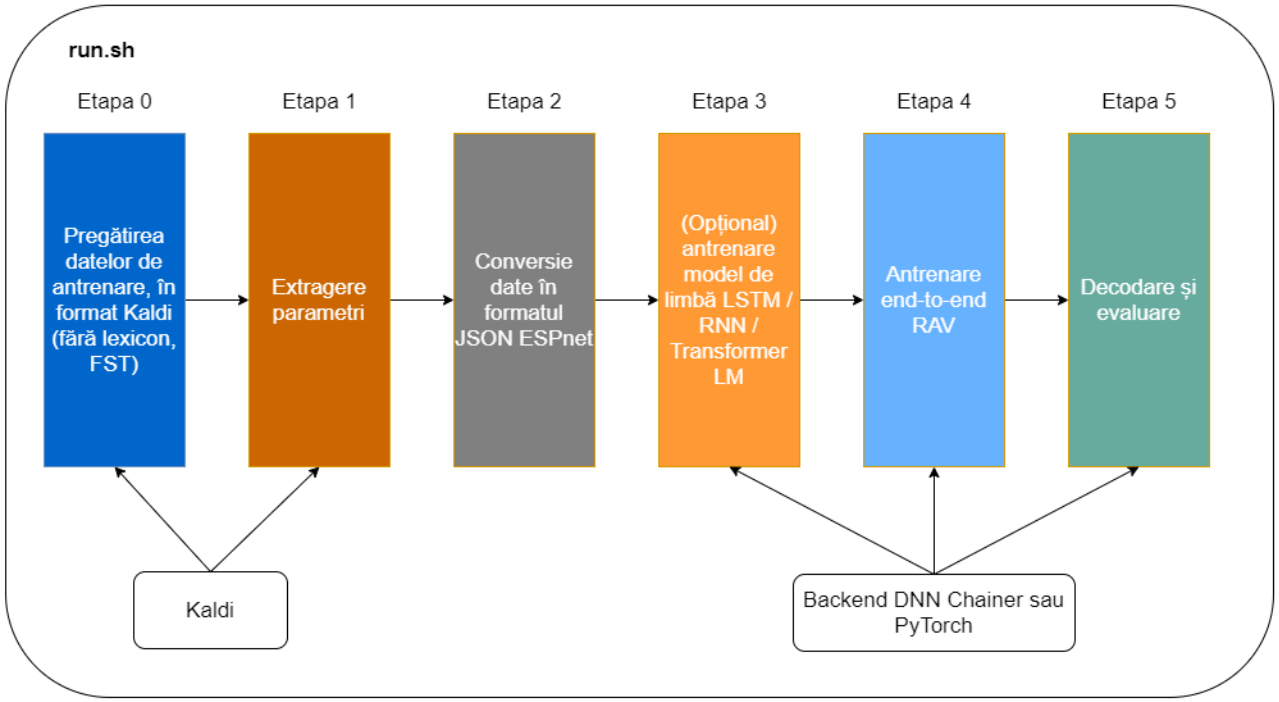

Figura 4. *Fluxul de execuție al rețetei ESPnet în Limba Română.*

Rețeta este semnificativ simplificată grație avantajului unui toolkit de RAV end-to-end, astfel, spre diferență de Kaldi, nu trebuie să includem un lexicon, compilarea transductorului de stare finită (FST), antrenarea / alinierea bazată pe modelarea mixturii HMM şi Gaussiene, precum şi generarea structurilor de tip latice pentru antrenarea secvențială discriminativă.

Etapele principale necesare fluxului unui sistem în limba română sunt:

- Etapa 0. Pregătirea datelor. Pentru această etapă a fost utilizat formatul Kaldi de stocare a datelor de antrenare (acelaşi format utilizat şi pentru celelalte sisteme de RAV ale SpeeD).
- Etapa 1. Extragere parametri. Din datele de intrare se extrag parametri de vorbire tipici, de tip MFCC, însă de înaltă rezoluție: 80 de coeficienți cepstrali; vectorul de parametri corespunzător fiecărei ferestre din semnalul de vorbire va avea 80 sau 83 de dimensiuni în funcție de cum la cei 80 de coeficienți cepstrali se adaugă sau nu şi informații despre frecvența fundamentală.
- Etapa 2. Conversie formate în ESPnet. Această etapă converteşte informația din structura de directoare Kaldi (transcrieri, id-uri vorbitor şi limbă, lungimea vectorilor de intrare) într-un singur JSON (data.json). Caracteristicile extrase din fişierele audio rămân în format Kaldi.
- Etapa 3. Antrenare model de limbă. Pas opțional, necesar dacă se doreşte utilizarea unui model de limbă suplimentar pentru reevaluare lingvistică. În experimentele noastre, am încercat ambele variante: sistem RAV end-to-end şi sistem RAV cu reevaluare lingvistică.
- Etapa 4. Antrenare end-to-end. În această etapă se antrenează model codor-decodor hibrid de tip CTC-Transformer, folosind una dintre cele două biblioteci de învățare profundă PyTorch sau Chainer.
- Etapa 5. Decodarea şi evaluarea. În această ultimă etapă se decodează setul de date de evaluare cu sistemul RAV nou antrenat format din modelul end-to-end (etapa 4) şi, eventual, modelul de limbă suplimentar (etapa 3). După decodare, se realizează alinierea transcrierii ipotetice cu cea de referință şi se calculează rata de eroare la nivel de cuvânt (WER).

2.3 Setup-ul experimental pentru experimentele de RAV

Experimentele realizate au urmărit dezvoltarea unui sistem de RAV bazat pe ESPnet în aceleaşi condiții în care a fost dezvoltat cel mai bun sistem de RAV SpeeD din anul 2019. Motivația acestei alegeri a fost bineînțeles dorința realizării unei comparații relevante între sistemul de RAV actual şi noul sistem dezvoltat cu ESPNet.

Astfel, au fost utilizate seturile de date listate în continuare şi descrise pe larg în secțiunea 2.1.1 a acestui raport:

- RSC-train şi SSC-train1+2 - set de date de antrenare restrâns utilizat în etapa de calibrare a hiperparametrilor;
- RSC-train, SSC-train1+2, SSC-train3+4-trans-v4 - set de date de antrenare utilizat pentru antrenarea sistemului final, după calibrarea hiperparametrilor;
- RSC-eval, SSC-eval1 şi SSC-eval2 - set de date de evaluare;
- news2014 şi talkshows - set de date de text utilizat pentru antrenarea modelului de limbă.

Pentru toate experimentele, setul de dezvoltare necesar în antrenarea rețelei neuronale ESPnet a fost selectat aleatoriu din setul de date de antrenare. Setul de dezvoltare reprezintă 10% din setul de date de antrenare şi a fost extras din acesta înainte de începerea procesului de antrenare.

2.4 Calibrarea hiperparametrilor sistemului de RAV

Utilitarul ESPNet include o serie rețete de antrenare pentru sisteme de RAV în limba engleză. Dintre acestea, cele mai populare sunt rețetele Librispeech, TED-LIUM v1, WSJ şi CommonVoice, rețete ce folosesc seturile de date cu acelaşi nume pentru antrenarea modelului de RAV. În ceea ce priveşte calibrarea hiperparametrilor sistemului, abordarea generală pe care am ales-o a fost următoarea:

- am analizat rețetele menționate mai sus şi am identificat hiperparametri ce diferă de la caz la caz;
- hiperparametri care nu diferă la diversele rețete pentru limba engleză au primit aceleaşi valori şi în experimentul pe limba română;
- pentru hiperparametri ale căror valori diferă de la o rețetă la alta, am ales pentru limba română o valoare adaptată dimensiunii setului de date de antrenare pe care îl avem la dispoziție, ținând cont, bineînțeles, şi de particularitățile limbii române.

Pentru exemplificarea procedurii de mai sus, luăm ca exemplu dimensiunea vocabularului de subcuvinte (*i.e.* Byte Pair Encoding [Sennrich, 2016]). ESPnet nu transcrie vorbirea direct în cuvinte, ci în unități lingvistice mai scurte numite informal subcuvinte. Acestea sunt pur şi simplu secvențe de litere ce apar în componența cuvintelor, însă fără să aibă vreo semnificație lingvistică anume (*e.g.* nu reprezintă prefixe, sufixe etc.). Subcuvintele se extrag din corpusul de date text de antrenare pe baza frecvenței lor de apariție. Dacă vocabularul de subcuvinte este limitat la dimensiunea N, atunci, folosind algoritmul prezentat în [Sennrich, 2016] se vor alege cele mai frecvente N subcuvinte cu care se pot genera toate cuvintele care apar în respectivul set de date.

În rețetele LibriSpeech, TED-LIUM v1, respectiv CommonVoice, dicționarul de BPE-uri este generat pe baza transcrierilor fişierelor de vorbire, transcrieri ce cuprind 9.4M cuvinte, 6.6M cuvinte, respectiv 4.5M cuvinte. Dimensiunile vocabularelor de subcuvinte variază de la 5000 (LibriSpeech) la 365 (CommonVoice). Astfel, raportul (număr subcuvinte)/(număr cuvinte în setul de antrenare) variază între 0.05% pentru Librispeech şi 0.0075% pentru TED-LIUM. Am decis ca pentru limba română să păstrăm acest raport între limitele menționate mai sus, însă am ținut cont şi de faptul că limba română este o limbă cu multe forme flexionate. Astfel, am ales să folosim un vocabular de subcuvinte de dimensiune mare: 1000 de subcuvinte, raportul (număr subcuvinte)/(număr cuvinte în setul de antrenare) situându-se la limita superioară: 0.05%.

În mod similar au fost analizate valorile hiperparametrilor pentru rețetele pentru limba engleză şi au fost alese valori concrete pentru următorii hiperparametri:

- pentru straturile de atenție:
  - adim (dimensiunea unităților neurale din straturile de atenție) a fost aleasă 256;
  - aheads (numărul de capete pentru straturile de atenție) a fost ales 4;
- pentru procedura de antrenare:
  - batch-size (numărul de rostiri per mini-batch) a fost limitat la 32, ca urmare a limitării memoriei RAM de doar 12 GB disponibile în procesoarele grafice Nvidia Tesla K40m, pe care s-a realizat antrenarea;
  - epochs (numărul de epoci de antrenare) a fost limitat la 60, deşi ar fi fost poate util să antrenăm modelul timp de 120 de epoci. Alegerea a fost influențată de timpul mare de antrenare pe hardware-ul disponibil;

    - transformer-lr (rata de învățare a modelului de tip transformer) a fost aleasă 5, ca un compromis între ratele folosite în rețetele pentru limba engleză
- pentru optimizarea obiectivului:
    - accum-grad (numărul gradienți însumați înainte de o ajustare a ponderilor rețelei) a fost ales 3 pentru că antrenarea s-a realizat în paralel pe 3 plăci grafice, astfel că ne-am permis să ajustăm ponderile rețelei după 3 treceri forward-backward realizate în paralel.
    - batch-bins (numărul de ferestre de vorbire ce intră în componența fiecărui mini-batch) a fost setat la 12M, ca urmare a limitării memoriei RAM de doar 12 GB disponibile în procesoarele grafice Nvidia Tesla K40m, pe care s-a realizat antrenarea;
- pentru modelul de limbă (de tip Transfomer):
    - att-unit (numărul de unități de atenție) a fost ales 256;
    - head (numărul de capete pentru straturile de atenție) a fost ales 2;
    - layer (numărul de straturi transformer) a fost ales 4;
- pentru procedura de antrenare a modelului de limbă:
    - batchsize (numărul de propoziții per min-batch) a fost ales 64;
    - epoch (numărul de epoci de antrenare) a fost limitat la 20, deşi ar fi fost poate util să antrenăm modelul timp de 50 de epoci. Alegerea a fost influențată de timpul mare de antrenare pe hardware-ul disponibil.

După antrenarea folosind parametri evidențiați anterior, au fost realizate un set de experimente pentru stabilirea valorilor optime pentru hiperparametri procesului de decodare:

- LMW (language model weight – ponderea scorului dat de modelul de limbă);
- CTCW (CTC weight – ponderea scorului dat de CTC).

2.5 Rezultatele experimentale

Rezultatele experimentale obținute sunt evidențiate în Tabelul 3. Putem observa că cele mai bune rezultate se obțin dacă ponderile date scorurilor provenind de la modelul de limbă suplimentar şi obiectivul CTC sunt egale (şi egale cu 0.5). De asemenea, putem observa că acest prim sistem RAV ESPnet se apropie de rezultatele obținute cu sistemul RAV dezvoltat cu Kaldi, însă nu este încă la fel de performant. Bineînțeles, trebuie luat în considerare şi faptul că atât modelul end-to-end de transcriere de vorbire, cât şi modelul lingvistic suplimentar au fost antrenate un număr de epoci relativ mic (mai puțin de 50% din cât ar fi trebuit) din cauza timpului lung de antrenare pe sistemele hardware avute la dispoziție. Într-un viitor apropiat, urmează să fie efectuate mai multe experimente cu sistemul ESPnet, iar aceste rezultate preliminare sunt îmbucurătoare şi ne permit să fim încrezători că vom putea egala performanțele sistemului dezvoltat cu Kaldi.

Din păcate, nu se poate afirma acelaşi lucru şi despre durata proceselor de antrenare şi decodare. Procesul de antrenare al sistemului RAV bazat pe ESPnet durează de aproximativ 5 ori mai multe decât un proces de antrenare similar pentru un sistem RAV Kaldi. În ceea ce priveşte procesul de decodare, experimentele noastre au arătat că sistemul ESPnet decodează o oră de vorbire în 2 până la 7 ore, în funcție de cât de similară este acustica respectivei ore de vorbire raportat la setul de date de antrenare. De partea cealaltă, decodarea unei ore de vorbire cu sistemul RAV bazat pe Kaldi durează doar 1 minut. Acest lucru ne-a permis să folosim sistemele bazate pe Kaldi în cadrul proiectului ReTeRom pentru a transcrise seturi de date mari (peste 500 de ore) de vorbire neadnotată şi apoi să proiectăm şi să aplicăm diverse metode pentru a obține un subset de date transcris corect (cu o acuratețe de peste 99%). Din cauza timpului extrem de lung de decodare pentru sistemul RAV bazat pe ESPnet acest sistem nu poate fi utilizat în mod similar în adnotarea automată a seturilor de date de vorbire.

Tabelul 3. *Comparație între sistemul RAV ESPnet şi sistemul RAV Kaldi. Optimizarea hiperparametrilor de decodare pentru sistemul RAV ESPnet în limba română.*

		Parametri decodare		WER [%]		
Set de antrenare de vorbire adnotată	**Model lingvistic**	**LMW**	**CTCW**	**RSC_eval**	**SSC_eval1**	**SSC_eval2**
RSC-train + SSC-train1+2	Nu	n/a	n/a	13.3	25.1	78.0
	Da	0.5	0.5	8.8	**21.3**	**38.9**
	Da	0.6	0.4	**8.7**	21.6	46.5
	Da	0.7	0.3	9.0	23.0	64.3
	Da	0.8	0.2	11.2	25.8	84.6
RSC-train + SSC-train1+2 + SSC-train3-trans-v4 + SSC-train4-trans-v4	Da	0.6	0.4	3.4	15.3	23.5
Sistem RAV baseline bazat pe Kaldi (antrenat pe acelaşi set de date ca mai sus, folosind model de limbă pentru reevaluare lingvistică)				1.8	11.0	14.0

3. Dezvoltarea metodei de adnotare bazate pe sisteme RAV complementare şi rezultatele obţinute pe parcursul proiectului TADARAV

În această secțiune prezentăm un studiu comparativ asupra performanțelor sistemului de RAV inițial (la începutul proiectului ReTeRom) şi a sistemelor de RAV rezultate în cadrul activităților A1.13/2018 şi A2.13/2019, ca urmare a proiectării şi aplicării metodei de generare automată de adnotări folosind sisteme RAV complementare.

Activitatea A1.13 din etapa 1/2018 a presupus utilizarea a două sisteme RAV inițiale pentru obținerea în mod automat de transcrieri cu grad ridicat de încredere, prin metoda sistemelor complementare. Sistemul inițial RAV #1 [Georgescu, 2017; Georgescu, 2018] a fost dezvoltat cu utilitarul CMU Sphinx [Huggins-Daines, 2006]. Modelele sistemului sunt probabilistice, de tip HMM-GMM (modelul acustic), respectiv 2-gram (modelul lingvistic). Acest sistem avea un vocabular de 64k cuvinte. Sistemul inițial RAV #2 [Georgescu, 2018] a fost dezvoltat cu utilitarul Kaldi [Povey, 2011], unde modelul acustic este unul de tip rețea neuronală cu întârziere în timp (time-delay - TDNN) specifică implementării NNET2. Modelul lingvistic este unul probabilistic, de tip 2-gram, folosit la decodare, în timp ce un model superior, de tip 4-gram, a fost folosit pentru reevaluarea lingvistică, corectând transcrierea inițială.

Cele două sisteme au fost utilizate pentru a transcrie seturile de vorbire neadnotată SSC-train3-raw, ce însumează 136 ore, şi SSC-train4-raw, cu o durată de 777 ore. În urma transcrierii şi aplicării metodei, s-au obținut seturile de date SSC-train3-compl-2018, cu o dimensiune de 49 ore, şi SSC-train4-compl-2018, cu o durată de 280 ore. Calitatea acestor noi seturi obținute a fost verificată prin aplicarea metodei asupra unor seturi de date pentru care există transcriere de referință: RSC-eval şi SSC-eval. S-a constatat că selecțiile rezultate sunt greşite într-o foarte mică proporție, numai 1.0% - 1.3%. Prin extrapolare, putem considera că şi noile date sunt corecte în mare măsură (aproximativ 99%), fiind astfel posibil ca acestea să fie folosite mai departe ca seturi de antrenare adiționale.

Tabelul 4 prezintă pe primele două linii caracteristicile şi performanțele sistemelor RAV inițiale, în timp ce a treia linie prezintă caracteristicile şi performanțele sistemului RAV #2 îmbunătățit, pentru a cărui antrenare au fost adăugate noile date obținute. Acest nou sistem a obținut o îmbunătățire relativă de 3.81% pe setul de evaluare cu vorbire citită, respectiv 8.87% pe setul de evaluare cu vorbire spontană.

Tabelul 4. *Performanța sistemelor RAV inițiale şi a sistemului RAV îmbunătățit din A1.13, etapa 1.*

Model acustic		Model linvistic	WER [%]		Îmbunătățire relativă a WER [%]	
Corpus antrenare	Tip model		RSC-eval	SSC-eval	RSC-eval	SSC-eval
RSC-train + SSC-train	HMM-GMM	Decodare RAV: 64k cuvinte, 3-gram	9.56	28.64	-	-
RSC-train + SSC-train	HMM-DNN (TDNN2)	Decodare RAV: 200k cuvinte, 2-gram Reev. lingv.: 200k cuvinte, 4-gram	3.41	17.91	-	-
RSC-train + SSC-train + SSC-train3-compl-2018 + SSC-train4-compl-2018	HMM-DNN (TDNN2)	Decodare RAV: 200k cuvinte, 2-gram Reev. lingv.: 200k cuvinte, 4-gram	3.28	16.32	3.81	8.87

Activitatea 2.13 din etapa 2/2019 a presupus aplicarea metodei sistemelor complementare, de data aceasta, cele două sisteme inițiale fiind sistemul RAV #2 prezentat în Activitatea 1.13 din 2018 şi sistemul RAV #3, având modelul acustic de tip time-delay (TDNN), creat cu utilitarul Kaldi folosind implementarea NNET3. Modelul de limbă folosit pentru decodare este un 2-gram, în timp ce modelul de limbă folosit pentru reevaluarea lingvistică este unul de tip rețea neuronală recurentă (RNN), cu un istoric de 5 cuvinte. Ambele modele de limbă folosesc un vocabular de 200k cuvinte.

Metoda a fost aplicată pe aceleaşi seturi de date de vorbire neadnotată: SSC-train3-raw şi SSC-train4-raw, cu o durată de 777 ore. În urma aplicării metodei, s-au obținut seturile de date SSC-train3-compl-2019 cu o dimensiune de 79 ore, şi SSC-train4-compl-2019, cu o durată de 452 ore. Calitatea acestor noi date a fost verificată prin aplicarea metodei asupra seturilor de referință RSC-eval şi SSC-eval. Selecțiile rezultate au fost greşite în proporție de 2.6% - 2.7%, mai mult decât în cazul sistemelor RAV #1 şi RAV #2. Acest fapt este pus pe seama diferențelor mai mici dintre cele două sisteme, ambele fiind dezvoltate cu ajutorul utilitarului Kaldi, iar modelele acustice sunt asemănătoare din punct de vedere al tehnologiei folosite.

Noile date obținute au fost adăugate la setul de antrenare deja existent, fiind creat sistemul RAV #3 îmbunătățit. Tabelul 5 prezintă pe primele 2 linii sistemele inițiale RAV #2 şi RAV #3. Următoarele două linii prezintă două versiuni ale sistemului RAV #3 îmbunătățit: (i) una antrenată după ce au fost adăugate datele SSC-train3-compl-2018 şi SSC-train4-compl-2018 la setul de date inițial, respectiv (ii) una antrenată folosind seturile SSC-train3-compl-2019 şi SSC-train4-compl-2019 suplimentar față de setul inițial. Urmărind rezultatele din acest tabel, putem trage două concluzii interesante:

- seturile de date obținute în 2018 sunt mai mici, însă probabil mai precise (sistemele inițiale din 2018 erau mai diferite decât sistemele din 2019) astfel că sistemul RAV reantrenat folosind seturile din 2018 este puțin mai bun decât sistemul RAV reantrenat folosind seturile generate in 2019.
- seturile de date nou obținute sunt mult mai utile pentru dezvoltarea unui sistem ce se doreşte a transcrie corect vorbire spontană: îmbunătățirea relativă a WER-ului este de peste 20% pentru vorbire spontană (SSC-eval) şi oarecum nesemnificativă pentru vorbire citită (RSC-eval).

Tabelul 5. *Performanța sistemelor RAV inițiale şi a sistemului RAV îmbunătățit din A2.13, etapa 2.*

Model acustic		Model linvistic	WER [%]		Îmbunătățire relativă a WER [%]	
Corpus antrenare	Tip model		RSC-eval	SSC-eval	RSC-eval	SSC-eval
RSC-train + SSC-train	HMM-DNN (TDNN2)	Decodare RAV: 200k cuvinte, 2-gram Reev. lingv.: 200k cuvinte, 4-gram	3.41	17.91	-	-
RSC-train + SSC-train	HMM-DNN (TDNN3)	Decodare RAV: 200k cuvinte, 2-gram Reev. lingv.: RNN 5-gram	1.88	14.96	-	-
RSC-train + SSC-train + SSC-train3-compl-2018 + SSC-train4-compl-2018	HMM-DNN (TDNN3)	Decodare RAV: 200k cuvinte, 2-gram Reev. lingv.: RNN 5-gram	1.80	11.70	4.26	21.79
RSC-train + SSC-train + SSC-train3-compl-2019 + SSC-train4-compl-2019	HMM-DNN (TDNN3)	Decodare RAV: 200k cuvinte, 2-gram Reev. lingv.: RNN 5-gram	1.87	11.78	0.53	21.26

Per ansamblu, integrând îmbunătățirile RAV obținute în 2018 si 2019 prin aplicarea metodei de adnotare automată propuse rata de eroare a scăzut pe vorbire citită de la 3.41% la 1.87% (o scădere relativă de 45%) şi de la 17.91% la 11.70% pe vorbire spontană (o scădere relativă de 35%).

Deşi activitățile din etapa 3/2020 nu prevedeau acest lucru, în cursul acestei etape s-a încercat dezvoltarea suplimentară a metodei utilizând ca sisteme inițiale sistemul RAV #3 (2019) şi un nou sistem de RAV dezvoltat folosind utilitarul ESPnet (vezi secțiunea 2). În acest demers am întâmpinat însă o problemă ce nu a putut fi încă depăşită. Sistemul RAV bazat pe ESPnet s-a dovedit a fi foarte lent în transcriere, astfel încât nu a putut fi utilizat pentru transcrierea seturilor de date neadnotate SSC-train3-raw şi SSC-train4-raw. Transcrierea acestor seturi de date, cu o durată totală de 913 ore, ar fi necesitat aproximativ 2700 de ore, adică peste 100 de zile. Pentru comparație, menționăm că transcrierea acelorași seturi de vorbire cu sistemul RAV bazat pe Kaldi a necesitat numai 16 ore. În urma acestei experiențe am tras concluzia că pentru aplicarea cu succes a metodelor proiectate în cadrul proiectului nu este suficient să existe sisteme de RAV inițiale performante şi complementare, ci, în plus, acestea trebuie să transcrie vorbirea suficient de rapid.

Bibliografie

[Ardila, 2020] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor We- ber, "Common Voice: A massively-multilingual speech corpus," in International Conference on Language Resources and Evaluation, 2020.

[Georgescu, 2017] A.-L. Georgescu, H. Cucu, C. Burileanu, "*SpeeD's DNN Approach to Romanian Speech Recognition,*" in Proc. 9th Conference on Speech Technology and Human-Computer Dialogue (SpeD), 8p, 2017.

[Georgescu, 2018] A.-L. Georgescu, H. Cucu, "Automatic annotation of speech corpora using complementary GMM and DNN acoustic models," In Proc. TSP 2018, pp. 1-4.

[Georgescu, 2020a] A.-L. Georgescu, H. Cucu, A. Buzo, C. Burileanu, "*RSC: A Romanian Read Speech Corpus for Automatic Speech Recognition,*" in the Proceedings of The 12th Language Resources and Evaluation Conference (LREC), pp. 6606-6612, 2020, Marseille, France.

[Georgescu, 2020b]A.-L. Georgescu, A. Caranica, C. Manolache, G. Pop, D. Oneață, H. Cucu, C. Burileanu, D. Burileanu, *„Proiect component TADARAV: Raport ştiințific şi tehnic în extenso 2020"*.

[Hori, 2019] Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Shinji Watanabe, *"Advanced Methods for Neural End-to-End Speech Processing – Unification, Integration, and Implementation,"* Interspeech 2019.

[Huggins-Daines, 2006] David Huggins-Daines, Mohit Kumar, Arthur Chan, Alan W. Black, Mosur Ravishankar, and Alexander I. Rudnicky. *"Pocketsphinx: A free, real-time continuous speech recognition system for hand-held devices."* in Proc. 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 1, pp. I-I. 2006.

[Kim, 2017] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint CTC-attention based end-to-end speech recognition using multi-task learning," in Proc. ICASSP, pp. 4835-4839, 2017.

[Panayotov, 2015] V. Panayotov, G. Chen, D. Povey and S. Khudanpur, *"Librispeech: An ASR corpus based on public domain audio books,"* in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206-5210, 2015.

[Paszke, 2019] Adam Paszke et al., *"PyTorch: An Imperative Style, High-Performance Deep Learning,"* Advances in Neural Information Processing Systems 32, pp. 8024-8035, 2019.

[Povey, 2011] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hanne- mann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely, "The Kaldi speech recognition toolkit," in Workshop on Automatic Speech Recognition and Understanding, 2011.

[Sennrich, 2016] Rico Sennrich, Barry Haddow, Alexandra Birch, "Neural Machine Translation of Rare Words with Subword Units," In Proc. 54th Annual Meeting of the Association for Computational Linguistics, pp 1715-1725, 2016.

[Tokui, 2019] Seiya Tokui, Ryosuke Okuta, Takuya Akiba, Yusuke Niitani, Toru Ogawa, Shunta Saito, Shuji Suzuki, Kota Uenishi, Brian Vogel, and Hiroyuki Yamazaki Vincent, *"Chainer: A deep learning*

framework for accelerating the research cycle," In Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 2002-2011, 2019.

[Watanabe, 2017] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R. Hershey and Tomoki Hayashi, *"Hybrid CTC/Attention Architecture for End-to-End Speech Recognition,"* IEEE Journal of Selected Topics in Signal Processing, vol. 11, no. 8, pp. 1240-1253, 2017.

[Watanabe, 2018] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "ESPnet: End-to-End Speech Processing Toolkit," in Proc. Interspeech, pp. 2207-2211, 2018.