Analiza erorilor sistemului RAV antrenat în proiectul 3 pe corpusul bimodal

- ICIA: Verginica Barbu Mititelu
- UPB: Horia Cucu, Lucian Georgescu, Cristian Manolache

Seturi de date de lucru

- 2 seturi de rezultate:
  - ale sistemului RAV baseline (cu antrenare pe RSC (100h) si SSC (125h), avea un WER de 2.79% pe RSC-eval si 16.63% pe SSC-eval) și
  - ale sistemului RAV îmbunătățit (antrenat pe RSC si SSC + SSC-train3+4-compl-2020 (550h), având WER de 2.52% pe RSC-eval si 13.22% pe SSC-eval și pe textele din CoBiLiRo)

Tipuri de erori

- Erori care afectează numele proprii
- Erori care afectează cuvinte scrise cu cratimă
- Erori care afectează cuvinte OOV

Liste de NEs

- Au fost create pentru a îmbunătăți resursele sistemului RAV
- Cuprind:
  - Nume de persoane
  - Nume de locuri și localități
  - Nume de firme
- Disponibile pe site-ul proiectului, la categoria Rezultate

Cuvintele cu cratimă

- Utilizarea resursei tbl pentru stabilirea modului corect de distribuire a cratimei
- Cazurile ambigue au fost tratate manual

Cuvinte OOV

- Cu ajutorul distanței Levenshtein, au fost propuse sugestii de corectare a cuvintelor inexistente in tbl

Concluzii

- Nici una dintre aceste abordări nu rezolvă complet problema evaluării rezultatelor sistemului de RAV
- listele cu echivalențe cuvânt de pornire – cuvânt corect pot fi folosite pentru îmbunătățirea treanscrierilor din CoBiLiRo
- există inconsecvență în folosirea diacriticelor: ambele variante ale literelor ș și ț se regăsesc în texte

TEPROLIN: Analiza erorilor sistemului TTS antrenate în proiectul 4 pe corpusul bimodal agregat în proiectul 1, adnotat și corectat (Activitatea 3.6)

Echipa de lucru:
Elena Irimia, Verginica Mititelu (ICIA);
Adriana Stan, Beata Lorincz (UTCN);

Obiectivele activității

Obiectiv principal

- evaluarea îmbunătățirilor pe care lexiconul construit de noi le aduce aplicațiilor de TTS dezvoltate la UTCN

Obiectiv secundar

- analiza a diferite scenarii de evaluare care folosesc informația din lexicon integral sau parțial

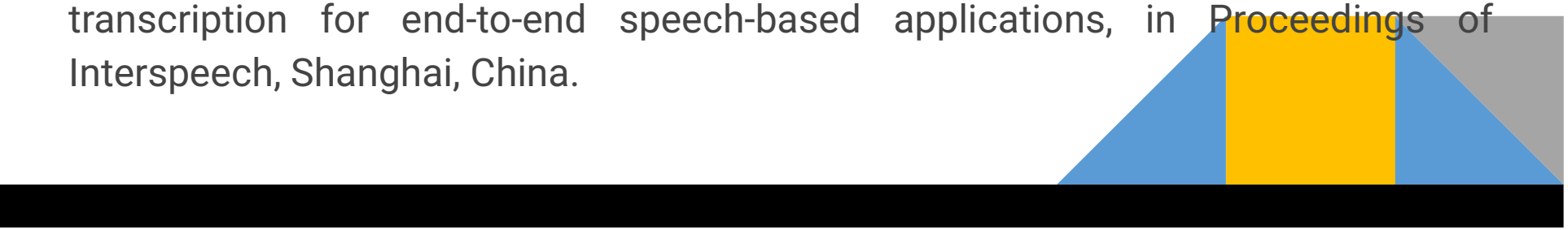
Contextul evaluării lexiconului RoLEX, dezvoltat în TEPROLIN

- testarea utilității lexiconului în cadrul unei aplicații de sinteză a vorbirii (TTS) dezvoltată de partenerul UTCN;
- pentru a simplifca procesul de evaluare, am restrâns contextul evaluat la nivel textual (calitatea vorbirii sintetizate este mai greu de evaluat în mod obiectiv, cuantificabil);
- rețeaua neuronală folosită pentru evaluarea lexiconului are ca scop predicția (în format text) concurentă a silabificării, accentului și transcrierii fonetice pornind doar de la forma (ortografică a) cuvântului, în lipsa unui context de utilizare.

Arhitectura rețelei neuronale Transformer utilizate

- Structură encoder-decoder cu straturi de atenție multi-head (4 centre de atenție), normalizare și feed-forward
- Selecția hiperparametrilor (Stan, 2020)
- Dimensiunea vectorului embedding = 128, cu ponderi inițializate aleatoriu înainte de antrenare;
- Dimensiunea batch=512; optimizatorul Adam; rata inițială de învățare: 0,0002

Stan, Adriana (2020) RECOApy: Data recording, pre-processing and phonetic transcription for end-to-end speech-based applications, in Proceedings of Interspeech, Shanghai, China.



Întrebări

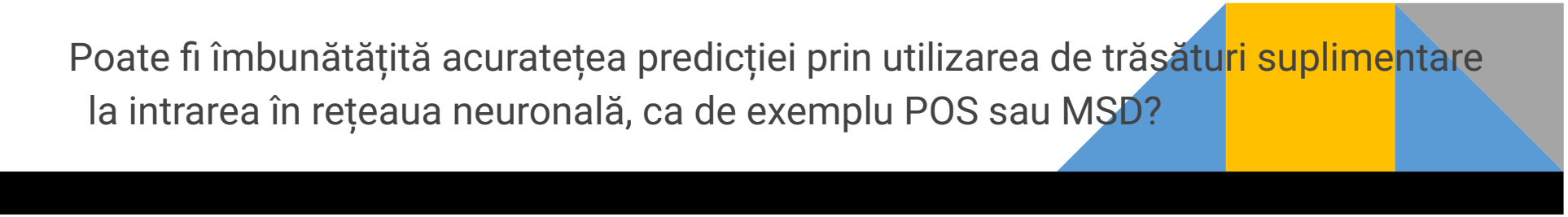
Performanța instrumentelor: antrenare pe RoLEX vs. antrenare pe seturile de date disponibile anterior?

Cum este afectată performanța predicției de dimensiunea setului de antrenare?

Este posibil să selectăm metodic subset-uri de mici dimensiuni care să conducă la rezultate mai bune decât subset-urile selectate aleator?

Care este contribuția fiecăruia dintre cele trei task-uri de predicție lexicală la rata de eroare globală?

Poate fi îmbunătățită acuratețea predicției prin utilizarea de trăsături suplimentare la intrarea în rețeaua neuronală, ca de exemplu POS sau MSD?



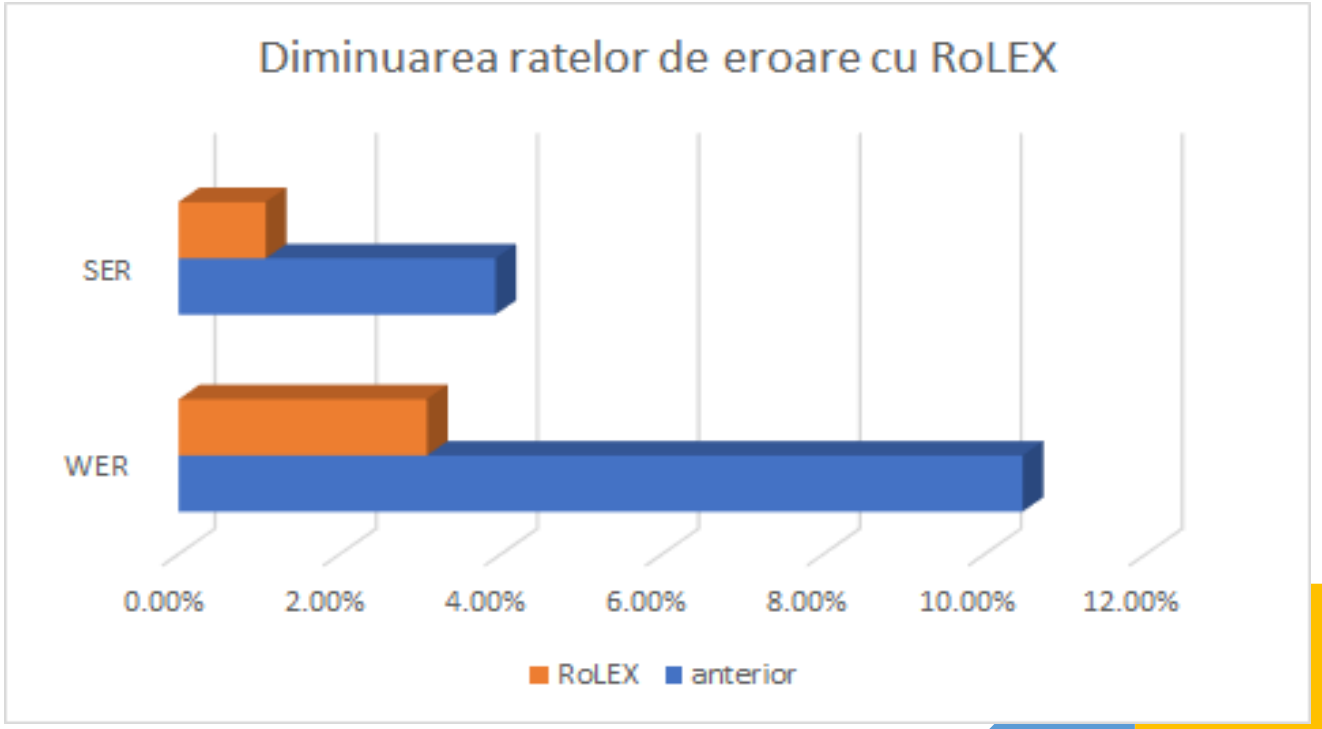
Condiții de evaluare

- 80% date de antrenare vs. 20% date de testare

WER (word error rate): rata de eroare la nivel de cuvânt = procentul de secvențe de ieșire prezise incorect;

SER (symbol error rate): rata de eroare la nivel de simbol (similară cu rata de eroare la nivel de fonem, doar că include și simbolurile care marchează silabificarea și accentul) = distanța Levenshtein între secvența prezisă și secvența țintă

Ratele de eroare: RoLEX vs resursele folosite anterior



Scenarii de evaluare

Selecția metodică a subseturilor

- Subsetul LEMMA: intrări din RoLEX corespunzătoare tuturor lemelor cuvintelor conținut și tuturor formelor cuvintelor funcționale (30.150);
- Subsetul 1-FORM: intrări corespunzătoare unei forme pt. fiecare lemă din lexicon (aleasă astfel încât să ne asigurăm de diversitatea morfologică a subset-ului) și tuturor formelor cuvintelor funcționale (30.150);
- Subsetul 2-FORM: intrări corespunzătoare a două forme pt. fiecare lemă din lexicon (aleasă astfel încât să ne asigurăm de diversitatea morfologică a subset-ului) și tuturor formelor cuvintelor funcționale (55.185);

În funcție de trăsăturile de intrare în rețea

- Ortho: forma cuvântului
- wPOS: forma cuvântului + partea de vorbire
- wMSD: forma cuvântului + eticheta morfosintactică MSD (https://github.com/clarinsi/mte-msd/blob/master/tables/msd-human-ro.tbl)

Ratele de eroare WER și SER ale predicției concurente pentru subseturi aleatoare de date, de dimensiuni diferite, în scenariile de evaluare Ortho, wPOS și wMSD

Subseturi	Ortho		wPOS		wMSD	
	WER	SER	WER	SER	WER	SER
5.000	24,83%	9,96%	22,68%	8,99%	21,78%	8,78%
50.000	10,74%	4,03%	7,79%	3,00%	6,39%	2,46%
100.000	7,84%	2,71%	5,64%	2,10%	4,36%	1,65%
150.000	6,24%	2,25%	4,94%	1,81%	3,59%	1,33%
RoLEX	5,60%	1,97%	4,34%	1,59%	**3,08%**	**1,08%**

WER și SER ale predicției concurente pentru subset-urile LEMMA, 1-FORM și 2-FORMS în scenariile Ortho, wPOS și wMSD

	LEMMA		1-FORM		2-FORMS	
	WER	**SER**	**WER**	**SER**	**WER**	**SER**
Ortho	40.13%	15.82%	9.99%	3.70%	**7.43%**	**2.76%**
wPOS	41.56%	16.52%	8.01%	3.04%	**5.24%**	**2.13%**
wMSD	36.24%	13.88%	6.25%	2.39%	**4.11%**	**1.59%**

Platforma integrată și configurabilă de prelucrare a textelor în limba română

ICIA: Radu Ion

radu@racai.ro

Disponibilă public

- https://github.com/racai-ai/TEPROLIN
- Are nevoie de Python 3.6, Java Runtime Engine 15 și Perl
- `pip3 install -r requirements.txt`
- Există un `README.md` cuprinzător pentru instalarea modulului
- Se poate testa la https://relate.racai.ro/index.php?path=teprolin/complete

Îmbunătățiri și adaptări

- Am înlocuit componentele proprietare TTS de la UTCN cu librăria MLPLA open source a lui Tiberiu Boroș, scrisă în Java
  - Despărțire în silabe
  - Detectarea accentului
  - Transcriere fonetică
- Expandarea numeralelor în forma literală
  - Am portat codul în Java dezvoltat pentru managerul de dialog din ROBIN
- Inserarea de diacritice se face automat în funcție de numărul de cuvinte care sunt ilegale în limba română fără diacritice (de ex. masina, cat, sipca, etc.)

Îmbunătățiri (continuare)

- Dependența operațiilor unele de altele se face acum într-un graf orientat
  - De exemplu, dacă dorim NER, nu mai executăm analiza cu relații de dependență sintactică
- Resursele necesare (modele, dicționare, etc.) se instalează acum automat
- Am adăugat **UD-Pipe 1** cu modelul pe limba română care e acum aplicația default pentru sentence splitting, tokenizare, POS tagging și lematizare

TO DO

- Un pachet wheel pentru Python 3 care să poată fi instalat cu `pip3 install teprolin`
- Testarea cu Python 3.9 și Perl 5.24 (ultimele versiuni)
- Altceva?