# ROBIN Dialog
# Raport științific și tehnic în extenso pe anul 2020

### Rezumatul etapei

În anul 2020 activitatea de cercetare a fost dublată de eforturile de dezvoltarea a unei noi soluții de ASR cu performanțe mai bune decât a variantei anterioare. Managerul de dialog a fost îmbunătățit și el substanțial prin extensia capacității de înțelegere a unor enunțuri eliptice precum și adaptarea la noua soluție de ASR. Au fost dezvoltate noi micro-lumi conform unor scenarii suplimentare. Noile dezvoltări sunt disponibile în github-ul public, https://github.com/racai-ai/ROBINDialog, clonabil și în platforma ERRIS. În plus, platforma publica RELATE a fost îmbogățită cu noul modul ASR și un modul de înregistrare a vorbirii, astfel încât utilizatorii să poată experimenta direct în browser noile facilități.
Deși programată pentru anul 2021, în această fază, a fost devansată modelarea interacțiunii pentru asistența șoferului unei mașini echipate cu dispozitivele de comunicare orală.
Activitatea de diseminare prevedea realizarea a 2 lucrări științifice, pe lângă actualizarea paginilor web ale proiectului (http://www.racai.ro/p/robin/). Au fost realizate 9 lucrări științifice deja publicate sau acceptate spre publicare în reviste cotate ISI, și au fost trimise, cel puțin, alte 3 lucrări la o conferință internațională ce se va desfășura în luna decembrie.
**Obiectivele au fost îndeplinite integral.**

## Descrierea științifică și tehnică

**Activitatea 3.12**. ”Implementarea sistemului de dialog în limbaj natural pentru micro-lumea unui robot asistiv/teleprezență” cu livrabilul **L9** - Descrierea sistemului de dialog în limbaj natural pentru micro-lumea unui robot asistiv.
În cadrul acestei activități au fost elaborate încă două microlumi (una la ICIA și UPB și alta la UTCN) pentru un robot asistiv în două ipostaze:

a) asistent casnic, pentru care robotul poate susține dialoguri referitoare la interogări/comenzi de genul: aprinde/stinge lumina, crește/scade/setează temperatura în cameră, pornește/oprește muzica/tv, adaugă/întreabă eveniment în calendar. A fost implementat un flux de prelucrare bazat pe RASA cu un manager de dialog implementat în Prolog. Pentru partea de prelucrare a vocii au fost folosite aplicațiile Google Speech. Contribuția este a partenerilor de la UTCN.
b) o persoană cu dizabilități sau în vârstă, pentru care au fost schițate 5 scenarii. Scenariile au fost documentate în fișiere de tip mw (microworld) la adresa src\main\resources\asistiv.mw din GitHub și pot fi utilizate cu managerul de dialog RDM.

## a) Descrierea scenariului Asistent Casnic (UTCN)

Un asistent casnic este un sistem conversațional inteligent care este capabil sa asiste utilizatorul în automatizarea diferitelor acțiuni ce țin de diferiții parametri ai casei (temperatura, lumina, etc), sau răspunde diferitelor interogări pe care utilizatorul le-ar putea efectua (e.g. întrebări despre vreme, planificarea activității zilnice, etc).

Pentru scenariul de asistent casnic, am definit mai multe tipuri de interogări/comenzi, intențiile și parametrii asociați acestora fiind prezentați în Tabelul 1.

Tabel 1 - Intențiile definite pentru scenariul de asistent casnic

| **Nume intentie (RO)** | **Parametri** | **Categorie** | **Tip** |
|---|---|---|---|
| *AprindeLumina* | Camera | Lumina | comanda |
| *StingeLumina* | Camera | Lumina | comanda |
| *ScadeIntensitateLumina* | Camera, nivel | Lumina | comanda |
| *CresteIntensitateLumina* | Camera, nivel | Lumina | comanda |
| *CresteTemperatura* | Camera, nivel | Temperatura | comanda |
| *ScadeTemperatura* | Camera, nivel | Temperatura | comanda |
| *SeteazaTemperatura* | Camera, nivel | Temperatura | comanda |
| *PuneMuzica* | Artist | Media | comanda |
| *OpresteMuzica* | - | Media | comanda |
| *CresteIntensitateMuzica* | Nivel | Media | comanda |
| *ScadeIntensitateMuzica* | Nivel | Media | comanda |
| *PornesteTV* | Canal | Media | comanda |
| *OpresteTV* | - | Media | comanda |
| *SchimbaCanal* | Canal | Media | comanda |
| *AdaugaEventCalendar* | Nume, data, ora inceput, durata | Calendar | comanda |
| *IntreabaEventCalendar* | Data, ora inceput, ora final | Calendar | interogare |
| *IntreabaVreme* | Loc, timp | Vreme | interogare |

### 1. Flux de procesare complet pentru sistemul de dialog

Orice sistem conversațional are o parte de înțelegere a intenției utilizatorului, și una de manager de dialog, care ia deciziile privind acțiunile/răspunsurile utilizatorului. Adițional, asemenea sisteme includ de regulă componente care transforma textul în vorbire, sau invers, precum generare de limbaj natural. Am proiectat un flux de procesare pentru un asemenea sistem, care integrează diferite componente pentru rezolvarea diferitelor sarcini. Figurile 1 și 2 prezinta fluxul în cele 2 variante: procesarea unei interogări, respectiv a unei comenzi. Figurile arată și deciziile de implementare posibile pentru fiecare componenta. Implementarea componentei de generare de

limbaj natural nu a fost momentan abordată.

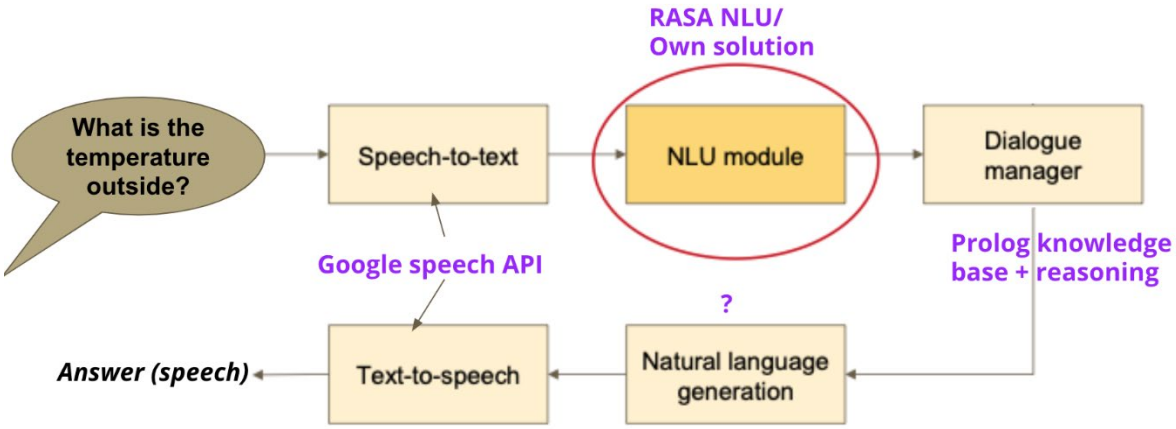


Figura 1 - Flux de procesare pentru cereri de tip ***interogare***

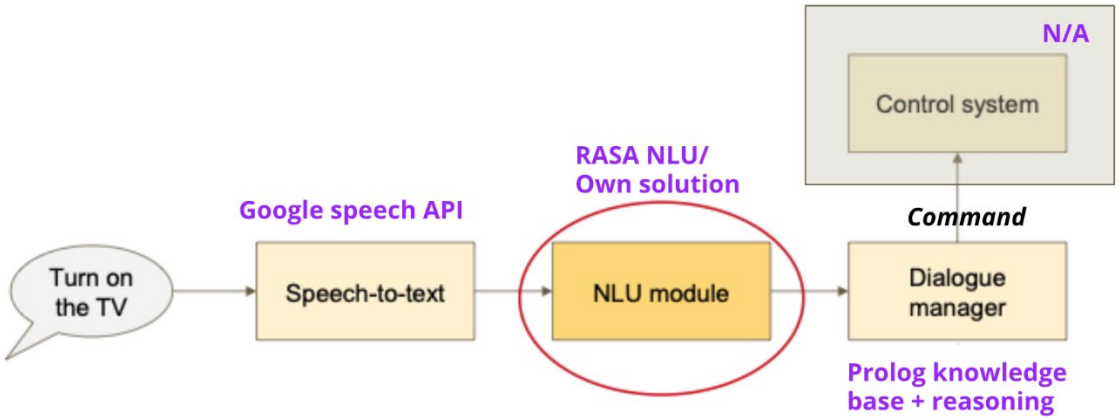


Figura 2 - Flux de procesare pentru cereri de tip ***comandă***

Ne-am concentrat pe implementarea modulelor de NLU (detecția intenției și a parametrilor acesteia), precum și pe implementarea unui prototip de manager de dialog pentru scenariul de asistent casnic.

**2.1 Detecția intenției și a parametrilor acestora - modulul NLU**

Am investigat doua opțiuni pentru implementarea modulului de detecție a intenției și a parametrilor acesteia: o soluție bazată pe RASA NLU [1], precum și o soluție proprie, bazată pe arhitecturi de tip Capsule Neural Nets [4]. Pentru implementarea ambelor soluții, a fost necesara mai întâi generarea unui set de date care sa permită învățarea intenției și a parametrilor asociați acesteia, pentru intențiile considerate de asistentul casnic, în limba romana. Am considerat pentru aceasta primele 14 intenturi din Tabelul 1 (următoarele 3 au fost adăugate în contextul managerului de dialog). În urma analizei de domeniu, am identificat o serie de provocări ce ar putea apărea în context real într-o asemenea aplicație, și am definit mai multe variante ale setului de antrenare - mai multe scenarii de analiza a performanței:

- Scenariul 0 - **baseline**: reprezintă setul de date de pornire fiind baza de dezvoltare a tuturor scenariilor următoare. În acest set de date putem găsi propoziții în limba romana cu diacritice, având o distribuție echilibrată (intre intenturi). Obiectivul este acela de a seta comportamentul în condiții optime (date de învățare cu o distribuție uniformă, cu date complete, fără zgomote, cu reprezentare a conținutului similara în învățare și testare).
- Scenariul 1 - **sinonime**: folosește setul de date de la Scenariul 0 și introduce o noua complexitate, și anume cuvinte (verbe) sinonime în setul de evaluare, sau formulări alternative (alte diateze ale verbelor, etc). Obiectivul este acela de a modela reprezentări diferite în învățare și respectiv testare (exprimări diferite ale aceluiași conținut în cele 2 situații).
- Scenariul 2 - **informație lipsă**: pornește de la Scenariul 1 la care adaugă complexitatea indusă de lipsa anumitor parametri în setul de date de test, pentru a putea observa comportamentul modelului în cazul cuvintelor cheie lipsă. Obiectivul este de a asigura o calitate satisfăcătoare în condițiile datelor incomplete, în vederea pregătirii pentru sistemul conversațional (realizat și descris în secțiunea 2. **Flux de procesare complet pentru sistemul de dialog**).
- Scenariul 3 - **dezechilibru**: utilizează Scenariul 2 ca și punct de pornire, la care adaugă complexitatea dezechilibrării intenturilor din setul de date dintr-o anumita super-clasa; acest scenariu are 3 sub-scenarii:
    - Scenariul 3.1, în care avem mai puține date despre temperatura;
    - Scenariul 3.2, în care avem mai puține date despre lumina;
    - Scenariul 3.3, în care avem mai puține date despre media.

  Obiectivul acestui scenariu este de a modela situațiile reale în care o categorie (de intenții) este (masiv) subreprezentată în învățare.

Rezultatele obținute de abordarea bazată pe RASA NLU sunt prezentate în Figura 3. O analiză calitativă a erorilor a indicat faptul că detecția este îngreunată de prezenta sinonimelor/ antonimelor, și a modului în care reprezentările distribuite (*word embeddings*) mapează aceste tipuri de relații: similaritatea intre sinonime este mai mica decât similaritatea intre antonime, ceea ce determina o confuzie pronunțată a intenturilor opuse.

| Pred/Act | L_ON | L_OFF | L_LO | L_HI | T_SET | T_LO | T_HI | M_ON | M_OFF | M_VOL | TV_ON | TV_OFF | TV_CH |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| L_ON | | 2, 3.3 | - | - | 2 | - | - | 3,3 | - | - | 1, 2, 3 | - | 2, 3.1 |
| L_OFF | 1, 2, 3 | | - | 1, 2 | 3,3 | - | 2 | 2 | 1, 3.1, 3.3 | - | 2, 3.2 | 3 | 2, 3.1 |
| L_LO | 2, 3 | 2, 3.2, 3.3 | | 1, 2, 3 | 3,1 | - | - | - | - | - | - | - | - |
| L_HI | 2, 3.1, 3.3 | 0, 3 | 2, 3 | | 3,1 | - | - | - | - | 3,2 | - | - | - |
| T_SET | 1 | - | - | - | | 1, 2, 3.1, 3.2 | 1, 2, 3 | 2, 3.1 | - | 3,3 | 2, 3.1, 3.2 | - | - |
| T_LO | - | - | - | - | 1, 2, 3 | | 1, 2, 3 | - | - | - | - | - | - |
| T_HI | - | - | - | 2 | 1, 2, 3 | 3.1, 3.2 | | - | - | 1, 3.2 | 2 | - | - |
| M_ON | - | 2, 3.1, 3.2 | 3,3 | - | 2, 3.1, 3.3 | - | 3,2 | | - | 3,3 | 1 | 3,2 | 2, 3 |
| M_OFF | - | 1, 3 | - | - | - | - | 3,3 | - | | - | - | 3.1, 3.2 | 3,3 |
| M_VOL | - | - | - | - | 1, 3.1, 3.3 | 3,2 | 3,2 | - | - | | - | - | 3,3 |
| TV_ON | 1 | - | - | - | 2 | 3,2 | - | 2 | 3 | 3,3 | | - | 1, 2, 3 |
| TV_OFF | 1 | 1, 3 | - | - | - | - | - | - | - | 1, 3.1, 3.2 | - | | 3 |
| TV_CH | - | - | - | - | - | - | - | - | - | 3 | - | - | |

Figura 3 - Rezultatele obținute de modelul RASA NLU – SUPERVISED EMBEDDINGS. Matricea indica scenariul din care încep să apară confuzii între clasele specifice

Prin urmare, am implementat o soluție de pre-procesare care sa “apropie” sinonimele, și sa evidențieze sensul opus al antonimelor [2,3]. Tabelul 2 prezinta performanta cantitativa a modelului bazat pe RASA înainte de a efectua aceasta preprocesare, respectiv tabelul 3 arată performanța modelului în prezența pre-procesării.

Tabel 2 - Performanta modelului RASA NLU fără pre-procesare

| | 0 | 1 | 2 | 3.1 | 3.2 | 3.3 |
|---|---|---|---|---|---|---|
| Intent F1 | 0.996 | 0.589 | 0.489 | 0.352 | 0.472 | 0.345 |
| Slot F1 | 0.998 | 0.885 | 0.534 | 0.553 | 0.501 | 0.745 |

Tabel 3 - Performanta modelului RASA NLU cu pre-procesare

| | 0 | 1 | 2 | 3.1 | 3.2 | 3.3 |
|---|---|---|---|---|---|---|
| Intent F1 | 0.998 | 0.708 | 0.677 | 0.466 | 0.585 | 0.411 |
| Slot F1 | 0.998 | 0.885 | 0.534 | 0.553 | 0.501 | 0.745 |

Se observă o creștere pronunțată a performanței de identificare a intentului corect, în toate scenariile de analiză. Totodată, tabelul 4 arată scăderea absolută a confuziilor de tip intenturi opuse, în prezenta pre-procesării.

Tabel 4 - Numărul de confuzii de intenturi opuse, cu diferite modele

| | 0 | 1 | 2 | 3.1 | 3.2 | 3.3 |
|---|---|---|---|---|---|---|
| FastText | 2 | 67 | 98 | 70 | 66 | 80 |
| Baseline | 1 | 70 | 85 | 59 | 61 | 55 |
| Our Solution | 1 | 41 | 28 | 17 | 24 | 24 |

Modelul propriu, bazat pe arhitecturi de tip Capsule Neural Networks, a obținut rezultatele prezentate în tabelul 5. Am comparat performanța cu modelul Wit.ai (Facebook) [5]. Pentru aceste modele nu s-a aplicat modulul de pre-procesare, dar analiza erorilor a indicat același fenomen de confuzie a intent-urilor opuse. în schimb, în urma analizei mecanismului de *self-atention* din aceste modele, s-a observat că - în anumite situații - atenția la detecție nu este de fapt pe verb. în prezent investigăm de unde ar putea apărea acest fenomen.

Tabel 5 - Comparația performantei modelelor CapsNets cu modelul Wit.ai

| | Wit.ai | | CapsNetI2S | | CapsNetS2I | |
|---|---|---|---|---|---|---|
| **Scenario** | **Intent F1** | **Slot F1** | **Intent F1** | **Slot F1** | **Intent F1** | **Slot F1** |
| **0** | **100** | 99.10 | 99.74 | 99.30 | **100** | **99.88** |
| **1** | 39.74 | **86.97** | 47.17 | 76.88 | **54.10** | 78.34 |
| **2** | **38.66** | **76.17** | 37.00 | 49.44 | 38.33 | 43.10 |
| **3.1** | 29.48 | **73.01** | 37.69 | 42.53 | **48.46** | 38.25 |
| **3.2** | 29.74 | **75.85** | 43.33 | 36.56 | **44.10** | 33.66 |
| **3.3** | 27.17 | **79.18** | **33.41** | 49.27 | 24.94 | 49.62 |

**2.2 Managerul de dialog pentru scenariul de asistent casnic**

Un prototip al managerului de dialog a fost implementat în Prolog, în 2 versiuni, exemplificând ultimele 3 intenturi din tabelul 1 (2 de tip interogare, 1 comandă). Alegerea acestor intenturi a fost făcută întrucât ele poseda o complexitate mai mare în ceea ce privește managerul de dialog. In ambele versiuni s-a urmărit asigurarea unei varietăți mari de fluxuri alternative de conversație, cum ar fi parametri lipsa și necesitatea solicitării explicite de informație asupra lor, sau furnizarea informației de timp în format relativ si/sau ambiguu, urmat de cereri de clarificare din partea managerului de dialog. Exemplificarea variațiilor poate fi văzută în doua video-uri disponibile online.

- Dialog Manager Version 1 demo: HomeAssistant_flux_procesare.mp4
  Cod: disponibil la cerere
- Dialog Manager Version 2 demo:
  https://www.youtube.com/watch?v=dLzlVVpMKW8&ab_channel=Marcus
  Cod: https://github.com/MarcusGitAccount/home_assistant_ro

**3. Referinte**


[1] Rasa. Rasa NLU: Language Understanding for Chatbots and AI assistants. Available: https://rasa.com/docs/rasa/nlu/about/

[2] N. Mrksic, D. O'Seaghdha, B. Thomson, M. Gasic, L. M. Rojas-Barahona, P.-H. Su,D. Vandyke, T.-H. Wen, and S. Young, "Counter-fitting word vectors to linguistic constraints", Conference of the North American Chapter of the Association for Computational Linguistics.

[3] J. Kim, G. Tur, A. Celikyilmaz, B. Cao, and Y. Wang, "Intent detection using se-mantically enriched word embeddings," în 2016 IEEE Spoken Language TechnologyWorkshop (SLT), 2016, pp. 414–419.

[4] C. Zhang, Y. Li, N. Du, W. Fan, and P. Yu, "Joint slot filling and intent detection via capsule neural networks", în *Proc. of ACL*, 2019

[5] Wit.ai: Natural Language for Developers. [Online]. Available: https://wit.ai/


## b) Descrierea scenariilor pentru un robot asistent al unei persoane cu dizabilități sau în vârstă (ICIA)

Așa cum am arătat în rapoartele anterioare, pentru a putea susține un dialog coerent și cooperant cu utilizatorul, robotului Pepper trebuie sa i se formalizeze conceptele și acțiunile relevante în micro-lumea respectivă. În cele ce urmează este prezentată reprezentarea parțială a micro-lumii asistentului social al unei persoane cu dizabilități sau în vârstă (care poate fi extinsă, în conformitate cu regulile sintactice ale limbajului de definire a micro-lumilor). Dăm aici listingul fișierului `src\main\resources\asistiv.mw` din GitHub. Acest fișier-definiție a unei micro-

lumi de robot asistiv dă posibilitatea managerului de dialog RDM să răspundă punctual unor întrebări cum ar fi: „Ce medicament trebuie să iau astăzi?” / „Ce zi este astăzi?” / „Cât e ceasul?” / „Câte grade sunt afară?” / „Trebuie să iau Extraveral astăzi?” / etc.

```
CONCEPT cine -> PERSON
CONCEPT unde -> LOCATION
CONCEPT când -> TIME
CONCEPT oră, ceas -> TIME
CONCEPT zi, azi, astăzi -> TIME
CONCEPT grad, temperatură -> WORD
CONCEPT medicament -> WORD
CONCEPT masă -> WORD

REFERENCE oră ro.racai.robin.dialog.generators.TimeNow = G1
REFERENCE grad ro.racai.robin.dialog.generators.DegreesNow = G2
REFERENCE zi ro.racai.robin.dialog.generators.DayNow = G3
REFERENCE medicament Extraveral = M1
REFERENCE medicament Thyrozol = M2
REFERENCE masă prânz, masa de prânz = E1
REFERENCE masă cină, masa de seară = E2

TIME ora 06:00 dimineața = T1
TIME ora 06:00 după-amiază = T2
TIME luni = Z1
TIME marți = Z2
TIME miercuri = Z3
TIME joi = Z4
TIME vineri = Z5
TIME sâmbătă = Z6
TIME duminică = Z7

PERSON fratele meu = P1
PERSON sora mea = P2
PERSON părinții mei = P3

PREDICATE lua, administra -> SAY_SOMETHING
PREDICATE veni -> SAY_SOMETHING

# Pepper știe când utilizatorul trebuie să ia medicamentele
# Utilizatorul trebuie să ia medicamentul M1 (Extraveral)
# în ziua Z1 (luni)
TRUE lua M1 Z1
TRUE lua M1 Z3
TRUE lua M1 Z5
```

```
TRUE lua M1 Z7
TRUE lua M2 Z2
TRUE lua M2 Z4
TRUE lua M2 Z6

# Pepper știe cine vine la masă și în ce zi
# Persoana P1 (fratele utilizatorului) vine la masa de prânz (E1) în
# ziua Z1 (luni)
TRUE veni P1 E1 Z1
TRUE veni P2 E2 Z3
TRUE veni P3 E2 Z6
```

În continuare sunt exemplificate scenarii de interacțiune, acoperite de definițiile de mai sus:

**Scenariul 1:**
User: Ce zi este astăzi?
Pepper: Este luni / marți / miercuri / joi / vineri / sâmbătă / duminică.
User (*nu mai știe ce are de făcut*): Am ceva astăzi în calendar? / Ce trebuie să fac astăzi?
Pepper: Da, azi trebuie să uzi florile. / E ziua de udat florile.
User: Anunț-o pe asistentă.
Pepper: Am anunțat-o.
User: Vine cineva în vizită la prânz / cină, azi?
Pepper: Fiul / nepotul tău / fiica / nepoata ta vine la masă.

**Scenariul 2:**
User: Am închis robinetul în baie? / Am închis robinetul în bucătărie? / Am închis aragazul (în bucătărie)?
Pepper (*presupunând că Pepper știe că aragazul e în bucătărie*): Merg să verific.
User: Ce ai observat?
Pepper (*după ce revine*): Da, l-ai închis. / Nu, nu l-ai închis!

**Scenariul 3:**
User: Pepper, unde-mi sunt ochelarii?
Pepper: Ochelarii sunt pe masa din dormitor.
User: Pepper, unde mi-am lăsat cartea / șosetele / telefonul / portofelul / cheile?
Pepper (*presupunând că le-a văzut deja, plimbându-se prin casă*): Pe masă în sufragerie / Sub pat / În bucătărie / În hol / În baie.

**Scenariul 4:**
User: Ce am de făcut la ora asta, Pepper?
Pepper: Este timpul să îți iei tensiunea / măsori pulsul!
User: Pepper, amintește-mi ce mai trebuie să fac!
Pepper: Ți-ai luat medicamentele la prânz / în această dimineață / seară?

User: Ce medicamente trebuie să iau acum?
Pepper: La prânz / Dimineața / Seara trebuie să iei Extraveral și Thyrozol.
User: Care se ia cu apă?
Pepper: Ia cu apă toate medicamentele.
User: Unde sunt medicamentele?
Pepper: Medicamentele sunt pe dulapul / masa din bucătărie / sufragerie / dormitor.
User: Pepper, ce mai am de făcut?
Pepper: Ți-ai făcut gimnastica astăzi?
User: Da. / Nu, o fac acum.
Pepper: Îmbracă-te comod!

**Scenariul 5:**
User: Pepper, nu mă simt bine.
Pepper: Cum anume te simți?
User: Sunt cam obosit / obosită / moleșit / moleșită / fără vlagă. / Am palpitații. / Mă doare capul. / Am amețeli. / Mi-a amorțit un picior / o mână.
Pepper: Vrei să chem asistenta?
User: Da. / Da, te rog. / Te rog. / Nu, mulțumesc. / Nu. / Mulțumesc!

**Activitatea 3.13** Implementarea sistemului cadru de dialog în limbaj natural, cu livrabilele L10- Tehnologie nouă dialog în limbaj natural pentru micro-lumea unui robot asistiv și L11- Descrierea modulului configurabil de dialog în limba română, Modul software configurabil de dialog în limba română

### L10.1 Îmbunătățiri aduse managerului de dialog RDM (ICIA)

Managerul de dialog pentru micro-lumi ROBIN Dialog Manager (RDM, https://github.com/racai-ai/ROBINDialog) a fost descris în lucrarea „A Dialog Manager for Micro-Worlds" (Ion et al., 2020) și care, de la momentul scrierii până în prezent, a primit următoarele îmbunătățiri:

1. *Referințe ale conceptelor din micro-lume rezolvate prin apeluri către clase Java*. În acest mod, putem răspunde în limba română unor întrebări care necesită un anumit tip de calcul. De exemplu, la întrebarea „Cât este ceasul?", RDM trebuie să afle ora curentă din sistem și s-o traducă în fraza corespunzătoare în limba română. Acest calcul este implementat în clasa `ro/racai/robin/dialog/generators/TimeNow.java`. Pentru a preciza referința substantivului comun „oră" în definiția micro-lumii, vom adăuga linia
   ```
   REFERENCE oră ro.racai.robin.dialog.generators.TimeNow = G1
   ```
   împreună cu predicatul
   ```
   TRUE fi G1
   ```
   Clasa `ro/racai/robin/dialog/generators/DegreesNow.java` ne permite să răspundem întrebării „Câte grade sunt afară?" sau „Ce temperatură e afară?", utilizând

informații despre locul unde se află RDM (după adresa IP) și serviciul web gratuit pus la dispoziție de Agenția Națională de Meteorologie (http://www.meteoromania.ro/).

2. Modulul de „Text-To-Speech (TTS)" a fost înlocuit cu unul mult mai rapid, până când găsim resurse pe care să instalăm modulul TTS descris în lucrare, bazat pe rețele neuronale, care are o intonație mai apropiată de realitate. Am folosit librăria SSLA (Boroș et al., 2018), scrisă în Java, care sintetizează foarte rapid (sub o secundă) fraze de 10-20 de cuvinte.
3. Modulul de „Automatic Speech Recognition" (ASR) a fost înlocuit cu unul mult mai performant (cu o eroare medie de recunoaștere a cuvintelor de trei ori mai mică decât modulul ASR descris în lucrare) și mai rapid, bazat pe rețeaua neuronală complexă Deep Speech 2 (Avram et al., 2020). În cele ce urmează, vom detalia acest modul de ASR.

## L10.2 Modul ASR pentru limba română bazat pe Deep Speech 2 (ICIA)

Avram et al. (2020) descriu un modul ASR pentru limba română bazat pe Deep Speech 2. Modulul a fost antrenat pe cca. 230 de corpus bimodal (voce și text) și reușește să obțină o rată medie a erorii de recunoaștere la nivel de cuvânt (eng. „Word Error Rate" sau WER) de 9.9%. Modulul poate transcrie fraze rostite în 70 ms pe frază, un timp de răspuns excelent pentru o aplicație cum e RDM în care utilizatorul așteaptă răspunsul într-un timp rezonabil.

Rata de eroare de 9.9% încorporează erori de recunoaștere a folosirii cratimei în limba română pentru a separa clitice sau pentru a lega cuvintele în rostire. Am dezvoltat module speciale de corectură pentru aceste situații, descrise în cele ce urmează.

### Module de corectură a transcrierii ASR

Modulul de recunoaștere a vorbirii (Avram et al., 2020) dezvoltat în cadrul proiectului ROBIN, deși are performanțe foarte bune (eroarea la nivel de cuvânt WER sub 10%) pentru texte apropiate celor din setul de antrenament, poate avea dificultăți în adaptarea la un vorbitor necunoscut sau la texte complet diferite celor de antrenament. Având în vedere acest aspect, precum și caracteristicile micro-lumilor investigate în proiectul ROBIN, au fost dezvoltate două module de corecție a textului recunoscut: restaurare cratimă + capitalizare pentru cuvinte cunoscute și corecție cuvinte necunoscute.

Modulele de corectură au fost implementate sub forma unor servicii web REST, expuse prin intermediul protocolului HTTP. Acestea permit transferul textului rezultat în urma procesului de recunoaștere a vorbirii și corectarea acestuia precum și returnarea textului final rezultat către aplicația client. Având în vedere specificul dialogului din proiectul ROBIN (propoziții unice sau texte relativ scurte), interogarea serviciilor web se poate realiza prin intermediul metodei HTTP GET și utilizarea unui parametru din structura URL-ului aferent fiecărui serviciu.

Rezultatul întors este sub formă de document JSON. Textul corectat se regăsește în proprietatea "text". Totodată pentru a ușura integrarea în aplicații în documentul JSON rezultat se regăsește și un câmp "status" care va conține valoarea "OK" dacă procesarea s-a efectuat cu succes.

Adițional, JSON-ul rezultat conține un câmp ”comments” care conține diferite informații despre aplicarea modelului și deciziile luate de acesta. Aceste informații putând fi apoi utilizate pentru a înțelege anumite corecții efectuate precum și pentru a investiga eventuale opțiuni pentru îmbunătățirea performanțelor.

**Modulul pentru restaurarea cratimei și a majusculelor în cuvintele cunoscute**

Considerând două forme de scriere (cu sau fără cratimă): ”sau” / ”s-au” acestea au semnificații diferite. Exemplu: ”Cei doi prieteni s-au dus la munte sau la mare.” În primul caz ”s-au” avem un pronume reflexiv ”s-” și un verb auxiliar ”au”, în timp ce în al doilea caz ”sau” este o conjuncție. Astfel, recunoașterea eronată a lui ”s-au” în forma fără cratimă conduce la erori în etapele ulterioare ale procesărilor din cadrul proiectului ROBIN.

Modulul de corecție realizat utilizează un model bazat pe bigrame de forma ($W_k$, $W_{k+1}$) pentru a corecta cuvântul curent $W_k$. Pentru a reduce dimensiunea modelului, au fost incluse doar formele $W_k$ care acceptă atât scrierea cu cratimă cât și fără. Acest model a fost antrenat utilizând corpusul CoRoLa (Tufiș et al, 2019). Dacă modelul nu poate identifica un bigram (posibil ca urmare a unei probleme în recunoașterea cuvântului următor), se recurge la un model unigram bazat pe frecvențele de apariție a celor două forme (cu și fără cratimă).

Scrierea corectă, cu majuscule atunci când sunt nume proprii, joacă de asemenea un rol important în procesările ulterioare. Astfel, identificarea părților de vorbire poate utiliza majusculă întâlnită în mijlocul propoziției pentru a identifica un substantiv propriu sau procesele de recunoaștere a entităților pot utiliza majuscula ca factor care contribuie la identificarea unui cuvânt ca parte a unei entități. Pentru a corecta cuvintele și a introduce majuscule acolo unde este cazul, a fost utilizată o listă de cuvinte cunoscute ca fiind asociate entităților cu nume (persoane, locații, organizații). Având în vedere că se urmărește doar corectarea literelor mici în majuscule, nu este relevant dacă același cuvânt poate avea mai multe semnificații (cum ar fi o locație care poate fi utilizată și în cadrul unei entități de tip organizație).

**Modulul pentru corectarea cuvintelor necunoscute**

Proiectul ROBIN se bazează pe existența unor micro-lumi în care este permisă interacțiunea om-robot. Astfel, vocabularul aferent acestor micro-lumi este relativ redus, ceea ce permite presupunerea că un cuvânt care nu există în vocabular este cel mai probabil recunoscut greșit de sistemul de recunoaștere a vorbirii. Astfel, modulul pentru corectarea cuvintelor necunoscute încearcă să înlocuiască orice cuvânt necunoscut cu un cuvânt apropiat care poate fi utilizat corespunzător contextului întâlnit. Se presupune că nu există o întreagă secvență de cuvinte recunoscută eronat.

Au fost construite 2 modele bigram ($W_k$,$W_{k+1}$), ($W_{k-1}$, $W_k$) și un model unigram ($W_k$). Ca și la modulul anterior, modelele au fost antrenate utilizând corpusul CoRoLa. În vederea corectării,

la identificarea unui cuvânt $W_k$ necunoscut, se încearcă identificarea posibilelor cuvinte corecte pe baza celor 3 modele. Cuvântul ales corespunde celui care se poate găsi în contextul curent și are distanța Levenshtein cea mai mică. Pentru a evita calculul tuturor distanțelor Levenshtein este impusă o limită ca diferență de dimensiune maximă între cuvântul curent și un potențial candidat. Algoritmul este descris în Figura 4.

*1. Pentru toate cuvintele necunoscute $W_k$*

*1.1. Sunt identificate 2-gram ($W_{k-1}$, $N_k$ ; $N_k$, $W_{k+1}$) existente în model; dacă nu există se caută 1-gram ($N_k$=tot vocabularul de dimensiuni apropiate ca număr de caractere)*

*1.2. Pentru toate alternativele identificate (dacă sunt mai mult de 1) calculează distanța Levenshtein cu cuvântul curent*

*1.3. Selectează cuvântul corespunzător modelului n-gram, dar cu distanța Levenshtein cea mai mică*

**Figura 4.** Algoritmul pentru corectarea cuvintelor necunoscute

**Activitatea 3.13** ”Implementarea sistemului cadru de dialog în limbaj natural” cu livrabilul L11 ”Descrierea modulului configurabil de dialog în limba română, Modul software configurabil de dialog în limba română”

## Integrarea modulelor de corectare (ICIA)

Având în vedere specificitatea modulelor pentru proiectul ROBIN, precum și faptul că mecanismul general de recunoaștere a vorbirii (ASR + corecturi) poate fi utilizat în diferite scenarii, este necesară integrarea diferitelor componente în funcție de necesitățile specifice ale aplicației care le utilizează. Este posibilă, astfel, fie utilizarea directă a ieșirii modulului ASR, fie combinarea unuia sau mai multor module de corectură. Schema bloc a unui sistem care realizează integrarea diferitelor module este prezentată în Figura 5.

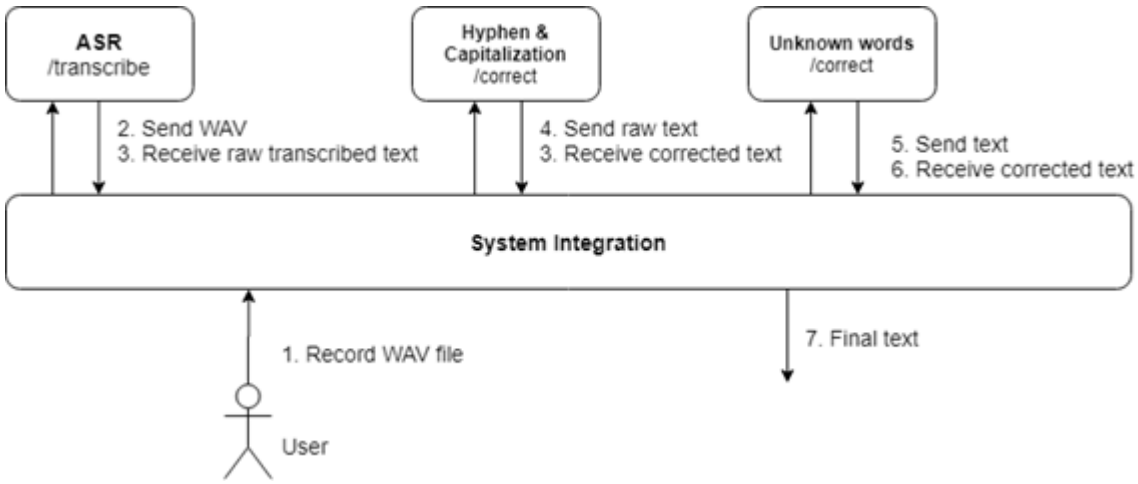

**Figura 5.** Schema bloc a unui sistem care realizează integrarea sistemului ASR cu modulele de corectură

În vederea testării acestei integrări, a fost realizată o pagină web demonstrativă în cadrul platformei RELATE (Păiș et al., 2020), disponibilă online la adresa http://relate.racai.ro. Utilizând platforma, este posibilă încărcarea sau înregistrarea unei propoziții în format audio (fișier WAV). Ulterior, componenta de integrare apelează diferitele module (ASR, corectură cratimă + capitalizare, corectură cuvinte necunoscute), conform diagramei prezentată în Figura 5. Rezultatul final este apoi afișat utilizatorului, fiind posibil apoi transferul automat al acestui rezultat către analiză, utilizând facilitățile oferite de platforma RELATE. În Figura 6 este prezentată componenta de interfață aferentă platformei RELATE care permite înregistrarea sunetului în vederea utilizării sistemului de recunoaștere a vorbirii. Ulterior, în Figura 7 este prezentat rezultatul recunoașterii vorbirii.

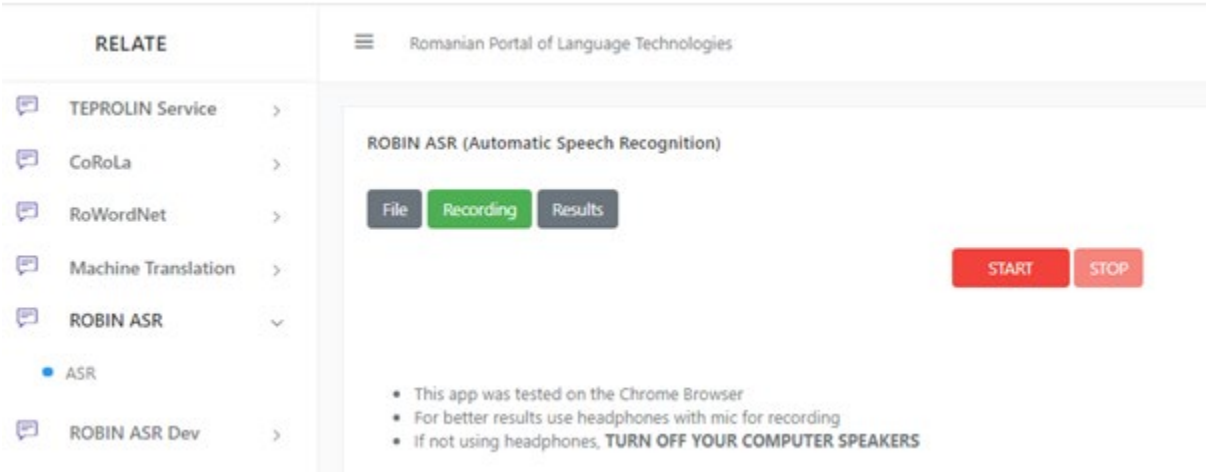


**Figura 6.** Componenta de interfață din platforma RELATE care permite înregistrarea sunetului și transferul către componenta de recunoaștere a vorbirii

Componenta de înregistrare a fost documentată cu un manual de utilizare ce arată pas-cu-pas operațiile necesare pe care trebuie să le facă un utilizator pentru a genera un fișier audio și a-l trimite componentei ASR.

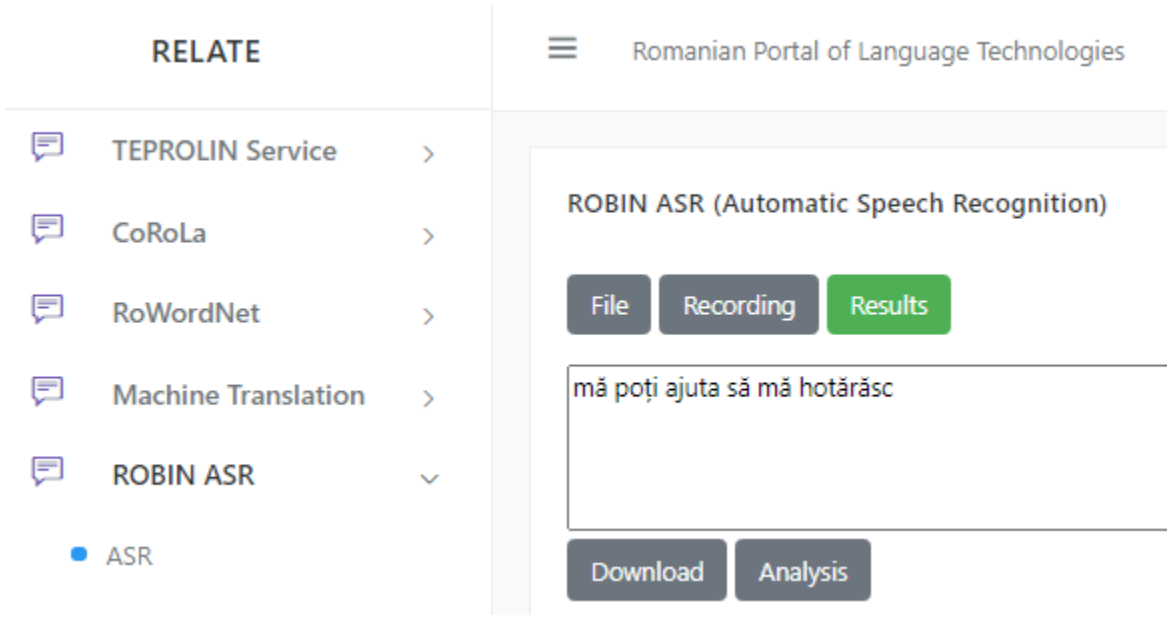


**Figura 7.** Recunoașterea vorbirii înregistrate prin platforma RELATE ca urmare a trecerii semnalului prin toate modulele de corectură, conform diagramei bloc din Figura 4.

Ulterior recunoașterii sunetului și corecturii, în cadrul platformei se poate invoca automat serviciul TEPROLIN (Ion, 2018) care permite analiza textului prin apăsarea butonului „Analysis” din Figura 7 (același lucru fiind realizat și intern în cadrul proiectului ROBIN). Prin intermediul afișării rezultatelor în cadrul platformei RELATE, este posibilă verificarea diferitelor nivele de adnotare, precum și explorarea grafică a relațiilor dintre componentele textuale, care permite ajustarea algoritmilor de nivel superior utilizați apoi în cadrul proiectului ROBIN. Figura 8 prezintă grafic structura frazei recunoscută de sistemul de recunoaștere a vorbirii.

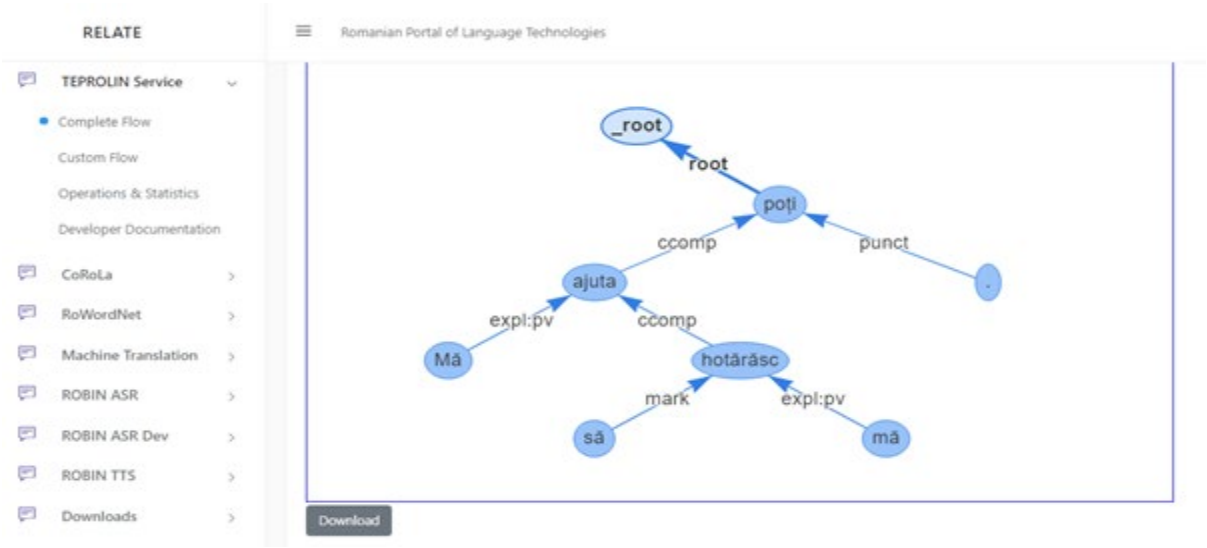


**Figura 8.** Structura frazei recunoscută de sistemul de recunoaștere a vorbirii

## Îmbunătățiri ale modelului ASR (ICIA)

În vederea îmbunătățirii modelului ASR și a adaptării acestuia la specificul micro-lumilor ROBIN, a fost creat un corpus bimodal (text + voce). Acesta este format din 700 de propoziții specifice antrenării unui ajutor în vânzări aferent unui magazin de calculatoare. În vederea înregistrării acestora, a fost dezvoltat un modul nou în platforma RELATE care permite încărcarea unui fișier CSV cu propozițiile și apoi înregistrarea semnalului audio aferent acestora de către utilizatorii platformei. Figura 9 prezintă un interfața de înregistrare implementată.

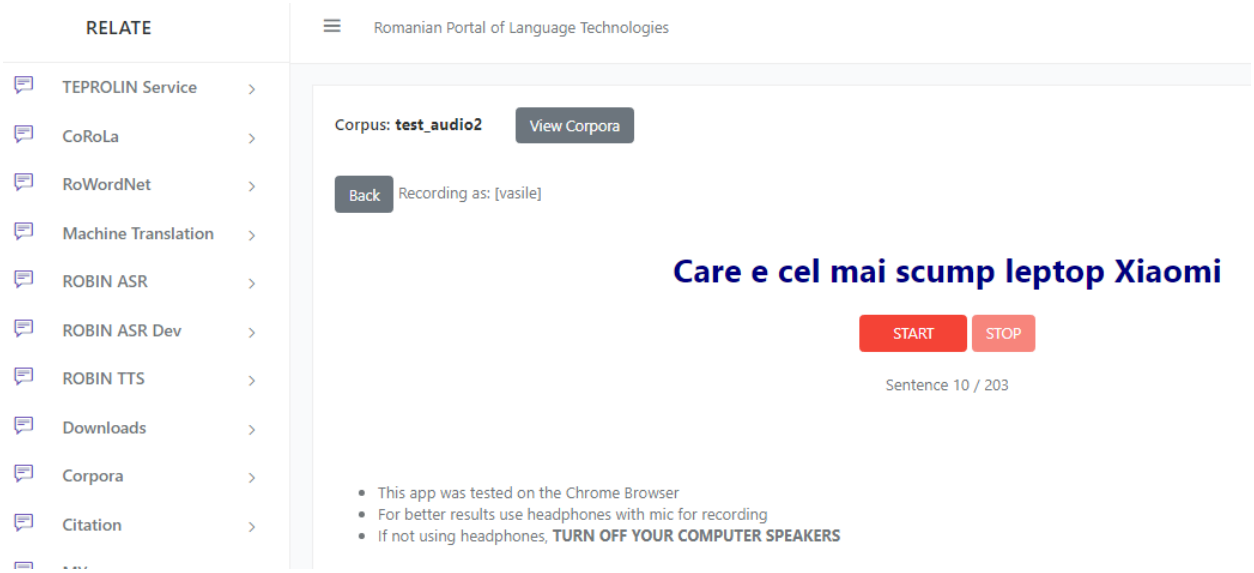


**Figura 9.** Interfața de înregistrare sunet implementată în platforma RELATE

Pentru a ușura munca celor care înregistrează sunetul și totodată pentru a asigura o aliniere perfectă cu textul, propozițiile din corpus conțin o scriere apropiată de pronunție. Astfel în loc de cuvântul „laptop” a fost utilizată scrierea „leptop” pentru a indica persoanei care înregistrează pronunția dorită.

În vederea înregistrării sunetului, au fost înregistrați în platformă 6 utilizatori. Deoarece nu toți utilizatorii vor înregistra integral textele aferente corpusului, s-a realizat o împărțire a acestuia în 4 sub-corpusuri. Fiecare utilizator va începe înregistrările cu un anumit sub-corpus astfel încât la final să fie asigurată existența a cel puțin doi vorbitori pentru fiecare sub-corpus. Schema propusă pentru înregistrare este prezentată în tabelul următor:

| **Vorbitor** | **Ordine înregistrare sub-corpusuri** |
|---|---|
| 1 | 1,2,3,4 |
| 2 | 2,3,4,1 |

| | |
|---|---|
| 3 | 3,4,1,2 |
| 4 | 4,1,2,3 |
| 5 | 1,3,4,2 |
| 6 | 3,1,4,2 |

**Tabelul 6**. Ordinea propusă de înregistrare pentru fiecare vorbitor.

În timpul înregistrării, utilizatorii pot asculta semnalul audio înregistrat și pot decide ștergerea fișierului dacă nu corespunde din punct de vedere calitativ. Această interfață este prezentată în Figura 10. Ulterior, fișierul poate fi reînregistrat.

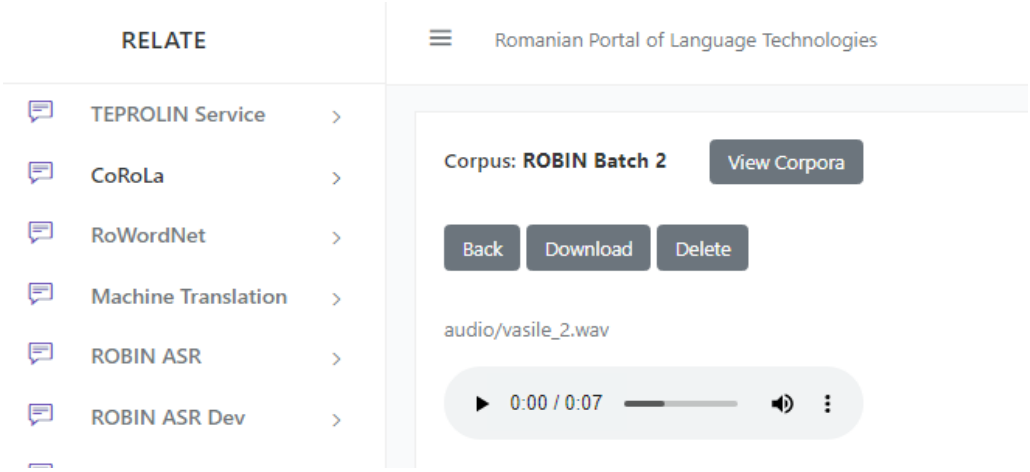


**Figura 10.** Interfața care permite redarea fișierului audio înregistrat și ștergerea acestuia

La finalul înregistrărilor, înainte de includerea corpusului bimodal rezultat în alte prelucrări, corpusul urmează să fie anonimizat și îmbogățit cu metadate, descriind cel puțin modalitatea de realizare, caracteristicile tehnice ale înregistrărilor, numărul de vorbitori, caracteristici ale vorbitorilor, etc.

## Descrierea sistemului de dialog în limbaj natural pentru asistența șoferului (ICIA)

Sistemul pentru asistența șoferului poate interacționa atât cu persoanele din mașină prin interacțiune verbală (persoanele din mașină rostesc întrebări iar sistemul răspunde la acestea tot verbal) cât și cu computerul mașinii (trimite comenzi mașinii pentru a fi executate și primește comenzi verbale din partea acesteia – vezi Figura 11).

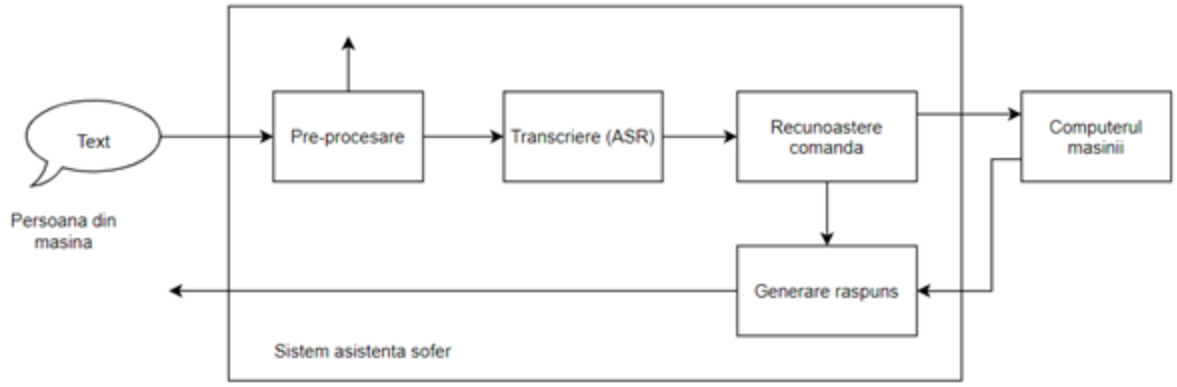


**Figura 11**. Schema bloc a sistemului de dialog pentru asistența șoferului

Principalele blocuri ale sistemului pentru asistența șoferului sunt:

- **Blocul de preprocesare**: are rolul de a verifica dacă semnalul audio captat este suficient de puternic pentru a putea fi considerat o cerere (evitând astfel activarea sistemului atunci când persoanele din mașină discută între ele, de exemplu). Semnalele audio slabe nu trec de blocul de preprocesare, prelucrarea lor oprindu-se aici. În schimb, semnalele audio puternice trec și sunt filtrate înainte de a fi trimise mai departe. Filtrarea are loc deoarece mediul din mașină poate fi unul zgomotos: discuții pe fundal, muzică pe fundal, etc.

- **Blocul de transcriere**: Același modul de detecție automată a vorbirii (ASR) folosit și pentru ROBIN Dialog.

- **Blocul de recunoaștere a comenzii**: similar cu managerul de dialog, blocul de recunoaștere identifică ținta comenzii (conceptul) și modul în care se dorește acționat asupra ei (pornire/oprire). De exemplu, următoarele texte sunt echivalente: „Pornește radioul!”/„Deschide radioul!” sau „Pornește faza scurtă!”/„Aprinde luminile de întâlnire!”. Dacă se găsește o comandă corectă, ea va fi trimisă către calculatorul mașinii pentru executare, altfel sistemul va genera singur un răspuns de clarificare pentru a cere persoanei să repete. Răspunsul de clarificare nu obligă utilizatorul să spună ceva, ci doar îl atenționează asupra faptului că sistemul de asistență nu a putut să recunoască o comandă (întrucât există riscul ca anumite semnale audio să treacă în mod incorect de blocul de preprocesare).

- **Blocul pentru generarea răspunsurilor**: redă unul sau mai multe fișiere audio preînregistrate. Poate fi apelat intern de către blocul de recunoaștere a comenzii sau extern (de către computerul mașinii). În urma primirii unei comenzi, computerul mașinii poate trimite înapoi un răspuns pozitiv („Luminile de avarie au fost pornite!”), un răspuns negativ („Radioul este deja închis.”) sau un răspuns general de eroare (pentru a permite șoferului să verifice problema apărută, de exemplu s-a ars un far). Computerul mașinii poate genera

răspunsuri fără să fie nevoie să primească o comandă. De exemplu, dacă modul de vedere artificială identifică pietoni pe partea dreaptă, computerul mașinii poate genera răspunsul compus „pieton”, „pe dreapta”. În mod similar se pot genera răspunsuri compuse pentru a descrie și alte rezultate ale modului de vedere artificială.

# Referințe


Avram, A.M., Păiș, V., Tufiș, D. (2020) Towards a Romanian end-to-end automatic speech recognition based on Deep Speech 2. în Proceedings of the Romanian Academy, Series A, in-print.

Boroș T., Dumitrescu Ș. D. și Păiș V. (2018). Tools and resources for Romanian text-to-speech and speech-to-text applications. arXiv:1802.05583 [cs.CL]

Ion, Radu. (2018). TEPROLIN: An Extensible, Online Text Preprocessing Platform for Romanian. în Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR 2018), November 22-23, 2018, Iași, România.

Ion R., Badea V. G., Cioroiu G., Barbu Mititelu V., Irimia E., Mitrofan M. și Tufiș D. (2020). A Dialog Manager for Micro-Worlds. Studies în Informatics and Control, 29(4) 401-410, December 2020. ISSN: 1220-1766

Păiș, V., Tufiș, D., Ion, R. (2020) A Processing Platform Relating Data and Tools for Romanian Language. în Proceedings of the 12th Language Resources and Evaluation Conference (LREC), European Language Resources Association, Marseille, France, pages 81-88.

Tufiș, D., Mititelu, V.B., Irimia, E., Păiș, V., Ion, R., Diewald, N., Mitrofan, M., Onofrei, M. (2019). Little strokes fell great oaks. Creating CoRoLa, the reference corpus of contemporary Romanian. în Revue Roumaine de linguistique, LXIV (3).


## **Activitatea 3.14** Diseminare

Andrei-Marius Avram, Vasile Păiș, Dan Tufiș (2020). Towards a Romanian end-to-end automatic speech recognition based on DeepSpeech2. In Proceedings of the Romanian Academy, Series A, vol. 21, no. 4, ISSN : 1454-9069, in print

Radu ION, Valentin Gabriel BADEA, George CIOROIU, Verginica BARBU MITITELU, Elena IRIMIA, Maria MITROFAN and Dan TUFIȘ (2020). A Dialog Manager for Micro-Worlds, In Studies in Informatics and Control, volume-29-issue4-2020, ISSN: 1220-1766, in print

Vasile Păiș, Radu Ion, Dan Tufiș (2020). A Processing Platform Relating Data and Tools for Romanian Language. In: Proceedings of the 1st International Workshop on LanguageTechnology Platforms (IWLTP 2020), European Language Resources Association (ELRA), George Rehm et al. (eds.), pp. 81-88 - indexed by DBLP and ISI Web of Science.

Păiș, Vasile and Ion, Radu (2020). TermEval 2020: RACAI's automatic term extraction system. In *Proceedings of the 6th International Workshop on Computational*

*Terminology*. European Language Resources Association, Marseille, France, pp. 101-105, indexed by DBLP and ISI Web of Science, May 2020

Georg Rehm... Tufiș, Dan .... (2020). The European Language Technology Landscape in 2020: Language-Centric and Human-Centric AI for Cross-Cultural Communication in Multilingual Europe. In *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, pp. 3315-3325, May 2020, indexed by DBLP and ISI Web of Science

Toncu, S., Toma, I., Dascălu, M., & Trăușan-Matu, S. (2020). Escape from Dungeon – Modelling User Intentions with Natural Language Processing Techniques. In *5th Int. Conf. on Smart Learning Ecosystems and Regional Development (SLERD 2020)*. Online: Springer.

Nenciu, B., Corlatescu, D. C., & Dascălu, M. (2020). RASA Conversational Agent in Romanian for Predefined Microworlds. In *International Conference on Human-Computer Interaction (RoCHI2020)*. Online: MatrixRom.

Boroghina, G., Corlatescu, D. C., & Dascălu, M. (2020). Conversational Agent in Romanian for Storing User Information in a Knowledge Graph. In *International Conference on Human-Computer Interaction (RoCHI2020)*. Online: MatrixRom.

A.-D. Stoica, A.-C. Rad, I.-H.-M. Muntean, G. Dăian, C. Lemnaru, R. Potolea, M. Dînșoreanu, "The Impact of Romanian Diacritics on Intent Detection and Slot Filling" - *2020 IEEE International Conference on Automation, Quality and Testing, Robotics* (AQTR), Cluj-Napoca, Romania, 2020, pp. 1-6, doi: 10.1109/AQTR49680.2020.9129947

Toate lucrările menționează cu multumiri, finanțarea cercetărilor de către proiectul ROBIN-Dialog. Site-ul proiectului ROBIN-Dialog a fost actualizat cu rapoartele integrale și lucrările științifice realizate.

## Prezentarea structurii ofertei de servicii de cercetare și tehnologice cu indicarea link-ului din platforma Erris

După cum s-a arătat mai sus, toate programele și structurile de date aferente lor au fost plasate în github-ul public https://github.com/racai-ai/ROBINDialog, clonabil și în platforma ERRIS. Utilizatorii interesați pot folosi serviciile implementate (analiza textelor- segmentare, lematizare, analiză morfo-lexicală, analiză sintactică, recunoașterea entităților cu nume, traducere automată RO-EN-RO, recunoașterea vorbirii, sinteza vorbirii, înregistrarea vorbirii) prin intermediul unui browser accesând platforma RELATE http://relate.racai.ro.

## Locuri de munca susținute prin program, inclusiv resursa umană nou angajată

6 cercetători cu vechime în ICIA (D. Tufis, V. Mititelu, E. Irimia, M. Carp (fostă Mitrofan), R. Ion, E. Curea) plus 2 tineri cercetători angajați pe proiect (G. Cioroiu, V. Badea).

### Detalii privind angajarea si menţinerea noilor cercetători

| | |
|---|---|
| **Nr. posturi asumate de noi cercetători** | 2 |
| **Nr. posturi ocupate de noi cercetători** | 2 |

ICIA le-a facut contracte de muncă normă întreagă pe doi ani celor doi cercetători angajați pe proiect, ei urmând a fi plătiți din surse bugetare și în funcție de implicarea lor activă din fonduri extrabugetare pe baza pontajului făcut de directorii proiectelor respective (CURLICAT, MARCELL, ELRC, ELG, ELE și eventual MULTIMORE-aflat inca in evaluare).

| **Lista noi cercetători** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **Nr. crt.** | **Instituție** | **Nume** | **Prenume** | **Poziția ocupata in cadrul proiectului** | **Data angajare in proiect** | **Perioada implicare in proiect** | **Costuri salariale alocate** | **Costuri salariale plătite** |
| 1. | ICIA | Cioroiu | George | ASC | 01.11.2018 | 01.11.2020 | 100.000 | 120.000 |
| 2 | ICIA | Badea | Valentin Gabriel | ASC | 01.12.2018 | 01.11.2020 | 100.000 | 115.000 |

## Prezentarea valorificării/ îmbunătățirii competențelor/ resurselor existente la nivelul consortului (cecuri)

Sumele alocate cec-urilor nu au fost valorificate, în principal pentru că nu s-au identificat oportunități conforme cu reglementările de acordare.