

**Modele de inteligență artificială (deep learning) aplicate în analiza
sintactico-semantică a limbii române vechi
DeLORo (Deep Learning for Old Romanian)
Cod proiect: PN-III-P2-2.1-PED-2019-3952**

**RAPORT ȘTIINȚIFIC ȘI TEHNIC
Etapa intermedieră II
1 ianuarie 2021 - 31 decembrie 2021**

I. Rezumatul etapei

În a doua etapă a proiectului au fost obținute următoarele rezultate:

- S-a continuat activitatea de identificare de resurse și de documente chirilice transcrise în caractere latine.
- După instalarea și configurarea serverului obținut prin finanțarea din proiect, s-a continuat procesul de dezvoltare a bazei de date de documente primare, precum și a standardului colecției - ROCC.
- S-a continuat îmbunătățirea interfeței OOCIAT de editare metadate și adnotare a imaginilor.
- S-a început inventarierea cuvintelor care apar în formă flexionată în transcrierile în alfabet latin în resursele deținute și s-au făcut experimente cu modele de clusterizare a formelor flexionate.
- S-au făcut primele experimente de identificare a obiectelor în imagini și de recunoaștere a caracterelor chirilice.
- S-a propus și implementat un model de aliniere între imagine și textul transcris.
- S-a experimentat modele de prelucrări semantice care să pună în evidență diferențe diacronice și sincrone în lexicul limbii române. Modelele asumate vor trebui comparate cu considerațiile făcute de experți lingviști, elaborate acum.
- Proiectul și realizările lui de până acum au fost diseminate în mai multe evenimente științifice, o parte din ele organizate de membrii proiectului.
- Site-ul proiectului a fost încărcat complet cu date.

II. Descrierea științifică și tehnică

(punerea în evidență a rezultatelor etapei și gradul de realizarea obiectivelor - se vor indica rezultatele și modul de diseminare a rezultatelor)

Act. 2.1 - Dezvoltarea corpusului - Tranșa II (partener: UAIC)

A. Continuarea identificării de resurse (listă de titluri, deținători, drepturi de autor) ce vor fi utilizate drept date de antrenare și validare în această etapă și în cea următoare.

În această etapă au fost identificate noi titluri de documente vechi românești din secolele XVI - XIX, redactate în alfabetul chirilic (în general, cărți tipărite și/sau copiate de mână prin reproducerea caracterelor de tipar - semiunciale), aflate în posesia unor mari biblioteci, pentru care există copii electronice obținute prin scanarea paginilor⁹. Lista acestor noi titluri este concatenată celei raportată în Etapa 1¹⁰. Ne-am concentrat, de asemenea, pe identificarea unor documente din aceeași perioadă pentru care există transcrierea conținutului în caractere ale alfabetului latin, care să fie folosite într-un proces de accelerare a procesului de constituire a colecției de antrenare a tehnologiei de recunoaștere a caracterelor chirilice în context. Lista acestor titluri este dată în Tabelul 1.

Tabelul 1: Lista surselor transcrise identificate

1. Iulia Mazilu Bucataru (editor): *Şapte taine* (Iasi, 1644)
2. Madalina Ungureanu: Regi 3-4, B1688 etc.
3. Roxana Vieru (editor): *Psaltirea Hurmuzaki* (Maramureş, 1500/1520) - CETRV
4. Roxana Vieru (editor): *Întrebare creştinească* (*Catehism*, Coresi) (Braşov, 1559/1564) - CETRV
5. Ana Sabie, Roxana Vieru (editori): *Apostol*, Coresi (Braşov, 1563) - CETRV
6. Bogdan Țâra (editor): *Cazania I* (Braşov, 1567-1568) - CETRV
7. Adina Chirilă (editor): *Cazania II* (Braşov, 1581) - CETRV
8. Ana Sabie, Roxana Vieru (editori): *Psaltirea Voronețeană* (Maramureş, 1551/1558) - CETRV
9. Alexandru Gafton (editor): *Codicele Bratul* (1559/1560) - CETRV
10. Ioana Ciobanu, Roxana Vieru (editori): *Tetraevanghel* (Braşov, 1561) - CETRV
11. Alexandru Gafton (editor): *Codicele Voronețean* (Maramureş, 1563/1583) - CETRV
12. Bogdan Țâra (editor): *Liturghier*, Coresi (Braşov, 1570) - CETRV
13. Alexandru Gafton (editor): *Psaltirea Scheiană* (Maramureş, 1573-1578) - CETRV
14. Gheorghe Chivu (editor): *Codex Sturdzananus* (Hunedoara - Măhaci, 1580-1619) - CETRV
15. Roxana Vieru (editor): *Floarea darurilor* (Putna, 1592/1604) - CETRV
16. Ioana Ciobanu, Roxana Vieru (editori): *Pravilă* (Govora, 1640) - CETRV
17. Roxana Vieru (editor): *Cronograful lui Moxa* (Mănăstirea Bistrița, 1618>) - CETRV
18. Gheorghe Chivu (editor): *Apocriful Iorga* (prima jumătate a secolului al XVII-lea) - CETRV
19. Bogdan Țâra (editor): *Gromovnic* (jud. Sibiu, 1636) - CETRV
20. Bogdan Țâra (editor): *Leastvița* (Mănăstirea Secu, deceniile 2-4 ale secolului al XVII-lea) - CETRV

⁹ Pentru că proiectul DeLORo nu include o activitate de scanare a originalelor, ne-am concentrat atenția asupra unor surse aflate în posesia unor mari biblioteci care dețin și copiile scanate ale paginilor.

¹⁰ Aflată la adresa:

https://docs.google.com/spreadsheets/d/1hv_jKwLzK8ZFiVsO8IA0L2I7nvXSAT7QOUdgxeJCIcs/edit?usp=sharing

21. Roxana Vieru, Adina Chirilă (editori): *Apostol* (Moldova, 1646)
22. Roxana Vieru, Adina Chirilă (editori): *Evanghelie* (Snagov, 1697)
23. Mihaela Onofrei (editor): *De obște gheografie* (Iași, 1795)
24. Andreea Drîscu (editor): *Gramatica lui Radu Tempea* (Brașov, 1797)
25. Mioara Dragomir (editor): *Hronograf den începutul lumii* (Ms.3517)

B. Continuarea procesului de dezvoltare a corpusului conform listei de resurse identificate (scanuri de pagini) prin depunerea lor în site-ul proiectului.

1. Dezvoltarea corpusului de documente primare

S-a continuat acumularea de documente primare în ROCC. În lista rezultată¹¹ fiecare titlu reprezintă o colecție de imagini de pagină. Toate aceste documente se află actualmente accesibile pentru adnotare prin interfața OOCIAT.

2. Dezvoltarea în continuare a standardului ROCC

Odată cu achiziționarea de date în corpusul ROCC (a se vedea RST-I), s-a continuat completarea schemei de structură a corpusului cu informații asupra alinierilor imagine - text. Această parte este descrisă mai jos.

<ROCC>
<pageCollection>

Notă: Următoarea secțiune descrie textele transcrise în alfabet latin ale documentului original chirilic, în integralitatea lor, precum și segmentate pe pagini.

<textsOfPages> (optional, în cazul în care documentul are atașată transcrierea lui), cu un argument și un set de pagini de text; numărul de elemente <textPage> este egal cu numărul de pagini care conțin text în document:

@integralTrinscribedTextFile (obligatoriu¹²): URL-ul fișierului care conține textul transcris integral al documentului.

<textPage> (obligatoriu): secvența de pagini segmentate;

@pageID (obligatoriu): ID-ul imaginii de pagină corespunzătoare din cadrul colecției, identificată prin <pageCollection> ⇒ <imagesOfPages> ⇒ <onePageImage> ⇒ @pageID;

@textPageFile (optional): URL-ul fișierului care păstrează textul transcris al paginii, în cazul în care se optează să păstrăm și textele de pagină separat; acest atribut este la concurență cu perechea @beginningOfPage, @endOfPage;

¹¹ La adresa https://docs.google.com/spreadsheets/d/11q60tE-QwYonQ_N35sg6Xk1rnKBC9Ku/edit#gid=1173026119

¹² Un element sau un atribut marcat “obligatoriu” sub un element “optional” are semnificația că elementul sau atributul trebuie să apară dacă există elementul superior.

@beginningOfPage (optional): când nu există fișier de text atașat paginii, offset la primul caracter al paginii din fișierul <textsOfPages> ⇒ @integralTrinscribedTextFile;

@endOfPage (optional, în pereche cu @beginningOfText): offset la ultimul caracter al paginii din fișierul <textsOfPages> ⇒ @integralTrinscribedTextFile;

```
</textPage>  
</textsOfPages>
```

Notă: Secțiunea care urmează este dedicată alinierilor făcute manual între obiecte grafice de tip slovă chirilică și caractere din textul latin. Nu numim aliniere și nu sunt cuprinse aici transcrierile în alfabet latin făcute de adnotatori asupra conținuturilor unor obiecte identificate prin operarea interfeței OOCIAT. Aceleia sunt incluse în descrierile de obiecte, în atributele @objectContent.

<characterAlignmentsImage2Text>

<onePageAlignments> (obligatoriu): toată această secțiune se completează automat de către programul de aliniere;

@pageID (obligatoriu): ID-ul de pagină în cadrul colecției, aici:

<pageCollection> ⇒ <imagesOfPages> ⇒ <onePageImage> ⇒ @pageID;

<charAlignment> (o înregistrare de acest tip se generează la fiecare aliniere efectuată manual sau automat);

@objectId (obligatoriu): ID-ul unui obiect tip <object:Character> din secțiunea <pageCollection> ⇒ <segmentationOfImages> ⇒ <pageSegmentation> ⇒ <object:Character> ⇒ objectId;

@characterOffset (obligatoriu): un offset de caracter relativ la începutul fișierului care conține pagina transcrisă, identificat de <pageCollection> ⇒ <textsOfPages> ⇒ @textPageFile;

@goldTestAlignment (obligatoriu): cu valorile: “gold” = alinierea a fost făcută manual ori de către mașină, validată apoi manual; “test” = aliniere generată automat, nevalidată;

@length (obligatoriu): indică numărul de caractere al transcrierii latine corespunzătoarea slovei chirilice, cu valorile: “0” = slova nu are un corespondent în transliterare (în acest caz, @characterOffset indică încă poziția următorului caracter din textul latin; “1” = slovei îi corespunde un caracter latin; “2” = slovei îi corespund două caractere latine).

</charAlignment>

Notă: Exemple de corespondențe slovă-la-literă:



⇒ VOID (length = "0")



⇒ f (length = "1")



⇒ st, ГЕНДАДІЕ ⇒ gh, ca în Ghenadie (length = "2")

</onePageAlignments>

</characterAlignmentsImage2Text>

Notă: Mai jos prezentăm o altă variantă pentru secțiunea <characterAlignmentsImage2Text>. În această variantă, sunt aliniate secvențe de slove grafice cu secvențe de caractere transcrise. Algoritmul de aliniere ar trebui să alinieze numai slove-la-litere care au probabilitățile de recunoaștere foarte mari (aproape sigure). Pentru ca alinierea să se facă cât mai corect, ar trebui să se caute în ROCC perechi slove-la-litere cu acuratețea de recunoaștere aproape de certitudine, aflate la distanță relativ mică (atât în pagina de slove cât și în fișierul cu transcrierea), între care există același număr (sau unul foarte apropiat) de slove, respectiv litere. Distanța “mică” ar trebui să se păstreze pe parcursul antrenării iterative a aliniatorului, în etapele târzii ale antrenării perechile de slove, respectiv litere, devenind tot mai dese (din cauza aglomerării cu sloverecognoscibile). De remarcat, de asemenea, că în această variantă slovele recunoscute sunt, în etapele inițiale ale antrenării iterative, mai întâi separate, ulterior ele devenind grupuri, adică secvențe alăturate. O descriere a algoritmului de aliniere se află în (Cristea et al., 2021-în curs de apariție).

<seqAlignmentsImage2Text>

<onePageAlignments> (obligatoriu): toată această secțiune se completează automat de către interfața de editare OOCIAT și de programul de aliniere;

@pageID (obligatoriu): ID-ul de pagină în cadrul colecției, aici:

<pageCollection> ⇒ <imagesOfPages> ⇒ <onePageImage> ⇒ @pageID;

<seqAlignment> (o înregistrare de acest tip se generează la fiecare aliniere dintre o secvență de slove grafice și un sir de caractere adiacente din text, efectuată manual sau automat);

@seqCharIDs (obligatoriu): o secvență de ID-uri de obiecte

<object:Character> din secțiunea <pageCollection> ⇒ <segmentationOfImages> ⇒

<pageSegmentation> ⇒ <object:Character> ⇒ objectID;

@seqCharLength (obligatoriu): lungimea secvenței de slove;

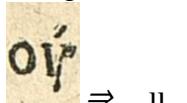
@seqCharOffset (obligatoriu): offsetul primului caracter din secvența transcrită, relativ la începutul fișierului paginii de text, identificat de <pageCollection> ⇒ <textsOfPages> ⇒ @textPageFile; Dacă atributul @textPageFile din <pageCollection> ⇒ <textsOfPages> lipsește (în varianta în care un unic fișier txt memorează textul transcrit al întregii colecții de pagini), atunci acest atribut indică un offset în fișierul <pageCollection> ⇒ <textsOfPages> ⇒ @integralTranscribedTextFile;

@seqTransCharLength (obligatoriu): lungimea secvenței de caractere transcrise;

@goldTestAlignment (obligatoriu): cu valorile: “gold” = alinierea a fost făcută manual ori de către mașină, validată apoi manual; “test” = aliniere generată automat, nevalidată;

</seqAlignment>

Notă: De semnalat că secvențe de 2 slove care se transcriu printr-o singură literă latină, ca de exemplu:



pot fi descrise la acest nivel de adnotare.

```

</onePageAlignments>
</seqAlignmentsImage2Text>
</pageCollection>
</ROCC>

```

C. Modificări în interfețele de adnotare a resurselor (obiecte identificabile și conținutul lor lexical) și de completare metadate.

1. Continuarea dezvoltării interfeței OOCIAT (*Online Old Cyrillic Annotation Tool*)

Din punctul de vedere al funcționalității interfeței putem distinge 4 module diferite:

- Login-ul

Această funcționalitate permite accesul în interfață pe baza unui *username* și a unei parole. Accesul este permis doar persoanelor care dețin astfel de credențiale. *Username*-ul este de asemenea util pentru a înregistra autorul adnotărilor.

- Editarea metadatelor

Pagina respectivă prezintă un formular unde, selectând un anume document din lista existentă, se pot realiza modificări asupra metadatelor corespunzătoare. Metadatele se referă la momentul elaborării operei, tipul de scris, limba în care a fost realizată lucrarea etc.

- Statistici

Modulul de statistică (Fig. 1) prezintă situația la zi a obiectelor adnotate, cumulat și pe adnotator.

Statistică per tip de obiect		
Adnotator:		
<input type="text" value="Cristian Padurariu"/>		▼
<input type="button" value="Încarcă statistici"/>		
Tip adnotare	Total instanțe adnotate	Instanțe adnotate de Cristian Padurariu
Acolade	0	0
Coloane	7	3
Chenare	0	0
Frontispicții	33	3
Litere ornate	398	0
Litere	146557	66746
Linii	6980	3455
Manșete	48	1
Modificatori	5490	4
Ornamente	78	0
Litere în afara linilor	1342	0
Referințe deasupra linilor	0	0
Referințe pe manșetă	0	0
Titluri	158	6

Figura 1: Afisarea statisticilor în OOCIAT

● Adnotare

Adnotarea se realizează în interiorul spațiului de lucru, descris în această secțiune. Vizualizarea parțială sau totală a spațiului de lucru se realizează cu ajutorul comenzi **ZOOM**. Interfața conține opțiunea de **Luminozitate** care-i permite utilizatorului să-și regleze intensitatea luminoasă a paginii scanate după dorință.

OOCIAT permite mai multe tipuri de operații care vizează: declararea calității paginii ce urmează să fie supusă procesului de adnotare, operații de marcarea a zonelor în care se află diverse obiecte în pagină și, pentru anumite tipuri de obiecte, transliterarea conținutului din scrierea în grafia chirilică în cea cu grafie latină. Pașii unei sesiuni de lucru sunt următorii:

1. Se deschide aplicația **OOCIAT**¹³, după obținerea datelor de logare (**ID** și **Parolă**).
2. Pentru a avea acces la reperitoriu de date se va alege opțiunea **Continuă adnotare document**. Se selectează resursa lingvistică (opera) care se dorește a fi adnotată și se apasă butonul **Încarcă imaginile** (v. Fig. 2).
3. Se navighează folosind butoanele << și >> către pagina care se dorește a fi adnotată sau se poate specifica numărul paginii la comanda **Mergi la pagina**.

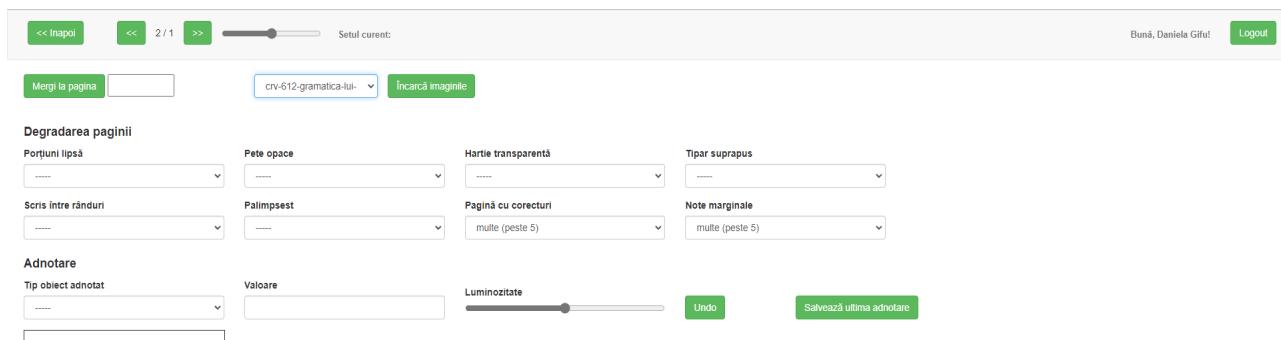


Figura 2: Interfața OOCIAT

4. Adnotatorul va examina imaginea și va completa câmpurile referitoare la gradul de degradare a paginii (**Degradarea paginii**) care conține următoarele metadate: **Porțiuni lipsă** (cu opțiunile **Adevărat** și **Fals**); **Pete opace** (cu opțiunile **Adevărat** și **Fals**); **Hartie transparentă** (cu opțiunile **Adevărat** și **Fals**); **Tipar suprapus** (cu opțiunile **Adevărat** și **Fals**); **Scris printre rânduri** (cu opțiunile **Adevărat** și **Fals**); **Palimpsest** (cu opțiunile **Adevărat** și **Fals**); **Pagină cu corectură** (cu opțiunile **Niciuna** și **Puține (între 1 și 5)**); **Note marginale** (cu opțiunile **Niciuna** și **Puține (între 1 și 5)**).
5. Pentru a începe adnotarea propriu-zisă (v. Fig. 3), se selectează **Tip obiect adnotat** (în care regăsim opțiunile: **Obiecte grafice - Frontispiciu și Ornament**; **Obiecte text primare - Titlu, Coloană, Rând, Caracter**; **Obiecte text secundare - Modificator, Text marginal, Text interliniar, Majusculă, Trimitere manșetă, Trimitere interliniară**). În funcție de tipul de obiect dorit a fi adnotat, se marchează cu mouse-ul dreptunghiul care conține obiectul, Pentru obiectele care au atașat un conținut textual se introduce în fereastra

¹³ La adresa [Image annotator login \(academiaromana-is.ro\)](http://academiaromana-is.ro).

Valoare transliterarea textului citit în imagine și se apasă butonul **Salvează ultima adnotare**. Butonul **Undo** anulează ultima comandă.

6. Adnotările realizate deja se vor afișa în dreapta imaginii într-un chenar gri.
7. Prin executarea unui click pe oricare chenar se va elimina adnotarea care a fost făcută incorrect.

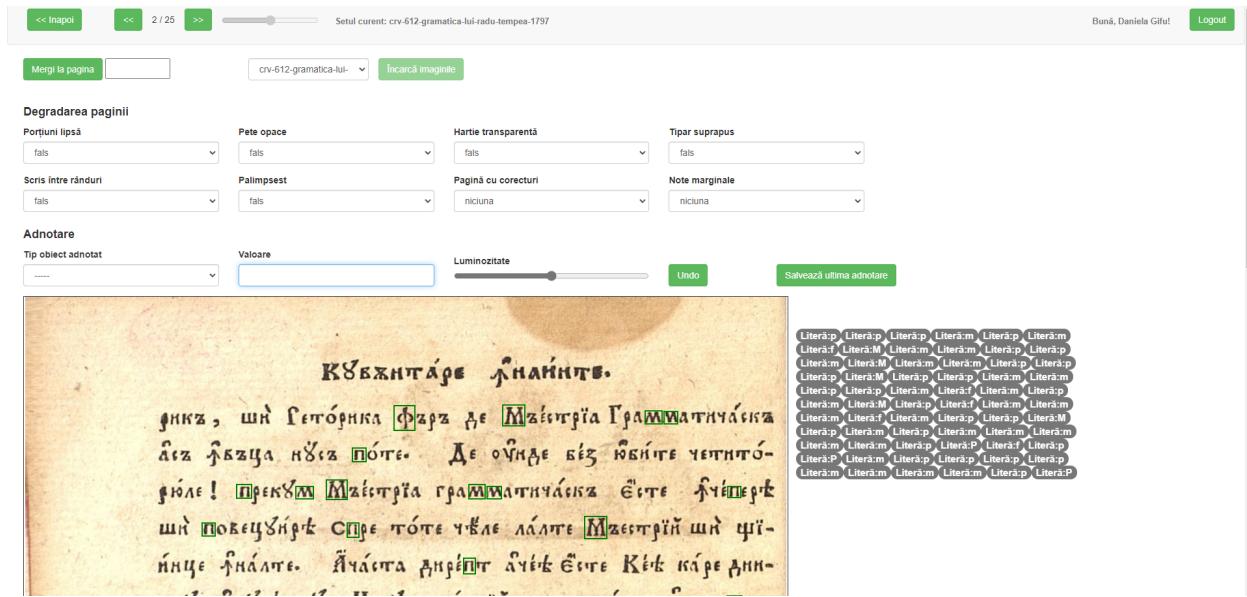


Figura 3. Sesiune de lucru cu aplicația OOCIAT: adnotare de litere dintr-un anumit set, pe sărite

2. Corectarea erorilor

Frontend-ul OOCIAT a fost perfecționat la fiecare raportare de funcționare eronată. Un tabel plasat în spațiul de lucru Google Drive al proiectului¹⁴ a fost pus la dispoziția membrilor proiectului pentru a raporta erorile detectate, rezolvate sau în curs de rezolvare, în funcționarea interfeței, pe parcursul etapelor I și II. O eroare a cărei sursă a fost greu detectabilă s-a referit la aparenta dispație a adnotărilor la nivel de rând (și câteva alte tipuri de obiecte secundare), ceea ce a descurajat un timp activitatea de adnotare, efectuată cu multă dedicație de către membrii lingviști ai proiectului. A mai fost raportată și se mai menține încă și eroarea care face să dispară adnotările privind degradarea paginii. De asemenea, pentru a simplifica operarea, prin proiectare s-a decis ca ștergerea voită a unui obiect deja adnotat să se facă printr-un singur click pe înregistrarea corespunzătoare lui (care apare în partea dreaptă a ecranului). Acest lucru însă mai înseamnă și că nu se pot face corecturi la nivel de literă în transcrierile deja făcute de colaboratori în obiecte de tip rânduri (din fericire, astfel de corecții nu au trebuit fi făcute până în prezent), dar în viitor va

¹⁴ La adresa:

https://docs.google.com/spreadsheets/d/1_bcxaN1rsa0JfqQoYnw19hPGX5y0yat4V7rMN8WQsnE/edit#gid=0

trebuie modificată funcționalitatea pentru a permite și efectuarea de corecturi în conținutul unui obiect, fără ștergerea și rescrierea obiectului în întregime.

D. Continuarea procesului de adnotare a resurselor (obiecte și conținut lexical)

Într-o primă fază din cadrul acestei etape, au fost transpusă metadatele documentelor primare primite de la BAR, în formatul convenit în cadrul proiectului DeLORO. Pentru acest proces s-a realizat un script care preia informațiile din fișierele XML aferente datelor achiziționate de la BAR și le transpune într-o bază de date relațională¹⁵.

Până în prezent baza de date cumulează următoarele:

- 55 de cărți ce includ 18.786 de pagini scanate, împărțite pe perioade după cum urmează:
 - XVI-1: 1500–1549 - **1 carte** (1 x uncial) - **248 pagini**
 - XVI-2: 1550–1599 - **12 cărți** (6 x tipar + 6 x uncial) - **4647 pagini**
 - XVII-1: 1600 - 1649 - **11 cărți** (5 x tipar + 6 x uncial) - **3808 pagini**
 - XVII-2: 1650 - 1699 - **4 cărți** (4 x tipar) - **2530 pagini**
 - XVIII-1: 1700 - 1749 - **11 cărți** (11 x tipar) - **3924 pagini**
 - XVIII-2: 1750 - 1799 - **9 cărți** (9 x tipar) - **2062 pagini**
 - XIX-1: 1800 - 1849 - **7 cărți** (7 x tipar) - **1567 pagini**
 - XIX-2: 1850 - 1899 - **0 cărți**
- s-au adnotat
 - **293.636 linii**
 - **8.514.008 caractere** din care cele mai frecvente sunt: 854.168 pentru litera “e”; 777.241 pentru litera “i”; 617.714 pentru litera “u”; 579.788 pentru litera “a”; 523.666 pentru litera “r”; 468.246 pentru litera “n” etc.

Act. 2.2 - Construirea de modele diacronice ale limbii - I (parteneri: UAIC, UB)

Activitatea are ca obiectiv colectarea și inventarierea cuvintelor care apar în formă flexionată în transcrierile în alfabet latin în resursele deținute.

Inventarul a fost actualizat la fiecare creștere masivă a repertoriului de resurse, o dată la câteva luni. Prezentăm, succint, modul în care a fost întocmit acest lexic pe baza textelor transcrise pe care consorțiul le-a dobândit din varii surse, precum și prin concatenarea liniilor adnotate la conținut prin interfața OOCIAT. Aceste texte au fost procesate de către un modul al bibliotecii spacy¹⁶, antrenat pe limba română modernă¹⁷. După eliminarea semnelor de punctuație, s-a aplicat o operație de tokenizare. Au fost eliminate termenii (*tokens*) ce conțin cifre și cei de la sfârșit și început de rând despărțiti prin cratimă. Un termen este numărat o singură dată pe o perioadă de 50 de ani. Dăm mai jos numărul de termeni identificați în lexicul fiecărei perioade: 1550-1600 => 2021; 1700-1750 => 4099; 1750-1800 => 802; 1800-1850 => 1244. Pentru perioadele 1600-1650 și 1650-1700 nu există date.

¹⁵ <https://github.com/deloro-project/rocc-schema>

¹⁶ <https://spacy.io>

¹⁷ https://spacy.io/models/ro#ro_core_news_lg

Din cauza incompletitudinii lor, nu considerăm relevante datele de natură lexicală acumulate până în acest moment. O atenție deosebită va fi acordată acestui aspect în etapa următoare, când vom dispune de mult mai multe documente procesate¹⁸. Remarcăm totuși că din datele colectate până la momentul scrierii acestui raport s-au putut identifica 918 termeni care apar în cel puțin două perioade.

Procesul de obținere a vocabularelor pe perioade și calculul statisticilor asupra acestora este automatizat într-un script Python¹⁹.

Act. 2.3 - Dezvoltarea modelelor de identificare a obiectelor - I

A. implementarea de metode statistice și neuronale pentru identificarea obiectelor relevante din imaginile de pagini scanate (parteneri: UAIC, UB)

Pentru identificarea obiectelor din imagini s-a folosit un algoritm de tip *object detection* (F-RCNN). Acesta primește un input de forma x_{min} , x_{max} , y_{min} , y_{max} , ce reprezintă coordonatele dreptunghiului care conțin obiectul ce se dorește a fi identificat. Datorită formei alungite pe care o are în general obiectul de tip rând comparativ cu obiectele de tip caracter care sunt mici și aproximativ pătrate, am hotărât să folosim două astfel de modele de recunoaștere, unul pentru detectarea liniilor și altul pentru restul caracterelor. Pentru a antrena și testa cele două modele s-au folosit seturi de date obținute prin adnotare manuală. Astfel, după împărțirea seturilor după tiparul 80/20 s-a realizat antrenarea și testarea modelelor. Metrica de evaluare a fost mAP(%), care a întors un rezultat de 81% pentru detectarea liniilor, aproximativ 40% pentru cele mai frecvente 40 de caractere și 20% pentru cele mai rare 40 de caractere.

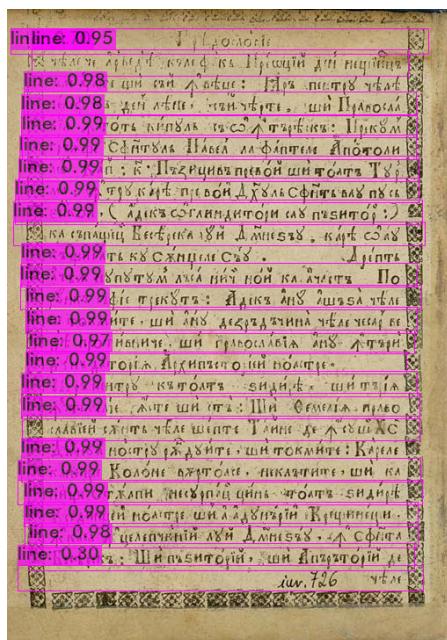


Figura 4: YOLO v4 specializat pe detecția de lini.

În cadrul echipei UB, am elaborat un model de identificare a liniilor bazat pe arhitectura [YOLO v4](#) - axată pe detecția de obiecte în imagini, cu care am obținut rezultate de nivel state-of-the-art în acest domeniu. Folosind drept etichete coordonatele chenarelor (adnotate în OOCIAT), am specializat YOLO v4 pe detecția chenarelor ce încadrează liniile. În acest scop, am împărțit setul de date într-un set de antrenare, format din 332 de pagini, conținând 5.715 rânduri adnotate, și un set de test cu 35 de pagini, având 492 de chenare trasate. Setul de antrenare și cel de test au fost selectate în aşa fel încât să conțină pagini din opere literare diferite. Astfel asigurăm condiții de evaluare mai apropiate de realitate și eliminăm elementul de “bias” dat de stiluri de

¹⁸ Pentru construcția lematizatorului și a modelelor de limbă, va putea fi folosită (deocamdată în regim privat, pentru cercetare, cu citarea autorilor) și baza de date “Corpus electronic al textelor românești vechi”.

¹⁹ Codul-sursă este disponibil la adresa <https://github.com/deloro-project/rocc-pipelines>.

scriere potențial similară în setul de antrenare, învățate de model, cu aceleia folosite pentru testare. Fig. 4 arată un exemplu de linii detectate de model într-una dintre paginile folosite pentru evaluare. Pe acest exemplu, se observă o încadrare bună a liniilor, modelul antrenat detectându-le, în general, cu o acuratețe de peste 95%. Acest model a fost evaluat pe setul de test, cu un *mean average precision* (mAP)@0.50 de 76.33%, utilizând drept *ground truth* chenarele de test așa cum au fost ele furnizate inițial. Inspectând chenarele desenate de model, deducem faptul că această metrică este, în realitate, mai mare de 76%, pentru că există pagini în setul de antrenare ce au o singură linie adnotată sau un număr de linii adnotate mai mic decât numărul real de linii. Din Fig. 5 se poate observa faptul că etichetele sănătoase din pagina din stânga marchează un singur rând, pe când modelul antrenat de noi în acest scop, detectează toate liniile de interes. Un prim pas, pentru a obține o valoare cât mai corectă a metricii de evaluare, va fi să retrasăm chenarele în datele folosite pentru testare, cu o verificare manuală și corectare a acestora. Ulterior, planuim utilizarea unei învățări de tip *self-paced*, utilizând modelul YOLO v4. Acest tip de învățare presupune împărțirea datelor de antrenare în câteva subdiviziuni (de exemplu 5), prima subdiviziune conținând paginile cu cele mai multe chenare trasate. Modelul va fi astfel antrenat în reprise, începând cu subdiviziunea cu cele mai multe chenare marcate pe pagină, cu adnotarea automată a liniilor lipsă din fiecare *fold* de antrenare. Considerăm că această tehnică de antrenare va duce la o creștere semnificativă a acurateții modelului.

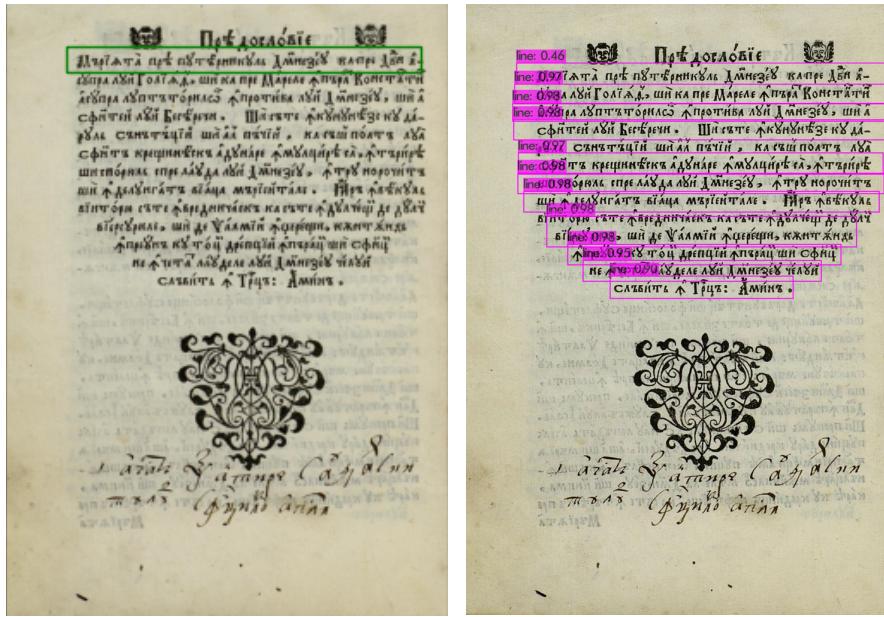


Figura 5: În stânga: imagine (pagină) din setul de test, cu chenarele trasate conform coordonatelor *ground truth*. În dreapta: aceeași pagină, cu chenarele trasate folosind modelul de detecție de linii bazat pe arhitectura YOLO v4.

B. Elaborarea unui model și a unei aplicații de aliniere a obiectelor identificate în imagini cu transcrierile lor textuale (partener: UAIC)

Un model de aliniere a obiectelor de tip caracter identificate în imaginile de pagini cu textul transcris a fost descris în secțiunea 4.2 din lucrarea (Cristea et al., 2021-submitted), trimisă la ConsILR-2021, actualmente în curs de recenzare. Algoritmul a fost elaborat în vederea scurt-

circuitării procesului foarte consumator de timp și obositor de adnotare manuală, care să susțină în siguranță pregătirea tehnologiei CR, presupunând că tehnologia mai ușoară a OI funcționează satisfăcător. În schimb, exploatăm un set de documente deja transcrise de experți, încercând astfel să scurtăm procesul de achiziție a datelor. S-a trecut apoi la implementarea acestui model. Actualmente suntem în faza în care algoritmul de aliniere este implementat și urmează să fie cuplat cu algoritmul de identificare a obiectelor de tip linie și a celor de tip caracter, la care se adaugă recunoașterea caracterelor.

Detaliem, mai jos, acest proces. În primul rând, introducem câteva notații. Fie *IMAGE* o pagină scanată (caracterizată printr-un ID de pagină în spațiul ROCC) a unui document în limba română redactat în alfabet chirilic, pentru care baza de date include o transcriere în latină. El conține un set de cadre dreptunghiulare de obiecte de tip caracter, cu laturile paralele cu axele orizontală și verticală, fiecare obiect fiind caracterizat prin: un ID unic, coordonatele (orizontale și verticale) a două puncte (stânga-sus, dreapta-jos) și o etichetă (care poate fi VOID, adică nerecunoscut, sau o literă latină). Cadrele acestor obiecte (numite „containere de caractere”) au fost identificate automat prin tehnologia OI, în timp ce conținutul lor este rezultatul aplicării tehnologiei CR.

Fie *TEXT* un text codificat ASCII de o pagină, reprezentând transcrierea interpretativă a lui *IMAGE*; caracterele din *TEXT* sunt sortate crescător după adresele byte ale lor (*offset*). Textele paginilor sunt delimitate de marcaje speciale <p> plasate între ele. Ca atare, spațiul de aliniere este un set de perechi *IMAGE-TEXT*.

Strategia de scurt-circuitare

Această strategie este aplicată iterativ pentru a efectua cicluri de adnotare-antrenare-evaluare-aliniere, cu scopul de a obține mai multe date de antrenament decât cele adnotate direct de experți, prin exploatarea textelor existente transcrise manual. Poate fi aplicat în diferite etape de-a lungul procesului dureros și plăcitor de adnotare manuală a datelor (prin interfață special concepută OOCIAT). Fiecare etapă de antrenare este urmată de o evaluare a acurateței recunoașterii tuturor obiectelor de tip <object:Character>.

Din experiența raportată până acum (Ren et al., 2015), se poate stabili o scară crescătoare de cantități de instanțe necesare într-un proces de instruire. Valorile din scară trebuie plasate logaritmice, pentru că diferite tipuri de obiecte au nevoie de cantități de instanțe de antrenament care conduc la același scor de acuratețe a recunoașterii din ce în ce mai mari. De exemplu, tipul <object:Character> poate avea scara: 1.000, 5.000, 10.000, 20.000 etc.). Apoi, se stabilește un prag pentru acuratețea recunoașterii unui caracter, să spunem θ (de exemplu, 97%), peste care caracterele recunoscute sunt considerate a fi „sigure”.

Inițializare:

1. Se construiește structura de date *LatinChars*, ca un set de obiecte, de cardinal egal cu setul de caractere din alfabetul Latin, fiecare instanță având structura: {char *C*, list *imCSet*, integer *count*, float *acc*, integer *safeAnnoLevel*}, unde: *C* este un caracter latin, *imCSet* este lista tuturor instanțelor de imagine reprezentând caracterul *C* din setul de antrenament (stocate ca ID-uri ale obiectelor de tip <object:Character>), *count* este cardinalul *imCSet*, adică setul de antrenament

pentru recunoașterea lui C , $safeAnnoLevel$ reprezintă următoarea valoare pe scara cantităților de instanțe de antrenament care ar trebui atinse pentru a obține pragul θ , iar acc va fi actualizat cu acuratețea recunoașterii caracterului C , măsurată după fiecare iterație de antrenament (în intervalul: 0 - 100%); inițial, aceste valori sunt lăsate necompletate.

2. Se propune o funcție de scor care să aprecieze încrederea în alinierea unui container din $IMAGE$ cu un caracter din $TEXT$ și un algoritm care propune un set de $ALIGNMENTS$ de forma $<c, container-ID, off-char, s>$, unde: c este o valoare de caracter, $container-ID$ este ID-ul unui container din $IMAGE$ a cărui valoare este c (diferită de VOID), $off-char$ este offset-ul unui caracter c în $TEXT$ și s este valoarea funcției de scor pentru această aliniere.

3. Pentru a efectua antrenamentul, calibrarea și evaluarea, se împart datele inițiale de tipul $<object:Character>$ din ROCC în două seturi disjunse, $Gold$ și $Test$, într-un anumit raport de distribuție (de exemplu 90%, 10%).

4. Se antrenează inițial rețeaua neuronală CR pe datele din $Gold$, apoi se evaluatează și se setează valorile acc ale structurii $LatinChars$, pentru a exprima acuratețea medie a fiecărui caracter latin.

După cum se va vedea, rețeaua neuronală CR va fi antrenată periodic pe tot mai multe instanțe ale setului $LatinChars$, și anume de fiecare dată ce numărul de obiecte din $Gold$ de un anumit tip, acumulat prin adnotare manuală și/sau rezultat prin aplicarea strategiei de scurt-circuitare, depășește următorul $safeAnnoLevel$ pentru acel caracter. Prin urmare, nivelul de precizie acc , atins pentru fiecare element din setul $LatinChars$, este actualizat după fiecare sesiune de antrenament + evaluare.

O apariție în $IMAGE$ a unei instanțe a clasei $<object:Character>$ poate fi clasificată în două moduri: ca un container de caracter cu conținut necunoscut, să-l numim $<CharacterVoid>$, dacă rețeaua neuronală raportează o încredere de recunoaștere pentru acea instanță mai mică decât un prag (în acest caz, atributul $@objectContent$ este setat la VOID) sau ca un container care poartă un conținut etichetat, să numim această instanță $<CharacterX>$, unde X este un literă latină care codifică caracterul (în acest caz, atributul $@objectContent$ este setat la X).

5. Pentru fiecare pereche $IMAGE-TEXT$ din spațiul de aliniere, se procedează astfel:

Liniarizare:

6. Se sortează containerele găsite în $IMAGE$ de la stânga la dreapta, folosind centrele de greutate ale formelor lor, aproximativ aliniate pe linii orizontale, apoi se concatenează liniile consecutive pentru a produce un singur sir de ID-uri de caractere.

7. Se livrează sirul obținut prin transcrierea în ordine a etichetelor containerelor și înlocuirea etichetei VOID cu „\$”. Când distanța orizontală dintre două container este mai mare decât un prag determinat experimental, se propune, cu un anumit grad de încredere, generarea unui spațiu.

Aliniere:

8. Se caută în $IMAGE$ o pereche de containere, $<CharacterX>$ și $<CharacterY>$, ambele cu $acc > \theta$, așa cum sunt raportate de elementele $LatinChars$ corespunzătoare, între care se află d (puține, de exemplu, între 1 și 10) alte container intermediare din tipul $<CharacterVoid>$, adică al căror $acc \leq \theta$ (spațiile nu sunt numărate). Această aliniere ar trebui să producă un 10-uplu $<pI, X, imX, offIX, offTX, pT, Y, imY, offIY, offTY>$, unde pI este ID-ul unei pagini $IMAGE$ în care a fost

descoperită secvența, pT - cea a paginii corespunzătoare $TEXT$, X și Y sunt cele două caractere, imX și imY sunt ID-urile containerelor $<CaracterX>$ și $<CaracterY>$ din pagina $IMAGE pI$, $offIX$ și $offIY$ sunt offset-urile imX și imY în aceeași pagină $IMAGE$, $offTX$ și $offTY$ sunt decalajele celor două caractere din pagina $TEXT pT$, $|offY-offX| = d+1$, între cele două caractere X și Y din pagina $TEXT$ există același număr d de litere interpuse, iar $offIX$ și $offIY$ sunt aproximativ egale cu $offTX$, respectiv $offTY$.

9. Se aliniază toate caracterele din $IMAGE$ plasate între imX și imY , unu la unu, cu caracterele din $TEXT$ corespunzătoare plasate între X și Y , ceea ce înseamnă:

- fiecare aliniere imagine-caracter este o pereche $<imC, C>$;
 - pentru fiecare caracter aliniat C din structura $LatinChars$, se incrementează $count$ și se adăugă imC la setlist-ul $imCSet$;
10. Se verifică dacă în setul $LatinChars$ apar caractere noi ale căror contoare $count$ depășesc $safeAnnoLevel$ (acesta este următorul prag al numărului de instanțe pe scara nivelurilor de antrenament, aşa cum este descris la pasul 1, care ar trebui atins pentru a începe un nou proces de instruire). Dacă da:

- se reia antrenamentul pentru perechile $<imC, C>$ cu seturile lor de antrenament $imCSet$ actualizate, pentru că numărul lor depășesc acum pragurile $safeAnnoLevel$;
- pentru fiecare caracter C nou antrenat, se actualizează:
 - acc în funcție de valoarea rezultată din evaluarea acurateței asupra noilor seturi $Gold$;
 - $safeAnnoLevel$ la următorul prag din lista de la pasul 1.

11. Dacă toate caracterele din setul $LatinChars$ au atins pragurile maxime de antrenament ($safeAnnoLevel$), atunci se ieșe.

12. Dacă nu au apărut caractere noi „sigure” în setul $LatinChars$, care include acum noi valori de acuratețe, se anunță mesajul „Sunt necesare mai multe date!”.

13. În caz contrar, se merge la pasul 5.

Act. 2.4 - Dezvoltarea modelelor de recunoaștere a caracterelor (parteneri: UAIC, UB)

A. Implementarea de modele neuronale de recunoaștere a caracterelor în context și de transliterare a lor din alfabet chirilic în latin.

Strategia UB vizează o recunoaștere a caracterelor direct din imagini, fără o detecție prealabilă a chenarelor ce încadrează fiecare caracter. Intuiția noastră este că, în acest caz, vom efectua o recunoaștere a caracterelor în context, mai eficientă decât detecția și clasificarea chenarelor individuale. Arhitectura propusă pentru recunoașterea caracterelor este bazată pe o rețea de tip LSTM (Long-Short-Term-Memory) și CNN (Convolutional Neural Network) cu CTC drept funcție de cost (*loss*). Detalii despre ansamblul sugerat, LSTM-CNN-CTC și aplicabilitatea sa în OCR (*optical character recognition*) pot fi găsite în (Shi et al., 2015)²⁰.

²⁰ <https://arxiv.org/abs/1507.05717>

Act 2.5 - Dezvoltarea modelelor de clasificare lexicală

A. Aplicarea de modele n-gram și de distanță lexicală la clusterizarea formelor flexionate ale cuvintelor în clase corespunzând acelorași leme și părți de vorbire (partener: UAIC)

Procesul de lematizare al formelor flexionate din limba română veche este abordat prin antrenarea unei rețele neuronale recurente bazate pe arhitectura codor-decodor prezentată în (Sutskever et al. 2014) și cu funcția de cost dată de distanță Levenshtein dintre lema prezisă și lema adevărată a formei flexionate, primită în intrare.

Din lipsa adnotărilor de lemă din vocabularul ROCC, în primă instanță modelul este antrenat pe un set restrâns de perechi (*formă flexionată, lemă*), extrase din corpusul Marcell²¹ și validate cu intrările din exporturile publice ale bazei de date dexonline. Setul de perechi a fost împărțit în mod aleatoriu în raport de 80/20 % pentru antrenare și respectiv validare. În urma evaluării, modelul a obținut următoarele rezultate din 3.191 predicții: (i) 2.690 (84,3%) leme au fost prezise corect (distanță Levenshtein 0), (ii) 347 (10,87%) - distanță Levenshtein 1, (iii) 103 (3,22 %) distanță Levenshtein 2, (iv) 39 (1,22%) - distanță Levenshtein 3, (v) 8 (0,25%) - distanță Levenshtein 4, (vi) 4 (0,12%) - distanță Levenshtein 5.

Pentru îmbunătățirea rezultatelor, arhitectura modelului va fi modificată pentru a adăuga un nivel de atenție la rețea-a-codor (Bahdanau et al. 2015). Arhitectura nouă va fi reantrenată pe un set de date nou, extras din Monumenta Linguae Dacoromanorum²².

A. Aplicarea de modele de kerneluri de siruri și clusterizare spectrală la clasificarea cuvintelor în clase de leme și părți de vorbire (partener: UB)

Metodele de tip *string kernel*, lucrând la nivel de caracter, pot învăța informații utile privind structura lexicală a cuvintelor. Ele au fost aplicate cu succes în detectarea automată a genului substantivelor în limba română (Năstase și Popescu, 2009). Clusterizarea spectrală în combinație cu *string kernels* au dat rezultate promițătoare pe date preliminare din limba română modernă. Urmează ca metoda să fie testată pe limba română veche când vor fi disponibile date.

Act 2.6 - Dezvoltarea de modele de prelucrări semantice (parteneri: UAIC, UB)

A. Implementarea de modele semantice (word embeddings, GloVe, bayesienne) utilizând corpusul lexical diacronic construit

Clusterele de cuvinte sunt obținute prin gruparea cuvintelor descrise ca vectori cu valori reale, N-dimensionale. O modalitate de a obține acești vectori este de a utiliza grupuri de cuvinte generate de algoritmul *word2vec*²³ (Mikolov et al., 2013). *word2vec* ia în intrare un corpus și produce vectori de cuvinte cu valori reale în ieșire, cu o dimensiune fixă, dată ca parametru.

²¹ <https://relate.racai.ro/marcell/>

²² <http://consilr.info.uaic.ro/~mld/monumenta/index.html>

²³ <https://deeplearning4j.org/word2vec.html>

word2vec este de fapt o rețea neuronală recursivă, neuronii putând fi instruiți pentru a anticipa cuvântul următor într-o secvență dată de cuvintele dintr-un context (modelul - *bag-of-words*²⁴), fie pentru a ghici cuvintele din context (într-o fereastră de lungime fixă) având dat cuvântul curent (modelul *Skip-gram*²⁵). Un produs de formare a rețelei neuronale este matricea de ponderi care, în timpul antrenamentului, realizează mediile valorilor tuturor contextelor pentru toate cuvintele din vocabular, cuantificând astfel toate contextele unui cuvânt într-unul singur. Fig. 6 prezintă un exemplu de vector, cu valori reale, cu 100 de dimensiuni pentru cuvântul *vicar*, generat de algoritmul *word2vec* folosind modelul Skip-gramelor. Acest rezultat a fost aplicat asupra unei plaje de aproximativ 150 de milioane de cuvinte din corpusul CoRoLa:

```

vicar -0.204837 -0.095946 1.545269 0.914710 0.379946 -0.837752 -0.837671 -0.210816
-0.906490 -0.835662 -0.014551 1.079050 0.700285 0.752045 -1.691685 0.224929 2.237422
0.365855 -1.145924 0.299107 -2.198976 -0.027662 0.115228 0.075495 -0.530107 -1.371561
0.321837 0.261549 0.806120 -2.381392 1.429545 -0.678362 -0.986926 0.012122 -0.954867
-2.356951 0.212765 -0.271346 -1.226954 -1.475972 -0.279546 0.682138 -1.209563 -0.516240
0.421203 0.123969 -0.944503 0.258330 0.789713 -2.288659 -0.464814 1.766363 0.155666
0.319247 0.425142 1.481797 0.708248 -0.894684 1.628142 0.371647 -0.160769 0.565580
-1.102181 0.255940 -0.732641 -0.599247 0.130204 -1.453263 -0.154969 0.120765 -1.267940
0.119760 1.153221 1.419330 -0.163977 0.584624 -0.559195 -0.991894 -0.116130 1.009611
0.295165 -0.033827 0.854728 -0.963736 0.899420 -0.126685 -0.775878 -0.908373 -0.647173
-0.812671 -0.524420 0.066738 0.116809 0.567850 1.479700 -2.346341 -1.186089 -0.112726
0.250200 -2.116366

```

Figura 6: Un vector de 100 de dimensiuni care codifică cuvântul *vicar* (cuvântul este urmat de valorile, separate spațial, ale celor 100 de dimensiuni)

Plecând de la acest exemplu, pe care deja avem rezultate de antrenare, ulterior, în funcție de cantitatea de date existente în baza de date a proiectului DeLORo, vom aplica acest algoritm și pe intrările aferente acestuia.

B. Construirea de modele semantice specifice provinciilor istorice ale României și diferitelor perioade și aplicarea lor în detectarea schimbărilor lexicale în evoluția limbii române

În experimente preliminare s-au utilizat clasificatori de text, gen *Naïve Bayes* (generativ) și *maximum entropy*²⁶ (probabilistic), rețele neuronale, gen *support vector machines* (clasificator bazat pe metode kernel) și *long short-term memory* (rețea recurrentă cu celule de memorie), precum și modele probabilistice, gen *latent Dirichlet allocation* și *latent semantic analysis*, prin care se poate măsura similaritatea semantică.

Act. 2.7 - Antrenarea și evaluarea iterativă a modelelor de identificare a obiectelor și recunoaștere a caracterelor - Iterația I (parteneri: UAIC, UB)

²⁴ <https://deeplearning4j.org/bagofwords-tf-idf>

²⁵ <http://mccormickml.com/2016/04/19/word2vec-tutorial-the-skip-gram-model/>

²⁶ <https://www.sciencedirect.com/topics/engineering/maximum-entropy>

A. Iterația-I (baseline): modelele implementare vor fi antrenate pe datele culese la zi prin adnotare manuală și rezultate din alinieri

Metoda de antrenare este similară celei descrise în secțiunea A.2.3 - B.

Act 2.8 - Analiza lexicală și semantică a limbii române în diacronie și sincronie - I

A. Studiu lexical-semantic asupra evoluției limbii române din perspectivă diacronică (secolele XVI – XIX) și sincronică (regiunile Moldova istorică, Muntenia și Transilvania) - (parteneri: UAIC, UB)

Considerațiile de diacronie și sincronie a limbii române, care urmează, vor putea fi utilizate ca reper pentru verificarea rezultatelor modelelor automate.

În sincronie:

Pentru perioada sec. XVI-XVII și începutul secolului următor, nu se pot face delimitări spațiale ferme, la nivel lexico-semantic, deoarece texte scrise într-o zonă au fost copiate de cărurari din celelalte regiuni de limbă română și fiecare dintre copiști intervine asupra textului originar, cu modificări mai mult sau mai puțin masive. De exemplu, *Palia de la Orăștie* a fost tradusă în zona Banat - Hunedoara și tipărită de Șerban Coresi, din Muntenia, astfel încât textul prezintă elemente specifice ambelor arii dialectale. Pe de altă parte, în cazul lui Coresi, mai cu seamă ultimele cărți publicate de el, texte sursă care au fost diortosite (editate), suportă modificări esențiale, dar este un caz particular.

Un alt exemplu de miscelană lexical este *Biblia de la 1688*, care a fost diortosită la București de editori munteni, are ca texte sursă, pentru Vechiul Testament, Ms. 45 care prezintă trăsături moldovenești, și, pentru Noul Testament, sursă este cartea tipărită la Alba Iulia în 1648 de ardeleni, *Noul Testament de la Bălgad*.

Pentru perioada mai nouă, a doua jumătate a secolului al XVIII-lea - secolul al XIX-lea, pot fi stabilite diferențe la nivelul limbii literare între cele trei regiuni lingvistice mari (Moldova istorică, Transilvania - cu Crișana, Banat, Maramureș și Muntenia, cu Oltenia și Dobrogea). G. Ivănescu, în *Istoria limbii române*, 1980, demonstrează existența unor variante de limbă literară în această perioadă. Odată cu marii clasici ai literaturii și cu dezvoltarea presei, limba română literară se unifică într-o formă supradialectală. Și aici avem însă și excepții. De exemplu *Gazeta de Transilvania* (cu suplimentul său literar) preia numeroase materiale (articole, știri, texte literare etc.) din Moldova - *Albina românească* și din Muntenia - *Curierul românesc*.

De aceea, credem că va fi foarte dificil să identificăm automat modele de limbă pentru cele trei regiuni pe baze statistice.

În diacronie:

Diferențele lexicale de la un secol la altul, mai ales între epoca veche și cea modernă, sunt importante și relativ facil de identificat cu mijloace statistice. Ne referim aici la partea formală a

lexicului. În privința semanticii, lucrurile sunt mai complicate. Numai corelarea cu eDLTR ar putea duce la un succes parțial în identificarea modificărilor semantice suportate de un cuvânt în timp.

Distincția pe regiuni poate fi făcută, cum am notat mai sus, pentru o perioadă relativ scurtă din istoria limbii: cca 1750 - 1863. Însă, la sfârșitul perioadei respective, începe să se manifeste procesul de impunere a limbii literare supradialectale.

Se impune o observație valabilă pentru ambele tipuri de analiză statistică (în sincronie și în diacronie): transliterările rezultate în urma adnotării imaginilor sunt dificil de prelucrat automat în situațiile cuvintelor care apar, în originalul scris cu alfabet chirilic, scrise incomplet, abreviate, și cele care prezintă suprascrieri (text interliniar). Cum s-a stabilit, au fost transliterate și adnotate ca atare toate aceste situații, păstrându-se forma din tipărituri și manuscrise, inclusiv erorile de scriere. Cu excepția abrevierilor care formează o listă relativ stabilă, celealte lexeme pot fi recunoscute numai pe baza lexiconului extras din texte editate critic (situația transcrierilor editate critic este notată în *Lista surselor*; tot acolo am notat dacă avem la dispoziție o variantă electronică fără erori a acestor transcrieri).

Considerații asupra modelelor:

Indiferent de tipul de algoritm folosit, cele mai cunoscute tehnici și modele de minerit al textului, utilizate frecvent în diverse aplicații din domeniul prelucrării limbajului natural, cu un accent deosebit pe diacronie, au atât avantaje, cât și dezavantaje. În cazul DeLoRo, mineritul textului urmărește să extragă informații utile dintr-un corpus (repozitoriu de date monitorizate în timp) prin identificarea și explorarea tiparelor care răspund cu o acuratețe de peste 90%.

Cercetările anterioare au fost centrate pe înțelegerea asemănărilor/deosebirilor de lexic din patru corpusuri (colecții de publicații din Moldova istorică, Țara Românească, Transilvania și Basarabia). Au fost combinate tehniciile amintite mai sus, punându-se în evidență mai ales repetarea cuvintelor și numărul normalizat de apariții, cu distanțele semantice extrase din ontologii sau bazate pe modele semantice precum LSA, LDA etc.

Considerăm că modelarea statistică a textelor este o direcție de cercetare care necesită noi metode de evaluare și interpretare a rezultatelor, pentru că rezultatele sunt adesea afectate de zgromot, proprietăți speciale ale cuvintelor (cum ar fi polisemia), dimensiunea unui corpus etc.

Act 2.9 - Diseminare și permanentizare II (parteneri: UAIC, UB)

A. Elaborarea și comunicarea de lucrări științifice.

Lista comunicărilor ținute de membri ai proiectului pe teme care au legătură cu proiectul:

- D. Cristea, C. Pădurariu, P. Rebeja, L. A. Scutelnicu, M. Onofrei-Plamadă, P. A. Crucianu, C. Bolea, D. Gîfu, G. Haja, I. Tamba, R. Vieru (2021). Short-circuiting manual annotation in the process of building a corpus of old Cyrillic Romanian, Simpozionul „Sisteme Inteligente și Aplicații”, în cadrul „Zilelor Academice Ieșene”, Ediția a XXXVI-a, 22 octombrie, Iași, Romania.

- D. Cristea, P. Rebeja, C. Pădurariu, C. Bolea (2021). Technologies and Resources for Language Processing. An European survey and the Iași Portal. Invited talk in the Romanian AI

Days (online), 24 noiembrie (online).

• S. C. Bolea, D. Cristea, P. A. Crucianu, G. Dumitrescu, D. Gîfu, G. Haja, M. Onofrei, C. Pădurariu, P. Rebeja, L. A. Scutelnicu, E. I. Tamba, R. Vieru, *Scriserile vechi românești în atenția cititorului de azi: Artificial Intelligence Models (Deep Learning) Applied in the Analysis of Old Romanian Language (DeLORO – Deep Learning for Old Romanian)*, la A X-a ediție a Colocviului internațional „Lexicografia academică românească. Provocările informatizării”, organizat de Institutul de Filologie Română „A. Philippide”, Institutul de Informatică Teoretică, de la ARFI, Institutul de Cercetări Interdisciplinare - Departamentalul Științe Socio-Umane, de la UAIC, Echipele proiectelor TAFOC și eRomLex, 27-28 mai 2021.

• G. Haja, E. I. Tamba, *Modele de inteligență artificială (deep learning) aplicate în analiza sintactico-semantică a limbii române vechi (DeLORO). Perspectiva filologică*, la Conferința Internațională „Zilele Sextil Pușcariu”, organizată de Institutul de Lingvistică și Istorie Literară „Sextil Pușcariu”, Academia Română - Filiala Cluj, 9-10 septembrie 2021.

Lista lucrărilor elaborate de membri ai proiectului pe teme care au legătură cu proiectul:

• V. Barbu Mititelu, E. Irimia, D. Tufiș, D. Cristea (2020). Proceedings of the 15th International Conference “Linguistic Resources and Tools for Natural Language Processing”, Online, 14-15 December 2020, ISSN 1843-911x, “Alexandru Ioan Cuza” University editing house, 191 pages.

B. Organizarea unor evenimente de diseminare a activităților și rezultatelor proiectului.

1. The 15th International Conference “Linguistic Resources and Tools for Natural Language Processing”, Online, 14-15 December 2020.
2. EUROLAN 2021, The 15th Eurolan Edition, Introduction to Linked Data for Linguistics, Online Training School, 8-12 February 2021.
3. Colocviul internațional Lexicografia Academică Românească. Provocările Informatizării, Ediția a X-a online, Institutul de Filologie Română „A. Philippide”, Institutul de Informatică Teoretică, 27-28 mai 2021, Iași, Romania.
4. Romanian AI Days 2021, The 2nd Edition, Online, 24-25 November 2021.

Act 2.10 - Site web, implementare și integrare - II (partener: UAIC)

A. Se va actualiza structura site-ului web al Proiectului, prin adăugarea interfețelor, aplicațiilor și serviciilor web create.

În cadrul acestei activități s-a dezvoltat în continuare site-ului proiectului²⁷. Acesta este găzduit pe serverul institutului și are ca repository adresa de Github²⁸. Site-ul este actualizat ori de câte ori intervin modificări sau completări în proiect.

²⁷ <http://deloro.iit.academiaromana-is.ro/>

²⁸ <https://github.com/deloro-project/deloro-project.github.io>

B. Se va ține la zi conținutul site-ului Proiectului, cu noile resurse achiziționate în ROCC, cu tehnologiile implementate și cu datele de antrenare ale modelelor elaborate.

În cadrul site-ului de prezentare al proiectului, există trimitere către interfața OOCIAT, unde informațiile sunt actualizate în timp real, în funcție de datele noi adăugate și existente în baza de date.

Referințe

- Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. CoRR.
- Cristea, D., Rebeja, P., Pădurariu, C.C., Onofrei, M., Scutelnicu, A. (2021, submitted). Data Structure and Acquisition in DeLORo – a Technology for Deciphering Old Cyrillic-Romanian Documents, for *Proceedings of the XVIth conference Linguistic Resources and Tools for Natural Language Processing*, ConsILR-2021, Dec. 13-14.
- Mikolov T., Sutskever I., Chen K., Corrado G., Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality, <https://arxiv.org/abs/1310.4546>.
- Năstase, V., Popescu, M. (2009) What's in a name? In some languages, grammatical gender. *Proceedings of the EMNLP 2009 Conference on Empirical Methods in Natural Language Processing*, Singapore.
- Pădurariu, C.C. (2021). Image to Text Aligner for Augmentation of Datasets, 3rd PhD Report, “Alexandru Ioan Cuza” University of Iași, Faculty of Computer Science, May.
- Ren, S., He, K., Girshick, R. and Sun, J. (2015). “Faster R-CNN: Towards real-time object detection with region proposal networks”. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Volume 1, ser. NIPS’15, pp. 111-212.
- Shi, B., Bai, X., Yao, C. (2015). An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition, CoRR.
- Sutskever, I., Vinyals, O., Le, Q. V., (2014). Sequence to Sequence Learning with Neural Networks, NeurIPS.

III. Concluzii

Obiectivele Etapei a II-a a proiectului DeLORo au fost realizate în totalitate.

Director Proiect,
Dan Cristea

