CURSUS

Științele limbajului

Roxana Patraș
Loredana Cuzmici
(editors)

Tools, Methods, and Solutions
for the Exploration of Romanian Corpora

INSTITUTUL EUROPEAN

Roxana Patraş, Loredana Cuzmici (editors), *Tools, Methods, and Solutions for the Exploration of Romanian Corpora*

INSTITUTUL EUROPEAN
Iaşi, Str. Băluşescu nr. 2, et. I, O P 6, CP 1309
euroedit@hotmail.com.; www.euroinst.ro

Descrierea CIP a Bibliotecii Naţionale a României

Tools, methods, and solutions for the exploration of Romanian corpora /
ed.: Roxana Patraş, Loredana Cuzmici. - Iaşi : Institutul European, 2024
Conţine bibliografie
ISBN 978-606-24-0395-9

I. Patraş, Roxana (ed.)
II. Cuzmici, Loredana (ed.)

811.135.1

PRINTED IN ROMANIA

Roxana Patraș, Loredana Cuzmici
(editors)

TOOLS, METHODS, AND SOLUTIONS FOR THE EXPLORATION OF ROMANIAN CORPORA

INSTITUTUL EUROPEAN
2024

This work was supported by a grant of the Romanian Ministry of Education and Research, CNCS–UEFISCDI, project number PN-III-P1-1.1-TE-2019-0127, within PNCDI III.

The revision and editing of this collection was performed by BolderWords translation & revision and was supported by the Altissia Chair in Digital Cultures and Ethics held by Chris Tanasescu at Université catholique de Louvain.

Table of Contents

Tools and Resources for Digitizing Historical Romanian Documents

Victoria BOBICEV, Tudor BUMBU
Ludmila MALAHOV, Alexandru COLESNICOV
Svetlana COJOCARU, Liudmila BURTSEVA
Cătălina MĂRĂNDUC

1. Introduction

Digitizing and providing online access to historical literary and cultural treasures are top priorities in the Digital Agenda for Europe.[1] The Romanian language, considered a less commonly used language, is increasingly benefiting from the development of digital editions, specialized corpora, archives, and dedicated platforms for preserving and analyzing cultural heritage. Cultural policy initiatives in Romanian-speaking regions focus on the study, digitization, and preservation of their heritage, including Romanian texts in Cyrillic and mixed scripts/ alphabets. The digitization process entails overcoming challenges related to recognizing, correcting, and transliterating printed Romanian texts in Cyrillic and mixed scripts (Boian *et al.* 2014, Ciubotaru *et al.* 2015). A complex technology has been elaborated in order to support the digitization of historical Romanian heritage printed in the Cyrillic alphabet in the 17th–20th centuries (Colesnicov *et al.* 2021).

The endeavor to address the linguistic heritage of Romanian history involves tackling several specific challenges, including: (1) dealing with a multitude of language evolution periods; (2) coping with the scarcity of widely available resources; (3) managing the diverse array of alphabets used in historical printings, including mixed Cyrillic–Latin "transition alphabets" (Cazimir 2006); (4) overcoming the absence of reliable tools for accurately recognizing Cyrillic letters from various historical eras; (5) addressing the shortage of lexicons suitable for the

[1] http://www.digitisation.eu/community/map-of-the-digitisation-landscape/

time periods of these resources. To surmount these obstacles, we have developed a comprehensive platform that seamlessly integrates a suite of software components that enable image processing, text recognition, and transliteration into contemporary Latin script. This platform has been finely tuned to handle text recognition and transliteration across different historical epochs and to accommodate variations in alphabets used in Romanian-language printings in both Romania and the Republic of Moldova. Follow-up efforts have been concentrated on annotating historical and other non-standard texts. Carefully checked morphological, syntactic, and semantic annotation of the collection of historical documents, regional folklore, and other non-standard text materials from Moldova and Romania resulted in the creation of the RoDia Dependency Treebank. In order for this valuable resource to be more widely accessible, its annotation has been transformed and a Non-Standard Romanian Universal Dependency corpus has been added to the collection of the Universal Dependency (UD) project.

The description of our work is organized into three main parts. The first part (*Exploring the Digitization of Old Romanian Printing*) introduces the collections of scanned historical Romanian documents and the challenges associated with digitizing them. The second part (*The Digitization Platform*) provides a detailed description of the new digitization platform, its functionality and the resources it uses. The third part (*Annotation and Conversion in UD*) covers the corpus annotation and is divided into three subsections: 4.1 concentrates on morphological tagging; 4.2 describes syntactic annotation; 4.3 gives an account of the transformation of the annotation in Universal Dependency format; 4.4 introduces the semantic annotation principles; the automated parsing experiments and their results are discussed in its subsection; 4.5. Finally, the conclusion completes the paper by listing the main achievements of the described work.

2. Exploring the Digitization of Old Romanian Printing

We work with old Romanian books written in the Cyrillic script. This is a challenge faced by both researchers of Romania and scholars of the Republic of Moldova. Since these countries previously constituted a single state, historical

documents (written in old Romanian Cyrillic) are common to them both (Cristea *et al.* 2018). The starting point for our work is the scanned text, *i.e.*, the text presented in the form of page images. These text images were sourced from two electronic libraries of texts: the Bucharest Digital Library[2] and the National Library of the Republic of Moldova.[3] The National Library holds a collection of approximately 21,000 old and rare books. An important source of scanned collections is its National Digital Library, known as *Moldavica,*[4] whose collections of old books printed in Slavonic characters consists of 1,194 scanned documents, and the collection of national magazines from 1867 to 1945 contains 4,276 documents. The National Digital Library of Romania[5] includes 724 old Romanian periodicals and 1228 old Romanian book and bibliophile resources in Romanian, Greek and Slavonic, among other languages (*Figure 1*).

17th century	*Noul Testament* [New Testament], 1648
18th century	*Fiziognomie* [Physiognomy], 1785 *Ducere către aritmetica* [Leading to Arithmetic], 1785 *De obste Gheografie* [General Geography], 1795 *Așezamânt* [Settlement], 1786
19th century	*Epistolariu* [Epistolary], 1841 *Gramatica românească* [Romanian grammar], 1835 *Legiuirea Caragea* [Caragea Law], 1818
20th century	*Folclor din părțile codrilor* [Folklore from the Parts of the Forests], 1973

Table 1: Scanned, recognized, and transliterated books.

Table 1 lists some of the linguistic sources with which we worked. The sources are of different historical periods starting with the oldest ones printed in the very first printing houses situated in Moldova in the 17th–18th centuries. Romanian Cyrillic fonts, especially from the selected epoch, are much less variable than Latin fonts. The usage of the Cyrillic script is connected with the Slavonic liturgical language of

[2] http://www.bibnat.ro/
[3] http://www.bnrm.md/
[4] moldavica.bnrm.md/
[5] http://digitool.bibnat.ro/R?RN=171659921

the Orthodox Church. *Figure 2* presents three samples of scanned books from different periods. The document quality can vary from perfect to almost unacceptable. The general problem is caused by the variations in writing and by language changes. The main objective is to make these documents accessible to the general public, by presenting them in a way that uses contemporary orthography.

The process of converting paper-based historical documents into searchable electronic files involves two obstacles that have not been fully overcome until now. The first obstacle is that historical typography is inconsistent and idiosyncratic. Due to their age, pages in old documents feature speckles, distortions, and stains. Also, historical fonts vary considerably, even within a given book, and they are more difficult to read than their contemporary counterparts. Furthermore, modern-day linguistic tools and resources often cannot detect the peculiarities of historical language variations. As a result—and especially since each text yields its own mix of features and problems—the quality of the text recognition varies dramatically. The second obstacle is the extent to which language, including spelling, has shifted since the historical documents were originally created. Texts in original historical orthography are unevenly supported by OCR software (for example, 18th-century Romanian Cyrillic writing).

For processing these historical documents, we propose a new digitization pipeline which enables recognition of historical Romanian heritage documents that were printed with Cyrillic letters in the 17th–20th centuries. This is based on the commercial OCR tool ABBYY Finereader (AFR)[6]. In comparison with open-source free OCRopus[7] and Tesseract,[8] AFR has numerous advantages. It offers users a convenient graphical shell from which the document's entire recognition cycle is available, starting with scanning. It provides complex image correction (for example, the correction of trapezoid distortion), page segmentation with automatic detection of the segment type (text, table, or picture), an analysis of the table structure, dehyphenation, manual editing after recognition. The user can choose to input a new language, specifying its alphabet and downloading an additional user dictionary. In

[6] https://pdf.abbyy.com/
[7] https://github.com/ocropus-archive/DUP-ocropy
[8] https://github.com/tesseract-ocr/tesseract

other cases, AFR can be trained on the challenging document, and the accumulated character patterns (OCR models) can be downloaded and uploaded for use in subsequent projects. OCRopus and Tesseract do not offer a single product that provides both a full recognition cycle and a graphical shell.

This technology supports virtual keyboards, fonts, a transliteration tool, and other possibilities. We have created additional Python programs such as a tool to select the historical period and the geographical region, where the text was printed, for instance Iași, Bucharest, Târgoviște, Bălgrad, Uniev, Sas Sebeș, Snagov, Buzău, etc. (Bumbu 2021, Cojocaru *et al.* 2017). In particular, for Bucharest, the system is trained in recognizing various fonts such as the Royal Typography's or the Bucharest Metropolitan Chair's.

3. The Digitization Platform

We have developed a Digitization Platform (DP) for digitizing old documents written in Romanian Cyrillic. DP is a web application that features an interactive graphical interface and a set of APIs designed and managed through Django Rest Framework.[9] This application offers seven consecutive steps to enable recognition of Romanian Cyrillic documents from the 17th–20th centuries, transliteration of the texts into Latin alphabet, editing of recognized/transliterated texts, and downloading or publication of the results (Bumbu 2023).

The platform includes:

- image processing engines, such as ScanTailor,[10] AFR, and OpenCV[11];
- OCR models for Cyrillic characters, trained on datasets gathered from documents printed in the 17th–20th centuries;
- a transliteration tool that converts from old Romanian Cyrillic into contemporary Latin script;

[9] https://www.django-rest-framework.org/
[10] https://scantailor.org/
[11] https://opencv.org/

- virtual keyboards specific to the alphabets used in previously mentioned periods, such as the Romanian Cyrillic alphabet, the transition alphabet, and the Soviet Cyrillic alphabet.

The DP architecture (*Figure 3*) consists of a set of modules that are organized into four functional groups. The first functional group (G1) deals with image processing. This group is equipped with two basic preprocessing modules—image binarization and resolution correction—and can be expanded through the integration of third-party applications like Scan Tailor, AFR, OpenCV, and GIMP. The second functional group (G2) consists of modules that handle the optical recognition of documents. Among the options we can mention the selection of the OCR model depending on the historical period, the selection of a word dictionary used during recognition, activation of the editing module for the recognized text or the use of the OCR exception dictionary. The OCR engine is based on AFR and has models trained with datasets collected from documents printed in the 17th–20th centuries. Users can also add new OCR models. These models use the FineReader XML format, which contains the OCR model configurations, training datasets, required alphabets, and word dictionaries. The third functional group (G3) contains components related to transliterating the recognized text. Primary modules in G3 ensure spelling updates (upon request), the use of dictionaries of exceptions for transliteration, editing of the resulting transliterated text with a spellchecker, and automatic text correction using an artificial intelligence system. Secondary modules in this group look after such tasks as replacing apostrophes with hyphens and deleting hyphens that had been used to break words across line.

The fourth functional group contains modules devoted to managing and publishing digitized documents. These tasks involve saving recognized/transliterated texts in various formats; downloading processed images; uploading original documents and processed images to the cloud; saving the state of the digitized object in the database; and expanding word dictionaries. The digitization process on the platform includes seven steps, as outlined below.

Step1: File upload

One or more files can be processed in a single digitization cycle. The following file types are supported: png, jpeg, tiff. The total size of all uploaded files must not exceed 700MB, and the size of each file is limited to 100MB. Files can be selected from the user's computer. When two or more files are selected, all of them will be processed with the same processing options, and the user is responsible for ensuring that the uploaded files are from the same period, use the same alphabet, and require the same image preprocessing options. If the user's document sets reflect multiple periods or require different image preprocessing options, then the sets should be processed in different digitization cycles (*Figure 4*).

Step 2: Preprocessing of uploaded images

In this step, the users can choose the preprocessing engine and preprocessing options needed for uploaded pages. Scan Tailor is the first preprocessing engine displayed in the list of options in step 2 and offers the widest range of options of all three engines proposed. Unlike with the other engines, two ways of using the Scan Tailor engine for image preprocessing are proposed (*Figure 5*).

Moving forward, the spotlight shifts to the preprocessing engine integrated into AFR. Currently, the testing phase is limited to exploring the fundamental image preprocessing functions, encompassing tasks such as rectifying image resolution discrepancies, identifying the optimal resolution for a given image, correcting automatic page orientation, transforming images to black and white, mitigating ISO noise,[12] and aligning text lines. The OpenCV engine, another formidable contender, affords users the flexibility to manually set the resolution of the pre-processed image. Additionally, it stands out by amalgamating filters within a single option, streamlining processes to clean the image of speckles and reduce ISO noise simultaneously.

[12] ISO noise is the digital equivalent of *grain* in digital cameras.

Step 3. Optical character recognition

In this step, the user identifies the historical period of the document and the OCR model based on which the document is recognized (*Figure 6*). An example of a recognized document is shown in *Figure 7*.

Step 4. Verification and editing of the recognized text

The text obtained in the previous step can be processed manually: this step integrates text editing modules such as virtual keyboards for different alphabets and spellchecks. An example is shown in *Figure 8*.

Step 5. Transliteration of the recognized and edited text

In this step, the transliteration tool from the old Romanian Cyrillic alphabet into the contemporary Latin alphabet is integrated. The transliteration is based on rules sets according to different epochs and alphabets (Romanian Cyrillic, Moldovan Cyrillic, transitional alphabets) and exception dictionaries. *Figure 9* provides an example, with recognized text on the left and its transliteration on the right.

Step 6. Verification and editing of the transliterated text

This step in which the transliterated text is verified is similar to step 4. Both the user and the spellchecker work with the same word dictionaries.

Step 7. Saving of the results

Together, all of the results, images, and texts that have been obtained from the input after steps 1 through 6 were completed, constitute a digitized document (see also *Figure 9*). This step also involves modules for downloading the preprocessed images, the recognized text, and the transliterated text and for uploading these texts and images to portal e-Moldova[13] (Caftanatov 2021).

To facilitate accessibility to the old documents we elaborated:

- two OCR models trained on over 10,000 training examples and a dictionary of over 4,500 words for the 17th century;

[13] https://emoldova.org/

- more than 10 OCR models with dictionaries containing more than 10,000 words for the 18th and 19th centuries;
- and an OCR model containing over 450,000 words for the 20th century.

The recognition of texts from the 18th and 19th centuries resulted in a WER (Word Error Rate) of 3-4.5%, and the WER for the 17th century was more than 6%.

4. Annotation and Conversion in Universal Dependencies

A qualitative corpus needs linguistic annotation. This section describes the work on morphosyntactic annotation of a corpus of old texts. Initially, the corpus building was started by Cenel Augusto Perez (Perez 2014) and called the Romanian Dependency Treebank (UAIC-RoDia-DepTb), or RoDia for short. In 2017, it obtained an International Standard Language Resource Number (ISLRN 156-635-615-024-0),[14] and its size was 16,000 sentences (or 320,000 tokens, including punctuation). Verification of this corpus's morphological and syntactic annotation was entirely manual. Since 2014, the corpus has been growing thanks to the ongoing work of Cătălina Mărănduc (Mărănduc 2022). In addition to being syntactically annotated, the corpus is presented in several formats: (1) the classic syntactic, containing more than 39,000 sentences, (2) the Universal Dependencies syntactic, and (3) a new syntactic-semantic format. To our knowledge, it is the largest syntactically annotated Romanian corpus where all of the verification was done only manually. Currently, the RoDia corpus contains annotated old texts and small volumes of non-standards texts, including regional and social media content, which enabled us to focus on the annotation of non-standard text types, such as oral regional fiction, social media communication, poetry, and old Romanian texts.

The creation of this corpus served several objectives:

(1) to attract attention to the linguistic diversity of the Romanian language;

(2) to conduct a diachronic analysis of the Romanian language;

(3) to create a standard approach to annotating various types of Romanian texts;

[14] https://islrn.org/resources/156-635-615-024-0/

(4) to build robust machine learning models for different types of annotations.

Old Romanian Cyrillic	Modern Romanian	Morpho syntactic tag	Morphosyntactic tag description
Їwсиф	Iosif	Npmsrn	Noun, personal, masculine, singular, direct case, non-articulated
фѳци	fugi	Vmis3s	Verb, main, indefinite, singular, 3rd person, subjunctive mode
кѳ	cu	Spsa	Preposition, accusative case
ЇС	Iisus	Npmsrn	Noun, personal, masculine, singular, direct case, non-articulated

Table 2: An example of text enriched with morphosyntactic tags and lemmas

4.1. Morphological annotation

The texts were automatically processed by the Ro-bin-hybrid Part-of-Speech (PoS) tagger (Simionescu 2011), using MULTEXT East project PoS tags (Erjavec 2012). This set of tags has been developed in unified mode for all European languages. Each tag is an abbreviation that starts with a upper-case letter, indicating the part of speech, and continues with a series of lower-case letters, which encode other morphological features dependent of the part of speech. For example, *Ncfpoy* deciphers as N=noun, c=common, f=feminine, p=plural, o=case (dative or genitive), and y=articulated.

FORM	POSTAG	LEMMA
anfivolie	Ncfsrn	amfivolie
anfivolii	Ncfprn	amfivolie
ăbăluiască	Vmsp3	ăbălui
palma	Ncfsry	palmă
sămînţa	Ncfsry	sămânță
partea	Ncfsry	parte
săvîrşi	Vmis3s	săvârși
săvîrşi	Vmn	săvârși

Table 3: A fragment of the lexicon for old Romanian texts, tagged by MULTEXT

Due to the rich morphology of the Romanian language, a set of as many as 614 morphosyntactic tags were developed for it during the MULTEXT project. This set was quite large and included a detailed description of the specifics of Romanian morphology. We simplified the tags, retaining 450 from the initial set and adding approximately 100 more tags to annotate specific elements of old Romanian. *Table 2* contains an example of text that has been enriched by the morphosyntactic tags and lemmas.

It is obvious that the accuracy of automated morphological tagging for this large set of tags is lower than it would be for a state-of-the-art equivalent used on the English language, for example. The best accuracy reported for the automated morphological tagging on modern Romanian texts and reduced tagset was 99% (Tufiș 2000); on old Romanian texts, it was considerably lower. To boost the accuracy for the non-standard texts, we enriched the dictionary for the Ro-bin-hybrid PoS-tagger with old Romanian words. *Table 3* contains several lexical units, each consisting of a word form, a word lemma, and a morphosyntactic description tag.

The first set of old texts was annotated using the tagger that had been trained on modern texts. Then, the texts were manually checked and corrected, and the tagger was trained on these corrected texts. Furthermore, a gold annotated corpus of non-standard texts was created through bootstrapping, whereby the processes of annotation, correction and re-training the tagger were repeated multiple times. Ultimately, the accuracy of PoS tagging for old texts reached 96% (Mărănduc 2022). It is extremely difficult to reach a higher level of accuracy, as each text that is subsequently added to the corpus differs considerably from those that have already been tagged, and the rate of words in that text that are unknown to the tagger is high, up to 12% of the new text. Each text that was added to the corpus was tagged in automate mode, and each tag was then checked and corrected manually. *Figure 10* presents an excerpt of the tagger output: text enriched with morphosyntactic information coded in XML format.

4.2. Syntactic annotation

Initially, the corpus had been annotated in accordance with dependency parsing conventions that were developed at "Alexandru Ioan Cuza" University of Iași (UAIC),

especially Functional Dependency Grammar (FDG), which features classical syntactic labels including numerous semantic sub-classifications of modifiers (Mărănduc 2017).

Dependency grammar is a class of modern grammatical theories that are based on the notion of dependency relation between linguistic units (that is, words) in a sentence. It means that words are connected to each other through directed links (Florea *et al.* 2014) forming a tree graph. The sentence predicate (verb) is considered the structural center of clause structure or a root of the graph. All other syntactic units (in this case, words) are connected to this verb or other words by means of links or dependencies forming branches and leaves of the graph. The main rule for this type of annotation is that every word except for the head (main verb) in a sentence has to be connected to only one other word, as a child is connected to a parent in a tree graph. The rules of connections vary depending on the corpus annotation guidelines. Usually, the type of connection is indicated by labels, such as "subject" or "modifier". The list of dependency types is limited and described in the annotation guidelines. For UAIC corpus annotation, 44 types of syntactic relations were defined; *Table 4* presents several examples of relation types.

Nr. crt.	**UAICtb syntactic relation**	English translation of relation	Example	English translation of example
1	a.adj.	adjectival attribute	fată *frumoasă*	*beautiful* girl
2	a.adv.	adverbial attribute	ziua *de ieri*	day *yesterday*
3	ap.	apposition	el, *autorul*	he, *the author*
4	a.pron.	attribute expressed by pronoun	cearta *cu el*	quarrel *with him*
5	a.subst.	attribute expressed by noun	cearta *cu tata*	quarrel *with Dad*
6	aux.	auxiliary verb	*am* terminat	*have* finished
7	a.vb.	verbal attribute	plăcerea *de a mânca*	the pleasure *of eating*
8	c.ag.	agent	bătut *de hoți*	beaten *by thieves*

Table 4: A sample of syntactic relation labels, with examples

The first texts of the corpus were annotated using a parser trained on the UAIC Romanian Dependency Treebank (UAICRoDepTb), and the automate annotation had to be verified and corrected manually because its accuracy on the non-standard old

texts was quite poor. Then, MaltParser[15] was trained on the verified texts, and new portions of the corpus were annotated using its various models. Every text, annotated by a parser was checked and corrected manually in order to create a gold standard corpus for further parser training.

Figure 11 presents RoDia in XML format, where the XML tag attributes *head* and *deprel* define the syntactic relation. The *head* value indicates the current word head; the *deprel* value specifies the label of the relation between the words. Each word is indexed within the sentence and has a unique identifier ("id") that is used to indicate the head. For example, the word with id=4 has head=1, which means that this word is connected to the word with id=1. The attribute *deprel*=c.c.l. means that, in relation to the head, this word has the role of *complement circumstanțial de loc* (adverbial phrase of place). The attribute *form* contains the initial form of the word as it appears in the text, and the attribute *postag* indicates the word's morphosyntactic description. In XML markup, it is possible to observe relations between words, but verifying and correcting the annotation in this format is quite difficult. In order to facilitate the work of linguists, we have developed a special graphical annotation tool called TreeAnnotator (Moruz 2008).

This graphical annotator has many advantages. For one, its visual design is user-friendly: the sentence is presented as a tree, where the head is located higher than its descendants, and where each of the other words in the sentence is presented as a branch stemming from the head and is positioned on the same vertical plane as the corresponding word listed in the version of the sentence that is presented along the bottom of the image. Another advantage is that the tree is completely editable. Any connection can be removed, added, redirected or renamed. All this can be done with the mouse, or through the use of a combination of keys, which increases the speed of work, as the mouse can be used to select a word or a relationship, while the left hand is used to push the keys. However, the labels are introduced manually, which increases the risk of typing errors. Still, the greatest drawback of this Graphical User Interface (GUI) is the fact that its software is no longer supported by its developper. The code

[15] http://www.maltparser.org/

can no longer be modified, which means that it can only be used for the old format of the UAIC treebank.

Figure 12 presents an example of a syntactic tree that has been visualized graphically using TreeAnnotator. The sentence is listed below the tree, and morphosyntactic tags (which are also editable) are presented below the words of the sentence. The menu at the top contains buttons with the editing options.

Currently, the corpus contains a small amount of contemporary texts, including social media content. A large part of the corpus consists of old texts from the 16th–19th centuries. The initial intention had been to annotate a larger number of language styles, in rather equal proportion; once we added the Alba Iulia New Testament (1645) to the corpus, however, the ecclesiastical style became the most represented. This is quite normal for that period, when written texts were predominantly found in churches. We recognize that the church style in our corpus is overrepresented to the detriment of other styles though it was not our intention. Subsequently, several other historical works were integrated into the corpus. These include a book containing moral philosophical sentences and narrative examples, *Flower of Gifts*, 1592 (*see* Gheţie and Mareş 1996); a book containing laws established by Prince Caragea in 1818; Neculce's *Chronicles of the Land of Moldavia* (1745); and Dosoftei's *Psalms*. Currently, the texts that have been annotated using the UAIC convention represent 34,794 sentences (714,377 tokens) and reflect an average of 20.53 words per sentence.

4.3. Universal Dependency Annotation

In order to widen the visibility of the corpus, its annotation has been transformed into the international Universal Dependencies project convention. Universal Dependencies[16] is a framework for the consistent annotation of grammar (parts of speech, morphological features, and syntactic dependencies) across different human languages. UD is an open community effort with over 500 contributors producing over 200 treebanks in over 100 languages (Marneffe *et al.* 2021).

The principles of the UAIC treebank annotation, which were established in 2007 and initially intended for didactic use, are quite different from those of the UD. UD

[16] https://universaldependencies.org/

annotation convention highlights meaningful words (such as nouns, verbs, adjectives) and subordinates functional words (such as conjunctions, prepositions, articles) to them. In the initially created UAIC corpus, the types of relations are highlighted, and the functional words are considered connecting nodes, thus they co-relate the head and the subordinated meaningful word in the tree. The main difference between these two systems of annotation lies in the annotation of functional words, such as copulative verbs: they are considered heads in the UAIC treebank, whereas they are subordinated in UD. Another difference is in the number and types of dependency labels. The UAIC treebank annotation uses the same tag to mark a relation expressed by a word as it does to mark a relation expressed by a subordinate clause. By contrast, UD uses a range of different tags, such as *ccomp*, *csubj* or *advcl* (where the initial *c* or the final *cl* means *clause*).

Figure 13 compares UAIC and UD connection principles. In the UAIC example, the complement "în ceriu" (to heaven) is connected to "sui" (ascends) as a *complement circumstanțial de loc* (adverbial phrase of place), where the preposition "în" is the head of "ceriu". In the UD example, the type of the complement is not specified, and the preposition "în" is subordinated to the more important word "ceriu."

UAIC and UD also differ in terms of the number and naming of syntactic relation types. UAIC identifies 14 kinds of adverbial modifiers, thus offers rich semantic information; however, UD proposes 65 syntactic relation types. *Table 5* presents several examples of how UD and UAIC syntactic relation types correspond to one another. As shown, there are some cases where a given UAIC relation type has an exact correspondence with a UD relation type, but there are also cases where a UAIC relation type corresponds to two or more UD relation types.

UD syntactic relation types	**UAIC syntactic relation types**	**Explanations**
amod, det, nummod	a.adj.	adjectival attribute
advmod	a.adv.	adverbial attribute
nsubj, csubj, nsubj:pass, csubj:pass	sbj.	subject
nmod:pmod, ccomp:pmod	c.prep	prepositional complement

Table 5: correspondences between UD and UAIC tags for syntactic relation types

As a result of these discrepancies, serious problems arose during the process of transformation of the syntactic annotation of the existing UAIC corpus into UD convention/ protocol of annotation. After detailed analysis, we created a rule-based transformation tool, called TREEOPS, to convert UAIC annotation into UD annotation (Colhon *et al.* 2017). The tool works in the XML format, and the rules follow an if-then (condition-action) logic. There are three types of rules:

- rules that have only one condition;
- rules that have two or more conditions (and actions); and
- rules that are used for structural transformations (such as reversing the direction of the relation between the noun and the prepositional case marker).

There is also a list of super-rules that determine the order in which the rules described above should be applied. There are more than 150 rules for the transformation into the UD convention. For example, for the transformation of an adverbial in the UD convention, there are three rules for each type of adverbials, as identified and explained below. Belonging to the second type of rules, these rules have two conditions: the condition of a dependency relation and that of a morphological analysis.

1. //word[@deprel='c.c.l.' and starts-with(@postag, 'R')]/@deprel => changeAttrValue('advmod')

- **If the word** has a dependency relation label c.c.l. and its morphological tag starts with R (which means it is an adverb), then the dependency relation label should be changed to advmod.

2. //word[@deprel='c.c.l.' and starts-with(@postag, 'V')]/@deprel => changeAttrValue('advcl')

- **If the word** has a dependency relation label *c.c.l.* and its morphological tag starts with *V* (which means it is verb), then the dependency relation label should be changed to *advcl*.

3. //word[@deprel='c.c.l.' and (starts-with(@postag, 'N') or starts-with(@postag, 'P') or starts-with(@postag, 'M'))]/@deprel => changeAttr Value('obl').

- **If the word** has a dependency relation label *c.c.l.* and its morphological tag starts with *N* (which means it is a noun), then the dependency relation label should be changed to *obl*.

The transformation errors are often caused by unexpected syntactic structures commonly encountered in a natural language, especially in a nonstandard one. Monitoring the output of TREEOPS is therefore absolutely necessary, although the task is not as time-consuming as that of correcting the morphological and syntactic output of the parser and PoS-tagger.

The final volume of the Romanian Nonstandard UD corpus is smaller than the initial RoDia corpus; all of the initial UAIC annotations that were converted into UD need to be verified manually, which is a slow process. A number of the transformed texts still needs to be verified and corrected.

4.4. Semantic Annotation

While UD version of the corpus improves the visibility of our corpus within the community, the UAIC annotation model is better tailored to the syntactic particularities of the Romanian language, and we still annotate in this format to preserve all the information that has been automatically annotated and carefully monitored in the initial corpus. However, syntactic annotation, even when it is detailed, does not code the semantic content of the sentences, and it therefore proved necessary to add a semantic layer to the classical treebank.

Used to highlight the semantic information present in syntactic and morphological UAIC-RoDepTb annotations (Mărănduc *et al.* 2017), the logical-semantic format has been inspired by the Tectogrammatic layer of the Prague Dependency Treebank (PDT) (Bohmová *et al.* 2003) and Abstract Meaning Representation (AMR) (Bănărescu *et al.* 2013). AMR semantic annotation is not a dependency tree, and rather than referring to words, its nodes consist of concepts; AMR therefore could not be adopted as it is for our semantic annotation. The principles

underlying the Tectogrammatic layer of the PDT correspond much more closely to our own. Furthermore, it includes categories for annotating semantic information found in exclamatory and interrogative sentence forms, and in verb moods and tenses, by considering the punctuation or morphological annotation as containing semantic information.

Enriching the syntactic annotation, we identified 96 semantic labels for semantic dependency relations, as summarized below (for the full list of semantic labels, *see* Mărănduc 2022).

- Adverbial relation labels indicate:
    - semantic labels for the relations of temporality and spatiality (tagged as TEMP, PAST, FTR, LOC);
    - logical relationships: cause (CAUS), purpose (PURP), consequence (CSQ), opposition (OPPOS), concession (CNCS), condition (COND), exception (EXCP), cumulation (CUMUL), association (ASSOC), reference (REFR), restriction (RESTR), result (RSLT).
- Information about the names of objects or actors is derived from the classification of the pronouns: appurtenance (APP), deictic (DX), emphatic (IDENT), undefined (UNDEF).
- The classification of functional words (which are considered to be connectors) provides the classification of relations.
    - Four of them are coordinating connectors: inclusion (CNADVS), reunion (CNCNCL), disjunction (CNDISJ), and intersection (CNCONJ).
    - A fifth is the subordinating connector (CNSBRD).
    - A sixth is the copulative verb “to be” (CNCOP) and serves to connect the subject and its description, as rendered by the predicate noun.
- Another set of labels conveys information regarding the reality of the action: optative (OPTV), uncertain (UNCTN), potentiality (POTN), generic (GNR), dynamic (DYN).

- Five quantifiers are considered: necessity (QNECES), possibility (QPOSSIB), existence (QEXIST), universality (QUNIV), and negation (QNEG).
- Another set of labels provides information about the communication in text: addressee (ADDR), blame (BLAM), greeting (GREET), politeness (POLIT), interrogative (INTROG), exclamatory (EXCL), incidence (INCID).

TREEOPS—the tool described in the previous section and used to convert UAIC annotation into UD—was also used to convert the initial UAIC format into the semantic one. In this transformation, the dependency connections remained intact, but each of the 44 syntactic labels had to be replaced with one of the 96 semantic labels. Twenty-one syntactic relations were semantically monovalent; each of them has exactly one corresponding semantic label. The others are semantically ambiguous, and specific rules of transformation have been created for them. For example, the syntactic tag *auxiliary* (AUX) can be transposed into OPTV or PAST, if the form of the auxiliary (and of the conjugate verb) indicates the conditional mode or the past tense, or if the auxiliary indicates the future tense (FTR) or passive (PASS).

Format	Number of sentences	Number of tokens
UAIC syntactic (RoDia)	32,753	671,235
UD syntactic (UD Non-Standard)	16,936	348,562
UAIC semantic	5,566	99,341

Table 6: Size of three treebank formats

Syntactic relations that are labeled according to morphological criteria—such as *a.subst.* (noun attribute), *a.vb.* (verbal attribute), and *c.prep.* (prepositional complement)—have high semantic ambiguity; they can have almost any semantic value. The automatic tool leaves such labels untouched, and they are edited

manually. As a result, the part of the corpus that is semantically labeled remains relatively small, and its expansion requires manual work by linguists. When a large enough portion of the corpus has undergone this type of annotation, machine learning parsers trained on it are able to produce semantically labelled text with high accuracy and manual work will be reduced considerably (Colhon *et al.* 2017).

4.5. Automate Parsing Evaluation

The RoDia Dependency TreeBank and Nonstandard Universal Dependencies corpus were created using the bootstrapping approach, meaning that, after undergoing automated annotation, which involved various applications, the treebank and corpus were manually verified and corrected. The sizes of each of the verified and corrected corpora are presented in *Table 6*. The corpora annotated syntactically are large enough to be used for machine learning systems training. In this section, we report on results from the parsing experiments we conducted using MaltParser, one of the tools used in the UD project for automated dependency parsing.

Book Entry	**Sub-corpus**	**# of sentences**	**Words per sentence**	**UAIC UAS**	**UAIC LAS**	**UD UAS**	**UD LAS**
1	apostles	5,894	20.86	83.76	76.29	87.42	82.99
2	Caragea.Law	1,188	23.95	73.34	65.31	83.62	79.09
3	Flower.gifts	1,088	19.46	83.24	74.72	87.56	83.69
4	Folk.Mol	1,806	18.69	80.04	72.37	86.75	82.22
5	Folk.Rom	695	23.61	71.85	64.5	86.08	78.29
6	gospel	5,172	17.86	85.11	78.66	84.06	76.47
Average			20.74	**79.56**	**71.98**	**85.92**	**80.46**

Table 7: The accuracy of automated parsing on different parts of the UAIC RoDia corpus and for the UD format of the same parts of the corpus

MaltParser implements a transition-based approach to dependency parsing, which means that it consists of a transition system for deriving dependency trees and a classifier for deterministically predicting the next transition based on a feature representation of the current parser configuration (Nivre *et al.* 2007). MaltParser is a complex system with many parameters that need to be optimized and adapted to the task at hand. *Table 7* presents the obtained accuracy for several parts of the UAIC syntactic corpus when various MaltParser algorithms are used. Parsing accuracy is

measured with two classical metrics. On the one hand, the Labeled Attachment Score (LAS) reflects the ratio between the number of correct attachments with correct labels and the total number of attachments. On the other hand, the Unlabeled Attachment Score (UAS) indicates the ratio between the number of correct attachments and the total number of attachments, where labels are not taken into consideration. The last two columns of *Table 7* present the same metrics for the UD format of the same corpus parts.

As the table shows, automated parsing for the UD format is better than for its UAIC counterpart. This is probably because MaltParser was created for the UD project and is better able to detect UD principles than to recognize those of the UAIC mode of dependency attachments.

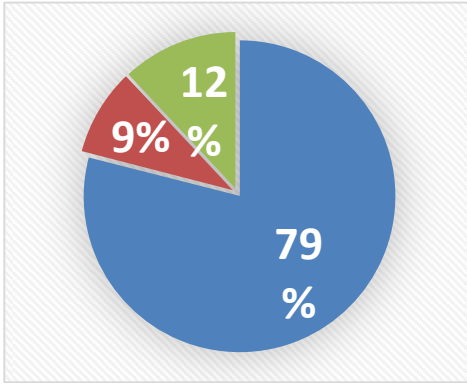
5. Conclusion

Our paper introduces a digitization platform[17] that offers access to a comprehensive set of tools and informational resources, specifically designed for digitizing Romanian documents printed in the Cyrillic script spanning the 17th to 20th centuries. We also take pride in our significant contribution to the development and enrichment of an annotated Romanian corpus, encompassing a diverse range of old texts, regional folklore, and nonstandard texts from Moldova and Romania.

The digitization of these old documents presents captivating scientific challenges, and since Natural Language Processing (NLP) for such texts is typically the purview of isolated and scattered research groups, the resulting products often lack uniformity in terms of formats and standards. Through our ongoing efforts, we are dedicated to digitizating old Romanian texts, facilitating their transliteration into modern Romanian (Latin) alphabet, enhancing optical character recognition, adding annotations, and ultimately incorporating these old texts into the Universal Dependency (UD) repository, adhering to a common standard (Bobicev *et al.* 2019). This initiative not only preserves a valuable piece of cultural heritage but also fosters collaboration and cohesion in the field of NLP, ensuring that a wider community of scholars and enthusiasts have access to these resources.

[17] https://digitizare.math.md.

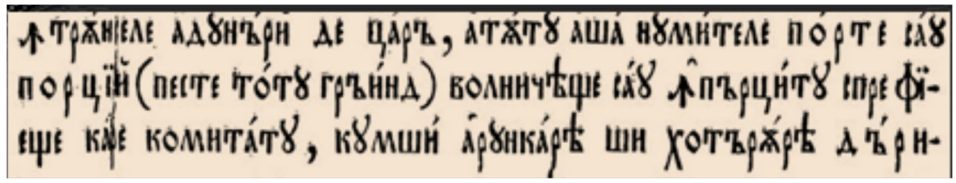
Annex 1: Figures



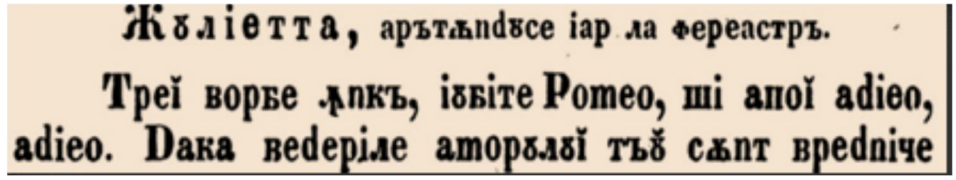
79% Romanian
9% Greek
12% Other (Bulgarian, English, French, German, etc.)

Figure 1: Statistics of prints from the years 1508-1830 at the Library of the Romanian Academy.

Sample of 18th century typographic script



Sample of 19th century typographic script (mixed/ transitional alphabet)



Sample of 20th century typographic script

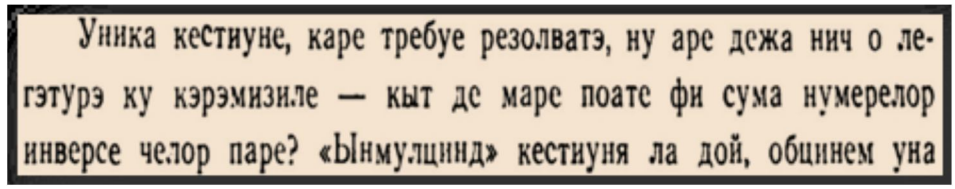


Figure 2: Samples of texts from different centuries.

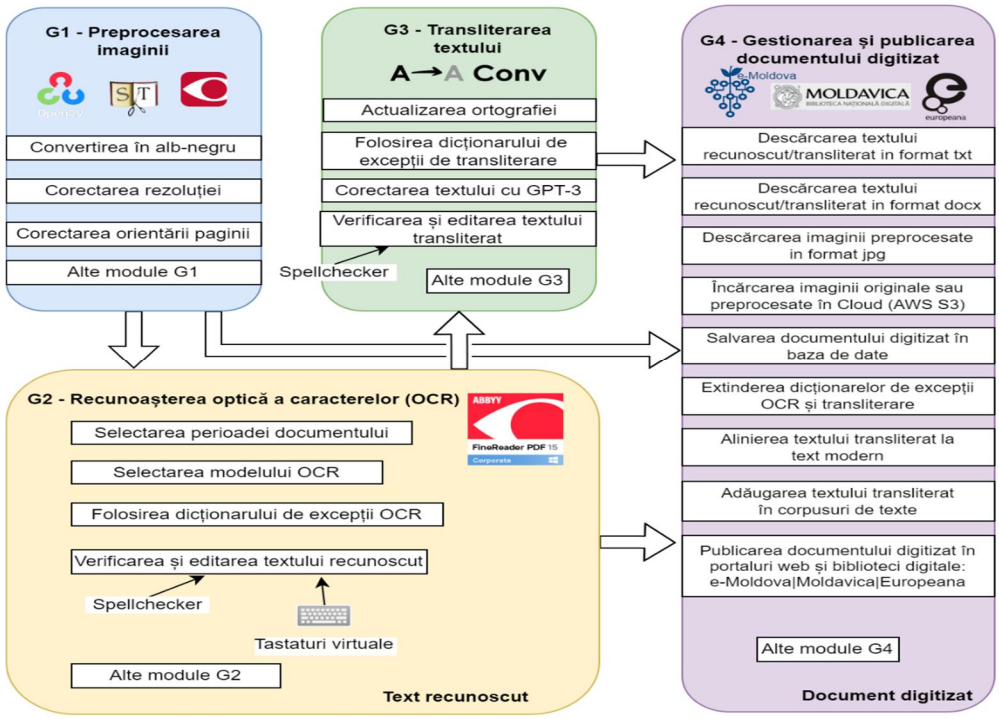
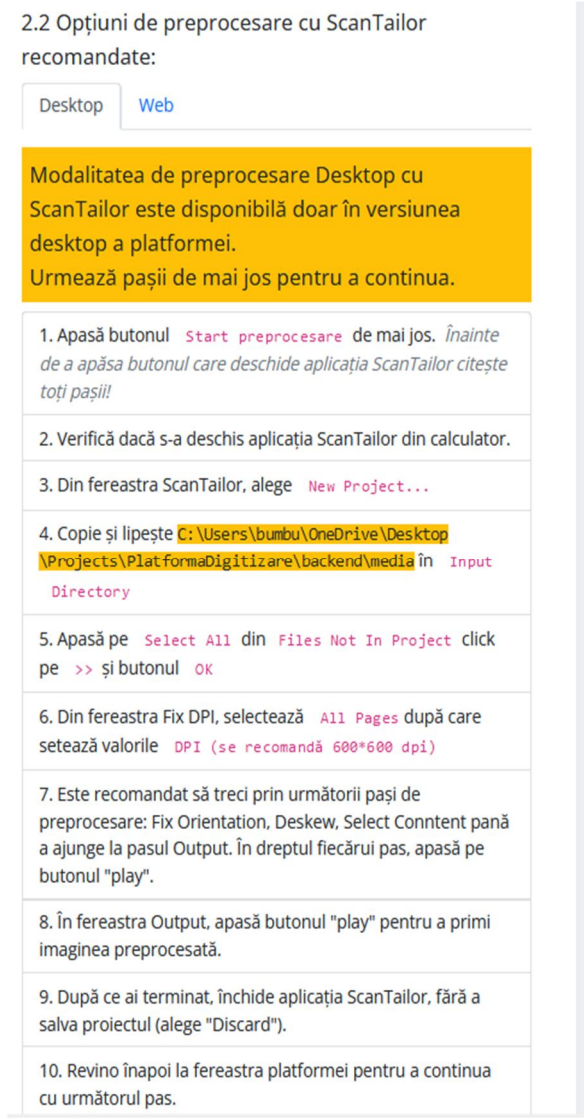


Figure 3: The architecture of Digitization Platform (DP)


2.2 Opțiuni de preprocesare cu ScanTailor recomandate:
Desktop
Web
Modalitatea de preprocesare Desktop cu ScanTailor este disponibilă doar în versiunea desktop a platformei.
Urmează pașii de mai jos pentru a continua.
1. Apasă butonul Start preprocesare de mai jos. Înainte de a apăsa butonul care deschide aplicația ScanTailor citește toți pașii!
2. Verifică dacă s-a deschis aplicația ScanTailor din calculator.
3. Din fereastra ScanTailor, alege New Project...
4. Copie și lipește C:\Users\bumbu\OneDrive\Desktop\Projects\PlatformaDigitizare\backend\media în Input Directory
5. Apasă pe Select All din Files Not In Project click pe >> și butonul OK
6. Din fereastra Fix DPI, selectează All Pages după care setează valorile DPI (se recomandă 600*600 dpi)
7. Este recomandat să treci prin următorii pași de preprocesare: Fix Orientation, Deskew, Select Conntent pană a ajunge la pasul Output. În dreptul fiecărui pas, apasă pe butonul "play".
8. În fereastra Output, apasă butonul "play" pentru a primi imaginea preprocesată.
9. După ce ai terminat, închide aplicația ScanTailor, fără a salva proiectul (alege "Discard").
10. Revino înapoi la fereastra platformei pentru a continua cu următorul pas.

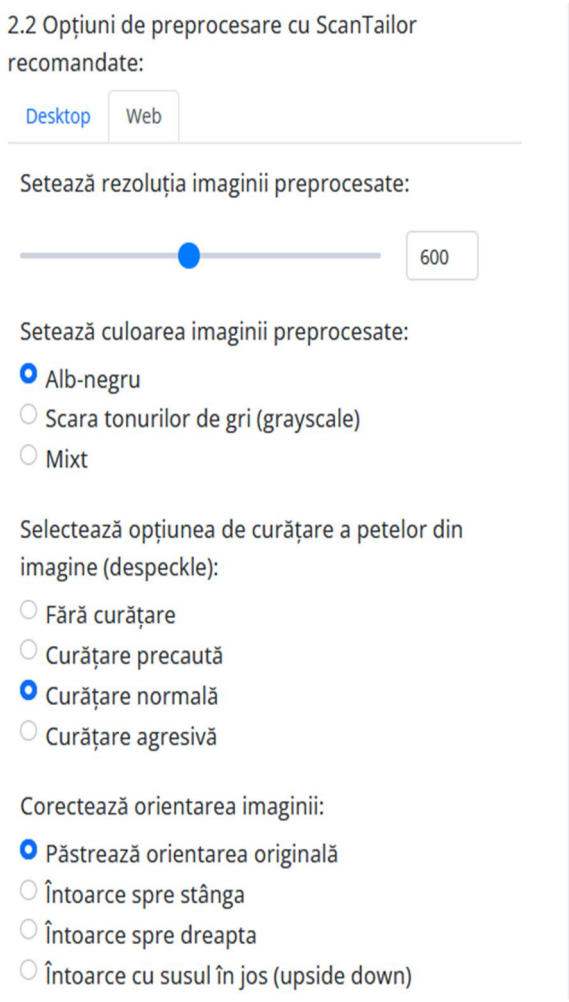
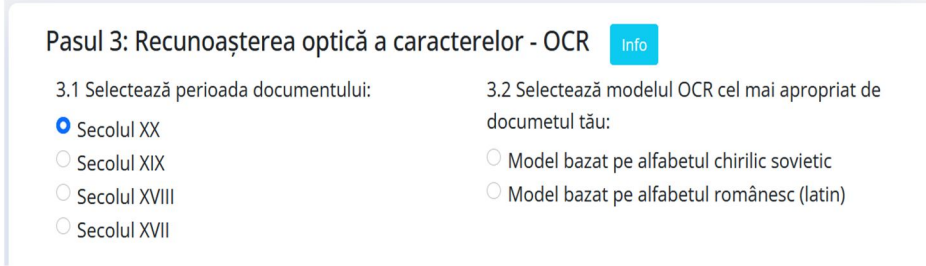


Figure 4: Two ways of preprocessing with Scan Tailor: desktop preprocessing (left), web preprocessing options (right).



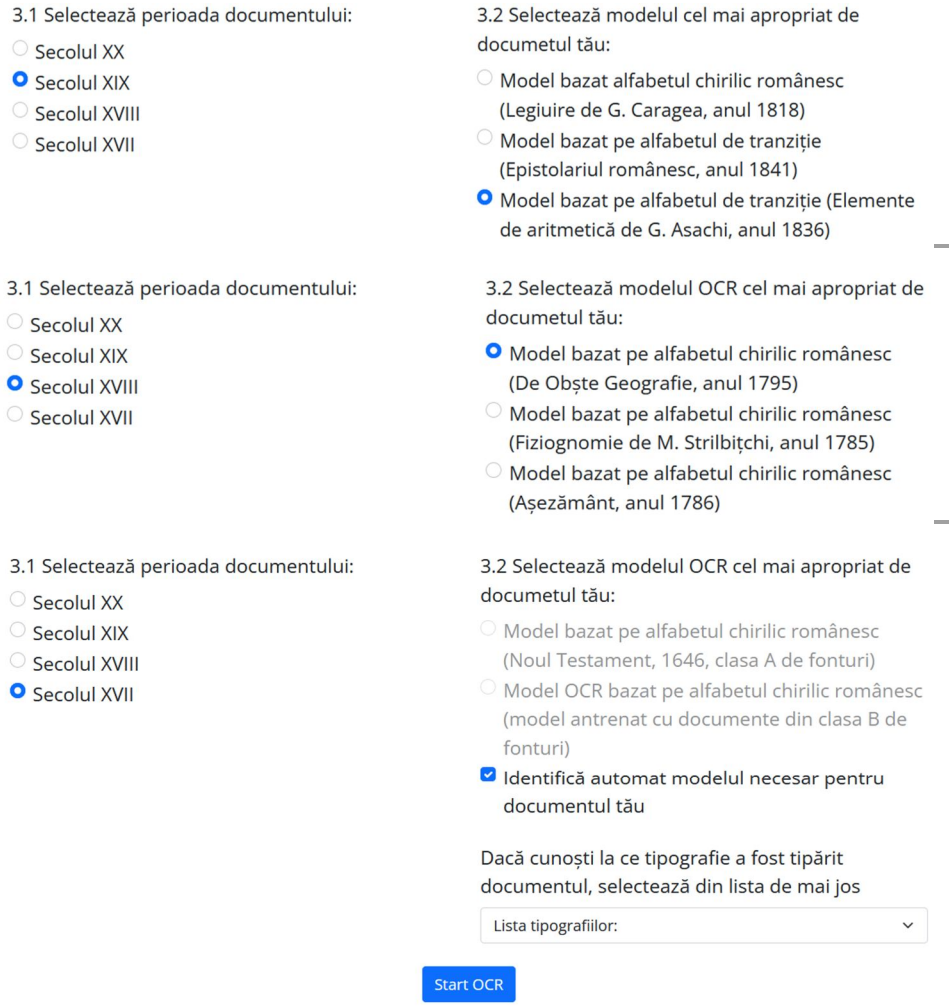


Figure 5: Options available in step 3.

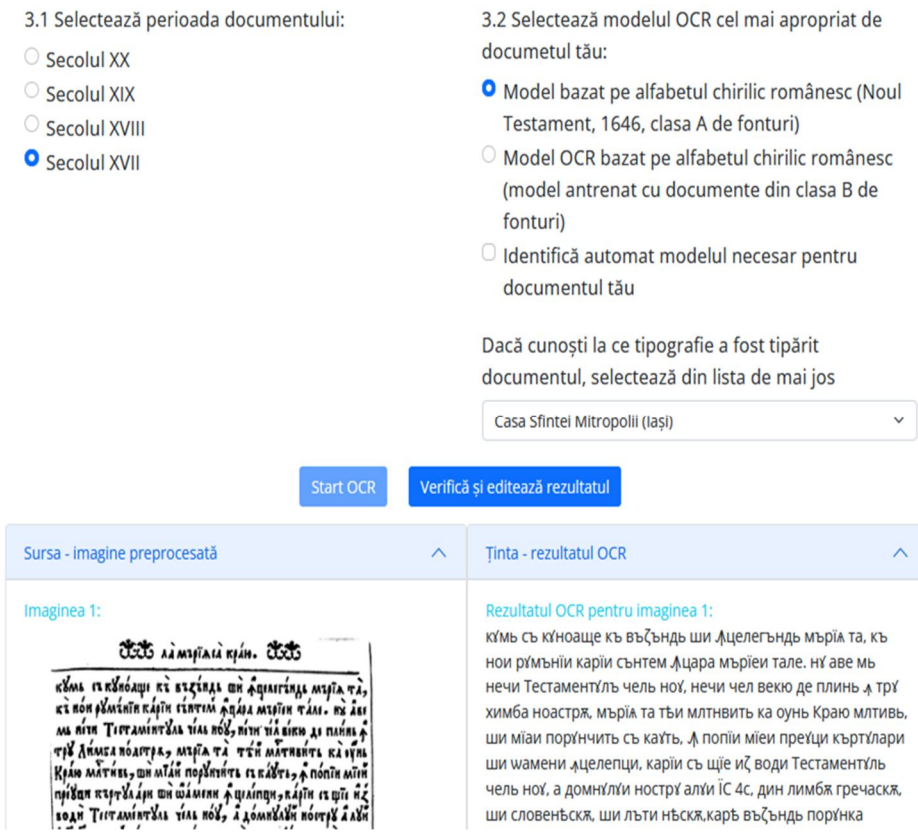


Figure 6: Running step 3 in the digitizing application.

Figure 7: Verification and editing of the recognized text: recognized text (left), processed image (right), virtual keyboard (bottom).

кѵ́мь съ кѵ́ноаще къ въ̀зьндь ши ѧ̀целегъндь мъріѧ̀ та, къ нои рѵ́мънїи карїи сънтем ѧ̀цара мъріеи тале. нѵ̀ аве мь нечи Тестаментѵ́лъ чель ноѵ̀, нечи чел векю де плинь ѧ̀ трѵ̀ химба ноастрѫ̀, мъріѧ̀ та тѣи млтнвить ка оунь Краю млтивь, ши мїаи	cum să cunoaște că văzând și Înțelegând măria ta, că noi rumânii carii sântem în țara măriei tale. nu ave m neci Testamentul cel nou, neci cel vechiu de plin întru limba noastră, măria ta te-ai milostivit ca un Craiu milostiv, și mi-ai poruncit să caut, în popii miei preuți cărtulari și

Figure 8: Transliteration of a recognized text (Cyrillic on the left side and Latin on the right side).

Figure 9: Showing the original document (left), the recognized text (middle), and the transliterated text (right).

```
<POS_Output>
 <S id="1" offset="0">
  <W Case="direct" Definiteness="no" Gender="masculine"
LEMMA="bun" MSD="Afpmsrn" Number="singular"
POS="ADJECTIVE" id="1.1" offset="0">Bun</W>
  <W Case="direct" Definiteness="no" Gender="masculine"
LEMMA="cal" MSD="Ncmsrn" Number="singular" POS="NOUN"
Type="common" id="1.2" offset="4">cal</W>
  <W EXTRA="tranzitiv" LEMMA="hrăni" MSD="Vmip3s"
Mood="indicative" Number="singular" POS="VERB" Person="third"
```

Figure 10: XML format of the tagger (W=Word, MSD=MorphoSyntactic Description). The initial word form, the lemma, all morphological features of the word are coded as separate attributes of the tag W.

```
<sentence id="20" parser="" user="ugla" date="2014-14-15">
  <word id="1" form="Era" lemma="fi" postag="Vmii3s" head="0" />
  <word id="2" form="una" lemma="unul" postag="Pi3fsr" head="1"
deprel="sbj."/>
  <word id="3" form="şi" lemma="şi" postag="Rg" head="4" deprel="a.adv."/>
  <word id="4" form="pe" lemma="pe" postag="Spsa" head="1"
deprel="c.c.l."/>
  <word id="5" form="faţada" lemma="faţadă" postag="Ncfsry"
```

Figure 11: An example of the XML format of RoDia.
Each word is presented in a tag "word" that contains the word and all its annotated attributes.

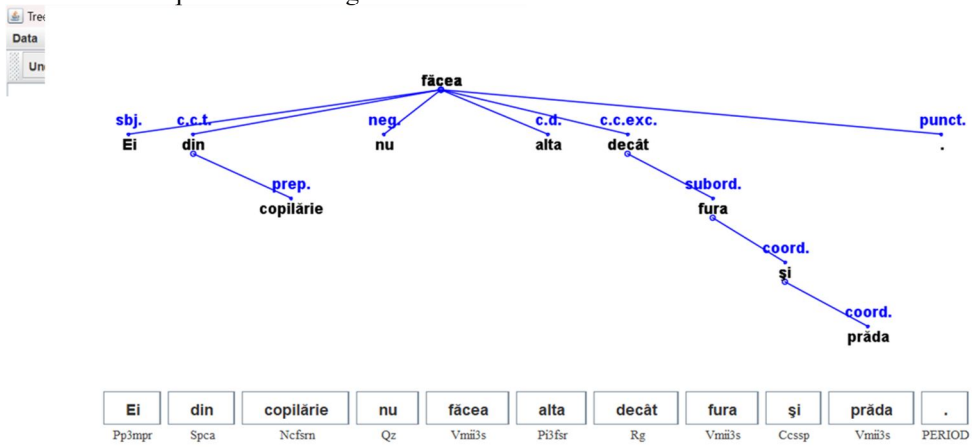


Figure 12: TreeAnnotator interface.

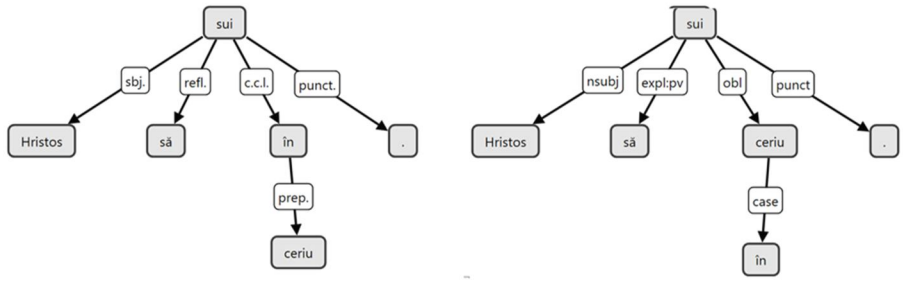


Figure 13: Comparison of UAIC (left) and UD (right) connection principles.
The sentence is: *Hristos să sui in ceriu* [Hristos ascends to heaven]

References

Bănărescu, Laura, Claire Bonial, Shu Cai, Mădălina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philip Koehn, Martha Palmer, and Nathan Schneider. 2013. "Abstract Meaning Representation for Sembanking." In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186.

Bobicev, Victoria, Maranduc Cătălina, Tudor Bumbu, Ludmila Malahov, Alexandru Colesnicov, and Svetlana Cojocaru. 2019. "Contribution to the Universal Dependencies Treebank of Non-Standard Romanian Texts." In *Proceedings of the Language Technologies for All (LT4All)*, Paris, UNESCO Headquarters, France, 5–6 December, 300–303.

Bohmová, Alena, Jan Hajič, Eva Hajičova, and Barbora Hladka. 2003. "The Prague Dependency Treebank: A Three-Level Annotation Scenario. Text, Speech and Language Technology." Berlin and New York: Springer.

Boian, Elena, Constantin Ciubotaru, Svetlana Cojocaru, Alexandru Colesnicov, and Ludmila Malahov. 2014. "Digitizarea, Recunoaşterea și Conservarea Patrimoniului Cultural-Istoric." *Akademos* 1(32): 61–68.

Bumbu, Tudor. 2021. "Towards a Font Classification Model for Romanian Cyrillic Documents." *Computer Science Journal of Moldova* 3(87): 291–298.

Bumbu, Tudor. 2023. "Tehnologii și Resurse Informaționale pentru Digitizarea și Procesarea Textelor din Patrimoniul Historico-Cultural." PhD diss., State University of Moldova.

Caftanatov, Olesea, Tudor Bumbu, Lucia Erhan, Iulian Cernei, Veronica Iamandi, Vasile Lupan, Daniela Caganovschi, and Mihail Curmei. 2021. "Discover the Moldovan Cultural Heritage through e-Moldova Portal by Using Crowdsourcing Concept." In *Proceedings of the Workshop on Intelligent Information Systems (WIIS2021)*, Chișinau, Republic of Moldova, October 14–15, 65–75.

Cazimir, Ștefan. 2006. *Alfabetul de Tranziție*. Bucharest: Humanitas.

Ciubotaru, Constantin, Svetlana Cojocaru, Alexandru Colesnicov, Valentina Demidov, and Ludmila Malahov. 2015. "Regeneration of Cultural Heritage: Problems Related to Moldavian Cyrillic Alphabet." In *Proceedings of International Conference "Linguistic Resources and Tools for Processing the Romanian Language"*, 177–184.

Cojocaru, Svetlana, Alexandru Colesnicov, and Ludmila Malahov. 2017. "Digitization of Old Romanian Texts Printed in the Cyrillic Script." In *ACM International Conference Proceeding Series*, 143–148.

Colesnicov, Alexandru, Ludmila Malahov, Svetlana Cojocaru, and Ludmila Burtseva. 2021. "Digitization Technology of Old Romanian Documents Printed in the Cyrillic Script." In *Horizons in Computer Science Research*. Volume 21, edited by Thomas S. Clary, 185–217. New York: Nova Science Publishers.

Colhon, Mihaela, Cătălina Mărănduc, and Cătălin Mititelu. 2017. “A Multiform Balanced Dependency Treebank for Romanian.” In *Proceedings of Knowledge Resources for the Socio-Economic Sciences and Humanities*, Varna, Bulgaria, 9–18.

Cristea, Dan, Daniela Gîfu, Svetlana Cojocaru, Alexandru Colesnicov, Ludmila Malahov, Mihai Popescu, Mihaela Onofrei, and Cecilia Bolea. 2018. “How to Find Out Whether the Romanian Language Was Influenced by the Two Historical Unions?” In *Proceedings of 13th International Conference on Linguistic Resources and Tools for Processing Romanian Language*, Iași, November 22–23, 77–87.

Erjavec, Tomaž. 2012. “MULTEXT-East: Morphosyntactic Resources for Central and Eastern European Languages.” *Language Resources and Evaluation* 46 (1): 131–142. Berlin and New York: Springer.

Florea, Iulia Maria, Traian Rebedea, and Costin-Gabriel Chiru. 2014. “Parser de Dependenţe pentru Limba Română Realizat pe Baza Parserelor pentru Alte Limbi Romanice.” *Revista Română de Interacţiune Om-Calculator* 7 (1): 1–20.

Gheţie, Ion, and Alexandru Mareş (eds). 1996. *Cele mai Vechi Cărţi Populare în Literatura Română. Vol. 1: Floarea Darurilor. Sindipa.* Bucharest: Minerva Publishing House.

Mărănduc, Cătălina. 2022. “Instruments for Processing the Old Romanian Language (Training Corpora, Models, Statistics).” PhD diss., “Alexandru Ioan Cuza” University of Iași, Romania.

Mărănduc, Cătălina, F. Hociung, and V. Bobicev. 2017. “Treebank Annotator for Multiple Formats and Conventions.” In *The 4th Conference of Mathematical Society of the Republic of Moldova (CMSM4)*.

Mărănduc, Cătălina, Monica Mihaela Rizea, and Dan Cristea. 2017. “Mapping Dependency Relations onto Semantic Categories.” In *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing (CICLing)*.

Marneffe, Marie-Catherine, Christopher Manning, Joakim Nivre, and Daniel Zeman. 2021. “Universal Dependencies.” *Computational Linguistics* 47 (2): 255–308.

Moruz, Alexandru. 2008. “Dezvoltarea unui Adnotator FDG (Functional Dependency Grammar) pentru Limba Română.” Master’s diss., Facultatea de Informatică, Universitatea „Alexandru Ioan Cuza” Iaşi.

Nivre, Joakim, Johan Hall, Jens Nilsson, and Atanas Chanev. 2007. “MaltParser: A Language-Independent System for Data-Driven Dependency Parsing.” *Natural Language Engineering* 13: 95–135. Cambridge: Cambridge University Press.

Perez, Cenel-Augusto. 2014. “Linguistic Resources for Natural Language Processing.” PhD diss., Al. I. Cuza University, Iași.

Simionescu, Radu. 2011. “Hybrid POS Tagger.” In *Workshop on Language Resources and Tools in Industrial Applications, Eurolan 2011 Summer School*.

Tufiș, Dan. 2000. “Using a Large Set of EAGLES-Compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging.” *LREC*.