

**PROCEEDINGS OF THE 17th  
INTERNATIONAL CONFERENCE  
“LINGUISTIC RESOURCES AND TOOLS  
FOR NATURAL LANGUAGE PROCESSING”**

**CHIȘINĂU AND ONLINE  
10-12 NOVEMBER 2022**

*Editors:*

Svetlana Cojocaru  
Victoria Bobicev  
Tatiana Verlan  
Dan Tufiş  
Dan Cristea

*Organisers*

Academy of Sciences of Moldova

Vladimir Andrunachievici Institute of Mathematics and Computer Science,  
Chișinău

Technical University of Moldova,  
Chișinău

Institute of Computer Science, Iași branch of Romanian Academy

Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy,  
Bucharest

“A. Philippide” Institute of Romanian Philology, Iași branch of Romanian Academy

“Alexandru Ioan Cuza” University of Iași,  
Faculty of Computer Science

Romanian Association of Computational Linguistics

The event was organized under the auspices of the Technical Sciences Academy of Romania and  
was dedicated to Ukrainian scientists.

The publication of this volume was supported by

The Faculty of Computer Science,

“Alexandru Ioan Cuza” University of Iași

ISSN 1843-911X

## SCIENTIFIC COMMITTEE

**Victor Ababii** – Technical University of Moldova, Chişinău  
**Verginica Barbu Mititelu** – Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest  
**Victoria Bobicev** – Technical University of Moldova, Chişinău  
**Natalia Burlacu** – Technical University of Moldova, Chişinău  
**Liudmila Burţeva** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău  
**Constantin Ciubotaru** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău  
**Alexandru Colesnicov** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău  
**Mihaela Colhon** – University of Craiova, Faculty of Mathematics and Natural Sciences  
**Svetlana Cojocaru** – Academy of Sciences of Moldova & Vladimir Andrunachievici Institute of Mathematics and Computer Sciences  
**Eugeniu Costetchi** – Meaningfy & University of Bremen  
**Dan Cristea** – Romanian Academy, Institute of Computer Science, Iaşi branch & “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science  
**Iulia Hlavceva** – Kharkiv Polytechnic Institute  
**Adrian Iftene** – “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science  
**Diana Zaiu Inkpen** – University of Ottawa, School of Electrical Engineering and Computer Science, Canada  
**Radu Ion** – Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest  
**Elena Irimia** – Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest  
**Olga Kanishcheva** – Kharkiv Polytechnic Institute  
**Maria Mitrofan** – Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest  
**Mihai Alex Moruz** – “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science  
**Constantin Orasan** – University of Surrey  
**Vasile Florian Păiş** – Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest  
**Mircea Petic** – Alecu Russo Balti State University, Republic of Moldova & Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău  
**Jatinderkumar R. Saini** – Symbiosis Institute of Computer Studies and Research, Pune, India  
**Elena Isabelle Tamba** – “A. Philippide” Institute of Romanian Philology  
**Horia-Nicolai Teodorescu** – Romanian Academy, Institute of Computer Science, Iaşi branch & “Gheorghe Asachi” Technical University of Iaşi, Faculty of Electronics, Telecommunications, and Information Technology  
**Diana Trandabăţ** – “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science  
**Ştefan Trăuşan-Matu** – Politehnica University of Bucharest, Faculty of Automation, Control, and Computer Engineering & Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest  
**Dan Tufiş** – Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest  
**Tatiana Verlan** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

## ORGANIZING COMMITTEE

**Victoria Bobicev** – Technical University of Moldova, Chişinău, Faculty of Computers, Informatics, and Microelectronics

**Tudor Bumbu** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

**Olesea Caftanaton** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

**Svetlana Cojocar** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

**Dan Cristea** – Romanian Academy, Institute of Computer Science, Iaşi branch & “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science

**Valentina Demidov** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

**Lucian Gâdioi** – “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science

**Daniela Gifu** – Romanian Academy, Institute of Computer Science, Iaşi branch & “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science

**Adrian Iftene** – “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science

**Galina Magariu** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

**Ludmila Malahov** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

**Andrei Scutelnicu** – Romanian Academy, Institute of Computer Science, Iaşi branch & “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science

**Diana Trandabăţ** – “Alexandru Ioan Cuza” University of Iaşi, Faculty of Computer Science

**Dan Tufiş** – Institute for Artificial Intelligence “Mihai Drăgănescu”, Bucharest

**Inga Țițchiev** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

**Tatiana Verlan** – Vladimir Andrunachievici Institute of Mathematics and Computer Science, Chişinău

# TABLE OF CONTENTS

|                                                                                                                         |            |
|-------------------------------------------------------------------------------------------------------------------------|------------|
| <b>FOREWORD.....</b>                                                                                                    | <b>VII</b> |
| <b>CONFERENCE KEYNOTE SPEECHES.....</b>                                                                                 | <b>1</b>   |
| REGIONAL ANNOTATION WITHIN GRAC, A LARGE REFERENCE CORPUS OF UKRAINIAN<br><i>Mariia Shvedova</i>                        | 1          |
| WITH A LITTLE HELP FROM NLP: MY LANGUAGE TECHNOLOGY APPLICATIONS WITH<br>IMPACT ON SOCIETY.....<br><i>Ruslan Mitkov</i> | 1          |
| PARLAMINT: MULTILINGUAL COMPARABLE CORPORA OF PARLIAMENTARY DEBATES...<br><i>Maciej Ogrodniczuk</i>                     | 2          |
| SCALES IN THE LEXICON: IDENTIFICATION, CONSTRUCTION, APPLICATIONS.....<br><i>Christiane Fellbaum</i>                    | 2          |
| WHAT MACHINE TRANSLATION TELLS US ABOUT SENTIMENTS IN OTHER<br>LANGUAGES.....<br><i>Constantin Orăsan</i>               | 3          |
| <b>DELOORO WORKSHOP KEYNOTE SPEECHES.....</b>                                                                           | <b>4</b>   |
| HeDY – A PLATFORM FOR HETEROGENEOUS DOCUMENT PROCESSING.....<br><i>Svetlana Cojocaru</i>                                | 4          |
| <b>CHAPTER 1. CORPORA CREATION AND ANNOTATION.....</b>                                                                  | <b>5</b>   |
| TERMINOLOGY ANNOTATION USING A STRING MATCHING ALGORITHM .....<br><i>Andrei-Marius Avram, Vasile Păiș, Dan Tufiș</i>    | 7          |
| FINE-GRAINED LEGAL REFERENCES IN THE LEGALNERO CORPUS.....<br><i>Daria Costea, Andrei Moldovan, Vasile Păiș</i>         | 15         |
| <b>CHAPTER 2. LANGUAGE PROCESSING TOOLS.....</b>                                                                        | <b>25</b>  |
| AI-BASED SOLUTIONS FOR TEXT LANGUAGE IDENTIFICATION.....<br><i>Radu A. Ciora, Daniela Gîfu</i>                          | 27         |
| CORRECTING POSTAL ADDRESSES WRITTEN IN ANY LANGUAGE.....<br><i>Georgiana Druță, Adrian Ifțene, Daniela Dănciulescu</i>  | 35         |
| SPLITTING HASHTAGS IN ROMANIAN MICRO-BLOGGING TEXTS.....<br><i>Roxana Micu, Carol Luca Gasan, Vasile Păiș</i>           | 49         |
| PERSONALITY PREDICTION BASED ON SOCIAL MEDIA POSTS.....<br><i>Bianca Ionela Strățianu, Adrian Ifțene</i>                | 59         |
| <b>CHAPTER 3. ROMANIAN LINGUISTIC RESOURCES.....</b>                                                                    | <b>71</b>  |
| MEDICAL CORPUS FROM ROMANIAN HOSPITALS.....<br><i>Rareș Arvinte, Diana Trandabăț</i>                                    | 73         |
| GIS-BASED PUBLISHING SYSTEM FOR THE ROMANIAN LINGUISTIC ATLAS BY REGIONS<br>– SYNTHESIS.....                            | 81         |

*Silviu-Ioan Bejinariu, Ramona Luca, Florin Iftene, Manuela Nevaci,  
Carmen Irina Floarea, Nicolae Saramandu*

|                                                                                     |    |
|-------------------------------------------------------------------------------------|----|
| PAREMCOR – CONTEMPORARY ROMANIAN LANGUAGE CORPUS OF PROVERBS AND ANTI-PROVERBS..... | 93 |
| <i>Anca-Diana Bibiri, Maria Moruz, Mihai Alex Moruz</i>                             |    |

#### **CHAPTER 4. DIGITAL HERITAGE..... 103**

|                                                                                                                                                                 |     |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|
| THE RECURRENCE OF THE 16 <sup>TH</sup> CENTURY ROMANIAN PSALTERS VOCABULARY IN THE 17 <sup>TH</sup> CENTURY LEXICONS BASED ON AUTOMATIC CORPORA COMPARISON..... | 105 |
| <i>Ana-Maria Gînsac, Mihai Alex Moruz, Mădălina Ungureanu</i>                                                                                                   |     |

|                                                                            |     |
|----------------------------------------------------------------------------|-----|
| APPLYING YOLOV5 LEARNING IN DETECTING OLDCYRILLIC-ROMANIAN CHARACTERS..... | 115 |
| <i>Dan Cristea, Petru Rebeja, Cristian Pădurariu</i>                       |     |

|                                                                     |     |
|---------------------------------------------------------------------|-----|
| EVALUATION OF PHILOLOGICAL RESOURCES IN THE ONLINE ENVIRONMENT..... | 123 |
| <i>Eugenia Mincu</i>                                                |     |

|                                                                                                                       |     |
|-----------------------------------------------------------------------------------------------------------------------|-----|
| THE FIRST SLAVONIC-ROMANIAN LEXICONS (17 <sup>TH</sup> CENTURY): A DIGITAL COMPARATIVE EDITION AND HOW TO USE IT..... | 131 |
| <i>Mihai Alex Moruz, Mădălina Ungureanu</i>                                                                           |     |

#### **CHAPTER 5. SEMANTICS AND PRAGMATICS..... 139**

|                                                                                                |     |
|------------------------------------------------------------------------------------------------|-----|
| A PLATFORM FOR PROCESSING HETEROGENEOUS DOCUMENTS.....                                         | 141 |
| <i>Tudor Bumbu, Lyudmila Burţeva, Svetlana Cojocaru, Alexandru Colesnicov, Ludmila Malahov</i> |     |

|                                                  |     |
|--------------------------------------------------|-----|
| HOW TO CREATE AN ADAPTIVE E-LEARNING SYSTEM..... | 153 |
| <i>Alexandr Parahonco, Mircea Petic</i>          |     |

|                                                                                                                      |     |
|----------------------------------------------------------------------------------------------------------------------|-----|
| ONTOLOGICAL STRUCTURING OF NATURAL LANGUAGE CONTENT IN ENCYCLOPEDIA WIKI-PORTALS AS A BASIS FOR SEMANTIC SEARCH..... | 161 |
| <i>Julia Rogushina</i>                                                                                               |     |

#### **INDEX OF AUTHORS..... 171**

## FOREWORD

The volume includes the papers of the 17th edition of the conference "Linguistic Resources and Tools for Natural Language Processing", which took place this year in Chisinău, Republic of Moldova. After two online editions, caused by the restrictions of the COVID-19 pandemic, we finally had the opportunity to take a step towards normality, using a mixed format (face-to-face and online). In the same framework, the second DeLORo workshop "Artificial Intelligence Models (Deep Learning) Applied in the Analysis of Old Romanian Language" was held.

This year's event - the first organized outside Romania - was dedicated to Ukrainian researchers, in appreciation of their country's heroism in the fight against the Russian aggressors.

The conference was held under the aegis of two Academies – the Romanian Academy and the Academy of Sciences of Moldova - and under the traditional auspices of the Romanian Academy of Technical Sciences. In the first case, the responsibility was assumed by the Consortium for Informatization of Romanian Language (Consortiul de Informatizare pentru Limba Română, whose acronym – ConsILR precedes the name of the conference), which operates within the Information Sciences and Technologies Section, in the second - the Exact and Engineering Sciences Section. The ConsILR conferences series aims to bring together researchers in computational linguistics, computer scientists, linguists, as well as researchers in other humanities, equally being also addressed to Ph.D. students, masters, and students in all these fields.

The host organizers were the Institute of Mathematics and Computer Science "Vladimir Andrunachievici" and the Technical University of Moldova, who worked in close collaboration with colleagues from the Faculty of Computer Science of the "A.I. Cuza" University in Iasi, the Research Institute for Artificial Intelligence "Mihai Drăgănescu" of the Romanian Academy, Institute of Computer Science (Iasi Branch of the Romanian Academy), "A. Philippide" Institute of Romanian Philology (Iasi Branch of the Romanian Academy), and the Romanian Association of Computational Linguistics. An important contribution was made by the members of the scientific committee from six countries, thanks to whose efforts the quality of the papers was substantially improved, and we take this opportunity to thank them for their assistance.

The conference program included 16 papers, grouped into 5 sections, which are also reflected in the structure of this volume:

1. Corpora Creation and Annotation. The papers in this section describe specific types of corpus annotation and experiments with various algorithms for automation of annotation.
2. Language Processing Tools. Solutions for various particular tasks, such as text correction and identification of specific features of text or of its author are presented in this chapter.

3. Romanian Linguistic Resources. The papers in this section are concentrated on distinct aspects of the Romanian language, namely, collections of regional particularities, processing of medical texts, proverbs, and their contemporary usage.

4. Digital Heritage. In this section, several diachronic lexicons have been discussed as well as the problem of old Romanian texts written in Cyrillic characters.

5. Semantics and Pragmatics. This chapter includes studies addressing human-computer interaction related to knowledge representation and e-learning.

Five Keynote Speeches were presented during the conference, namely:

– Maria Shvedova. "Regional Annotation within GRAC, a Large Reference Corpus of Ukrainian" (Lviv Polytechnic National University, Ukraine);

– Ruslan Mitkov. "With a little help from NLP: My Language Technology applications with impact on society" (University of Wolverhampton, UK);

– Maciej Ogrodniczuk. "ParlaMint: Multilingual Comparable Corpora of Parliamentary Debates" (Institute of Computer Science, Polish Academy of Sciences);

– Christiane Fellbaum. "Scales in the Lexicon: Identification, Construction, Applications" (Princeton University, USA);

– Constantin Orăsan. "What machine translation tells us about sentiments in other languages" (University of Surrey, UK).

A keynote speech by Svetlana Cojocaru (Academy of Sciences of Moldova) entitled "HeDy - a platform for heterogeneous document processing" was delivered in the frame of the DeLORo Workshop.

The papers presented at the conference brought new and valuable contributions to the development of the field of natural language processing, many of them being focused on applications for the Romanian language.

November 2022

The Editors

## CONFERENCE KEYNOTE SPEECHES

### REGIONAL ANNOTATION WITHIN GRAC, A LARGE REFERENCE CORPUS OF UKRAINIAN

MARIIA SHVEDOVA

*Lviv Polytechnic National University, Ukraine*  
corpus.textiv@gmail.com

#### Abstract

The General Regionally Annotated Corpus of Ukrainian (GRAC; [uacorporus.org](http://uacorporus.org)) is a general-purpose reference corpus of Ukrainian and as such intended for a wide range of research tasks. In terms of structure, annotation, and metadata it generally follows the model of existing reference corpora such as the BNC. What sets GRAC apart from these corpora is the distinctive feature of regional markup. The need for such markup follows from specific properties of standard Ukrainian: due to its complex history, Ukrainian exhibits significant regional variation which has not yet been systematically investigated on the basis of a large corpus. Taking this variation into account is both a challenge for any comprehensive research into Standard Ukrainian and constitutes an object of inquiry in its own right. In this talk, we present and motivate the principles of regional markup realized within GRAC and discuss issues of territorial representativity. We then present case studies of regional variations of Ukrainian and discuss questions and difficulties that arise in this context.

### WITH A LITTLE HELP FROM NLP: MY LANGUAGE TECHNOLOGY APPLICATIONS WITH IMPACT ON SOCIETY

RUSLAN MITKOV

*Research Institute of Information and Language Processing, University of Wolver-*  
*hampton, UK*  
R.Mitkov@wlv.ac.uk

#### Abstract

The talk will present three original methodologies developed by the speaker, underpinning implemented Language Technology tools that are already having an impact on the following areas of society: e-learning, translation and interpreting, and care for people with language disabilities.

The first part of the presentation will introduce an original methodology and tool for generating multiple-choice tests from electronic textbooks. The application draws on a variety of Natural Language Processing (NLP) techniques which include term extraction, semantic computing, and sentence transformation. The presentation will include an evaluation of the tool which demonstrates that the generation of multiple-choice test items with the help of this tool is almost four times faster than manual construction, and the quality of the test items is not compromised. This application benefits e-learning users (both teachers and students) and is an example of how NLP can have a positive societal impact, which the speaker passionately believes.

The talk will go on to outline two other original recent projects which are also related to the application of NLP beyond academia. First, a project, whose objective is to develop next-generation translation memory tools for translators and, in the near future, for interpreters, will be briefly presented. Finally, an original methodology and system will be outlined which helps users with autism to read and better understand texts.

The presentation will finish with a brief outline of the latest (and forthcoming) research topics (to be) pursued by the speaker.

## **ParlaMint: MULTILINGUAL COMPARABLE CORPORA OF PARLIAMENTARY DEBATES**

MACIEJ OGRODNICZUK

*Institute of Computer Science, Polish Academy of Sciences*  
maciej.ogrodniczuk@gmail.com

### **Abstract**

ParlaMint is a CLARIN ERIC-supported initiative aimed at the harmonization of freely available, comparable, and uniformly annotated multilingual corpora of parliamentary sessions. In 2021 its first version was compiled using data of 17 European parliaments with almost half a billion words. In October 2022, a new release is planned to add newer data and metadata to existing corpora, provide corpora for several more languages, and improve the usability of the infrastructure. The talk will introduce the ParlaMint corpora, standards, and services and demonstrate how this valuable data can be used for further research in Digital Humanities.

## **SCALES IN THE LEXICON: IDENTIFICATION, CONSTRUCTION, APPLICATIONS**

CHRISTIANE FELLBAUM

Joint work with Vera Sheinman, Isaac Julien, Yannick Mathieu, and Pulkit Singh

*Princeton University, USA*  
fellbaum@princeton.edu

### **Abstract**

Many adjectives and verbs represent scalar or intensity values (small-tiny-microscopic; scare-frighten-terrify). Speakers often have an intuitive sense of the relative position of such words on a shared scale, but empirically determining their order is challenging. Focusing on emotion verbs, we report on corpus-based methods for constructing scales in two languages as well as experiments matching human judgments for scalar emotion verbs and emojis. We consider the encoding of scales in WordNet and discuss their usefulness in NLP tasks.

# WHAT MACHINE TRANSLATION TELLS US ABOUT SENTIMENTS IN OTHER LANGUAGES

CONSTANTIN ORĂSAN

*University of Surrey, UK*  
c.orasan@surrey.ac.uk

## **Abstract**

Machine translation engines are commonly used to translate user-generated content on social media sites like Twitter and Facebook. In some cases, the translation happens automatically and the users do not even get the chance to see the original text. However, the automatic translation of sentiments and emotions is difficult especially when the input is noisy, as is the case with user-generated content. This talk will discuss various challenges encountered when translating user-generated content and proposes a number of solutions.

# **DELORO WORKSHOP KEYNOTE SPEECHES**

## **HeDy – A PLATFORM FOR HETEROGENEOUS DOCUMENT PROCESSING**

SVETLANA COJOCARU

Joint work with Tudor Bumbu, Lyudmila Burtseva, Alexandru Colesnicov, and  
Ludmila Malahov

*Institute of Mathematics and Computer Science, Academy of Sciences of Moldova*  
svetlana.cojocaru@math.md

### **Abstract**

The paper discusses the problems of heterogeneous text digitization. The archives of scanned printed documents grow dramatically as the result of projects concerning cultural heritage preservation. Existing document processing systems are not always able to display document content correctly due to the presence of heterogeneous components. We proposed a Web platform to maximize the support of semi-automated work of all used tools for the recognition of heterogeneous documents. For implementation, convergent technology is used, which assembles complex software systems from ready-made heterogeneous modules on a single platform.

## **CHAPTER 1**

### **CORPORA CREATION AND ANNOTATION**



# TERMINOLOGY ANNOTATION USING A STRING MATCHING ALGORITHM

ANDREI-MARIUS AVRAM<sup>1,2</sup>, VASILE PĂIȘ<sup>2</sup>, AND DAN TUFIȘ<sup>2</sup>

<sup>1</sup>*University Politehnica of Bucharest, Bucharest, Romania*

<sup>2</sup>*Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, Bucharest, Romania*

## Abstract

The approximate term matching research is divided into two principal directions: deterministic algorithms that use some distance metrics (e.g. Levenshtein or Hamming distance) to compute the distance between the given strings or statistical algorithms that usually imply training a statistical model that can be used to make predictions at a later point. This work introduces a deterministic term matching algorithm that traverses the input document text with a set of windows and verifies whether any string delimited by the respective windows is found in a dictionary containing pre-defined terminology. We use this algorithm to annotate the Romanian part of the CURLICAT corpus which contains roughly 4.5 million sentences in the CoNLLU Plus format, by adding a new column to each file, representing the IATE IDs terms for the given tokens. Furthermore, the scripts developed in this work were packed into a web service and deployed into the RELATE platform to improve accessibility and ease of usage.

## 1 Introduction

Natural language processing (NLP) is a sub-domain of artificial intelligence that aims at making machines understand and manipulate human languages. One of the main areas of research that NLP focuses on is finding keyword terms in the text that can be later used for many use cases such as document indexing, document search, or document summarization (Litvak and Last 2008; Bao et al. 2009). The state-of-the-art systems for this problem rely on machine learning techniques (Kumar et al. 2022) and usually involve a manually annotated corpus to train on. However, this dataset is usually hard to obtain due to the large number of keywords that have to be identified and thus deterministic string matching algorithms are still used nowadays.

Our paper proposes a deterministic solution to the problem raised in the Curated Multilingual Language Resources for CEF AT (CURLICAT) project (Váradi et al. 2022), where the Romanian corpus had to be annotated with the latest version of the Inter-Agency Terminology Exchange (IATE) terminology (Johnson and Macphail 2000). The proposed algorithm simply traverses the given document using a set of windows that move from one token to the next one, verifying whether any string that is found in one of the windows exists in the IATE terminology. We further compared the keyword terms found by our method with the keywords found by using another work (Coman et al. 2019) and found out that most of the differences come from the different versions that were used in their IATE version. The algorithm developed in this work was also integrated into the RELATE platform (Păiș et al. 2020) and was open-sourced for public usage<sup>1</sup>.

The rest of the paper is organized as follows. Section 2 outlines the related work, while Section 3 presents more details about the CURLICAT project and dataset. Section 4 introduces the deterministic string matching algorithm developed in this work and Section 5 presents the evaluation procedure and the results. Section 6 shows how this algorithm was deployed in the RELATE

---

<sup>1</sup> <https://github.com/racai-ai/TerminologyAnnotator>

platform and how it can be accessed. The paper concludes in Section 7 with potential future work.

## 2 Related Work

The approximate string matching research has been concentrated in two main directions in the previous years: deterministic algorithms that use a distance metric (i.e. Levenshtein or the Hamming distance) to measure the similarity of the strings and probabilistic algorithms that are based on machine learning techniques.

The probabilistic string matching algorithms usually involve training a statistical model for sequence tagging (Goyal et al. 2018; Li et al. 2020) where the model has to learn to classify each input token as being part of a specific term or not. However, this approach involves firstly collecting a manually annotated dataset for terminology on which the model is to be trained, resulting in a prolonged development process. In addition, the classification problem might be infeasible for a probabilistic algorithm because the terminology usually contains a large vocabulary of terms distributed in an uneven way in the corpus, leading to poor performance on uncommon terms.

The deterministic algorithms mitigate these two issues by creating rules to detect the terms in the given terminology. Past research in this area focused on the k-mismatch Levenshtein Automaton, built around the concept of Levenshtein distance (Charalampopoulos et al. 2020) and the Aho-Corasick data structure (Aho and Corasick 1975). Recent work in the rule-based direction was introduced in (Coman et al. 2019) where the authors annotated with IATE and EuroVoc descriptors a Romanian corpus that contained approximately 150k legal documents. The authors created Levenshtein automata using the Aho-Corasick data structure on which a shorter mask for each term in the dictionary was used. The mask was created by employing the following rules: (1) remove all the vowels from the term, (2) keep the punctuation and the spaces found in the term, and (3) keep the capitalization of acronyms.

## 3 The CURLICAT Dataset

The aim of the CURLICAT project was to generate a multilingual dataset that contains the seven languages that participated in the consortium: Bulgarian, Croatian, Hungarian, Polish, Romanian, Slovak, Slovenian, aiming to improve the eTranslation machine translation system in the domains relevant to the European Digital Service Infrastructure (DSI) such as eHealth or eGovernment. When the corpus is completed, it will contain at least 2M sentences for each language, so more than 14M sentences in total with an estimated number of tokens of at least 140M words.

*Table 1. Statistics of the Romanian CURLICAT corpus*

| Domain       | # Sentences      |
|--------------|------------------|
| culture      | 577,307          |
| education    | 320,484          |
| economy      | 311,721          |
| health       | 417,681          |
| nature       | 338,953          |
| politics     | 379,188          |
| science      | 2,113,454        |
| <b>TOTAL</b> | <b>4,458,788</b> |

Apart from the dataset being classified by domain, the text found inside was linguistically annotated following the CoNLL-U Plus format: tokenization, lemmatization, part-of-speech<sup>2</sup>, dependency parsing, and sentence splitting. In addition, the CURLICAT dataset had to be annotated with the latest IATE terminology so that future possible language models built using it can take advantage of the multi-word expressions found inside.

The Romanian component of the CURLICAT corpus used in this work consists of texts that were selected mostly from The Reference Corpus of the Contemporary Romanian Language (CoRoLa) (Mititelu et al. 2018; Tufiş et al. 2019). All the documents were retrieved based on the metadata attributes found in CoRoLa domain metadata scheme. Table 1 outlines the statistics for each relevant domain of the Romanian dataset extracted from CoRoLa for the CURLICAT project.

Because the protection of people in regard to the processing of their personal data is a fundamental right (Spiekermann 2012), text anonymization should be one of the mandatory processing phases for all corpora. Thus, for the Romanian language, an anonymization system was developed (Păiş et al. 2021) using the "local" pseudonymization method. It must be noted that this implementation kept all the suffixes specific to Romanian-named entities as part of the pseudonym since most anonymization requirements are related to either news or blogs. The system was also integrated into the RELATE platform for public usage<sup>3</sup>.

#### 4 Methodology

The approach used in this work to annotate a corpus with a given terminology relies on a rule-based string matching algorithm that finds a term by extracting all the successive lemmatized words sequences and comparing each sequence with the lemmatized terms found in the terminology, annotating the corpus with the corresponding ID of the term (as depicted in Fig. 1).

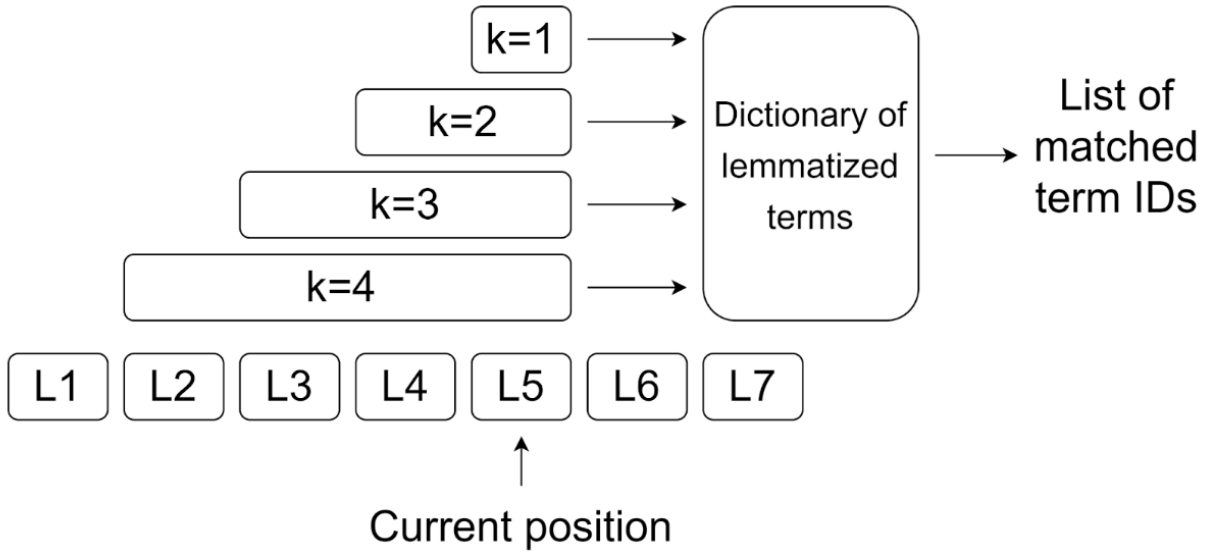


Figure 1. The term annotation algorithm that employs lemma windows of size 1 to K used to probe the dictionary of lemmas and extract the corresponding list of matched term ID

More concretely, we employ the following steps in our method:

1. Lemmatize each term in the terminology and create a dictionary that contains as key the lemmatized words of the term and as value the ID of the term. To lemmatize the words in each term, we used the Romanian Word Form dictionary (Tufiş, 2009).

<sup>2</sup> Both universal part-of-speech (UPOS) and extended part-of-speech(XPOS)

<sup>3</sup> <https://relate.racai.ro/index.php?path=roanon/anonymize>

2. Lemmatize the words in the input document using the same methodology as in the previous step and iterate the sequence, creating, at each index, windows that contain the previous 1 to  $K$  lemmas of the sequence. Then, at each index, verify which windows are found in the dictionary of lemmatized terms, returning the ID of each matching term.
3. Annotate each window in the corpus with the corresponding term ID if it was found in the terminology dictionary.

A simple analysis of the algorithm outlines an  $O(K \cdot N)$  time complexity and a  $O(M \cdot N)$  memory complexity for a document, where  $N$  is the number of lemma in the document,  $K$  is the number of windows and  $M$  is the number of terms found in the terminology.

The implementation of the algorithm was done in Python. The script takes as input the path to a file in CONLL-U Plus format<sup>4</sup> and outputs another CONLL-U Plus file that has one extra column to the right - the ID of the terms found in the document for each token, named CURLICAT:IATE by default. Other arguments that the script can take as input are the path to the terminology (defaults to the IATE terminology), the path to the word form dictionary, the name of the added column, the maximum lemmas admitted in a term, and the terminology language. In addition, for terminology to be compatible with our script, it must be in a tabular form separated by "|" and contain the following headers:

- $E\_ID$  - the ID of the term.
- $L\_CODE$  - the language of the term.
- $T\_TERM$  - the term in words.

The implementation of the algorithm was open-sourced for public usage with an open license<sup>5</sup>.

## 5 Evaluation

We used the algorithm developed in this work to identify the IATE terms found in the CURLICAT corpus. We compare the terms identified with our string matching algorithm with those identified by the algorithm developed by Coman et al. (2019) on 100 randomly sampled documents. The documents were processed using the UDPipe processing pipeline (Straka and Straková, 2017), resulting in a dataset that contained 5.8M tokens in total, or 57.9k tokens for each document.

The results of the evaluation are depicted in Table 2. (Coman, Mitrofan, and Tufiş, 2019) algorithm managed to identify 632.4k terms, out of which 92.5k were unique. The algorithm developed in this work identified 722.1k terms, out of which 100.8k terms were unique. The total common terms found by the two algorithms amounted to 384.7k, while the unique terms amounted to 57.2k. However, although this difference between the two algorithms is high, it must be considered that the versions of IATE employed by the two are different, and that out of the 35.3k unique terms that were not found in the intersection by Coman et al. (2019), 18.8k was not found in the IATE terminology used in this work<sup>6</sup>.

Table 2. Evaluation results on the Romanian CURLICAT corpus

| Algorithm                         | # Total | # Unique |
|-----------------------------------|---------|----------|
| (Coman, Mitrofan, and Tufiş 2019) | 632.4k  | 92.5k    |
| String Matching                   | 722.1k  | 100.8k   |
| Intersection                      | 384.7k  | 57.2k    |

<sup>4</sup> <https://universaldependencies.org/format.html>

<sup>5</sup> <https://github.com/racai-ai/TerminologyAnnotator>

<sup>6</sup> We could not make a comparison with the IATE version used in that work because it is no longer available.

## 6 RELATE Integration

The RELATE platform is a modern, state-of-the-art platform, allowing the automatic annotation of large corpora, using multiple Romanian language processing pipelines (Păiș, 2020). It provides parallelization capabilities, aiming to reduce the overall time required to handle large text collections. Previous terminology detection processes (Păiș et al., 2019), available within the platform, were focused on hard-coded terminologies with the output restricted to programmer-defined columns, inside a CoNLL-U Plus file. For example, for annotating the MARCELL legal corpus (Tufiş et al., 2020; Váradi et al., 2020), a previous version of the IATE terminology was embedded into the annotation tool, with the resulting data stored in the MARCELL:IATE column. This meant that new terminologies could not be used to annotate new documents. Furthermore, using the existing tool implies post-processing to adapt the output format.

For the purposes of this work, the platform was extended to handle the uploading of arbitrary terminology specification files. Furthermore, a new task was created within the platform to handle the selection of uploaded terminology, define the output column, and schedule the task. The platform’s scheduler then distributes the task to the available processing elements (multiple processes on the same server or different servers, if available) and executes the algorithm described in this paper. Finally, it gathers the output and presents the annotated files to the user. General integration options, for different tools, were described in (Păiș et al., 2019). For the purposes of this work, we decided to employ the integration via a web service. The tool was made available as a dedicated web service. The web service can be deployed any number of times, aiming for one process for each available CPU node. This facilitates parallelization within the platform, as each document will be distributed to a dedicated process (either locally or in the network).

The easy-to-use web interface for scheduling a terminology annotation process is depicted in Fig. 2. The user can specify the column, select the terminology, as well as configure the algorithm’s parameters. The interface was implemented using web-specific technologies, such as HTML5, CSS, and JavaScript. Figure 3 presents an annotated file from the anonymized (Păiș et al., 2021) version of the CURLICAT corpus (Váradi et al., 2022). In this case, the IATE terminology was used and the results are placed in the 13th column (identified as "RELATE:IATE") in the generated CoNLL-U Plus file.

Corpus: CURLICAT\_Anonymized View Corpora

Files Standoff **Tasks** Annotated Statistics Archives

**Corpus tasks**

| Type                 | Status | Date |
|----------------------|--------|------|
| 1 unzip_text         | DC     |      |
| 2 statistics         | DC     |      |
| 3 curlicat           | DC     |      |
| 4 curlicat           | DC     |      |
| 5 unzip_text         | DC     |      |
| 6 curlicat           | DC     |      |
| 7 terminology        | DC     |      |
| 8 unzip_text         | DC     |      |
| 9 brat2conllup       | DC     |      |
| 10 ner_legalner      | DC     |      |
| 11 zin_bacis_tanning | DC     |      |

Page 1 of 2

**Add task Terminology**

Annotate text a custom terminology. The text must be already tokenized and lemmatized. At least one file with .terminology extension must be present in the corpus standoff folder.

Column: RELATE:TERM

Max Term Size: 10

Terminology: IATE.terminology

Description:

Runners (optional):

Overwrite: ☐

Add Cancel

Figure 2. Terminology annotation task window

```

# global.columns = ID FORM LEMMA UPOS XPOS FEATS HEAD DEPREL DEPS MISC RELATE:NP RELATE:NE RELATE:IATE
# newdoc
# newpar
# sent_id = 1
# text = Sigmund (n. 2 decembrie 1912, Slivilești, județul Bărăbaș - d. octombrie 1986, Grănicești a fost un poet și pictor român.
1 Sigmund Sigmunde PROPN Np - 21 nsubj - Np#1 0 -
2 ( ( PUNCT LPAR - 3 punct - SpaceAfter=No 0 -
3 n. n. NOUN Yn Abbr=Yes 1 appos - Np#2 0 -
4 2 2 NUM Mc-p-d Number=Plur|NumForm=Digit|NumType=Card 5 nummod - Np#2 B-TIME -
5 decembrie decembrie NOUN Ncms-n Definite=Ind|Gender=Masc|Number=Sing 3 nmod - Np#2 I-TIME -
6 1912 1912 NUM Mc-p-d Number=Plur|NumForm=Digit|NumType=Card 5 nummod - SpaceAfter=No Np#2 I-TIME -
7 , , PUNCT COMMA - 8 punct - 0 -
8 Slivilești Slivilești PROPN Np - 3 conj - SpaceAfter=No Np#3 0 -
9 , , PUNCT COMMA - 10 punct - 0 -
10 județul județ NOUN Ncms-sy Case=Acc,Nom|Definite=Def|Gender=Masc|Number=Sing 3 conj - Np#4 0 1:3532165
11 Bărăbaș Bărăbaș PROPN Np - 10 nmod - Np#4 B-LOC -

```

Figure 3. Example annotated file from the pseudoanonymized version of the CURLICAT corpus

## 7 Conclusion

Deterministic approximate term matching remains an important area of research due to its ability to identify terms in a corpus without using any annotated data. This paper presented a rule-based term matching algorithm that verifies whether each window of lemmas up to size  $K$  is found in the employed terminology dictionary. This algorithm was used to annotate the Romanian CURLICAT dataset with the terms found in the most recent version of the IATE terminology. Our evaluation of 100 randomly selected documents found that it manages to identify approximately 60% of the unique terms detected by a similar annotation tool, while, out of the remaining 40% distinct terms, more than 50% are not found in the newest version of IATE. The algorithm was packed into a web service and was made available for public usage on the RELATE platform. This integration allows using the algorithm in complex processing pipelines, adding a new layer of terminology annotation to any Romanian corpus. Furthermore, it allows easy access to the processing options (including terminology selection), by means of a simple web interface, as depicted in Section 6.

## Acknowledgements

This research was supported in the context of the "Curated Multilingual Language Resources for CEF.AT" (CURLICAT) project, CEF-TC-2019-1 – Automated Translation grant agreement number INEA/CEF/ICT/A2019/1926831.

## References

- Aho, Alfred V, and Margaret J Corasick. (1975). Efficient String Matching: An Aid to Bibliographic Search. *Communications of the ACM*, 18 (6): 333–40.
- Bao, Zhifeng, Tok Wang Ling, Bo Chen, and Jiaheng Lu. (2009). Effective Xml Keyword Search with Relevance Oriented Ranking. In *2009 IEEE 25th International Conference on Data Engineering*, 517–28. IEEE.
- Charalampopoulos, Panagiotis, Solon Pissis, Jakub Radoszewski, Tomasz Waleń, and Wiktor Zuba. (2020). Unary Words Have the Smallest Levenshtein  $k$ -Neighbourhoods. In *CPM 2020-31st Annual Symposium on Combinatorial Pattern Matching*, 1–12.
- Coman, A, M Mitrofan, and D Tufiş. (2019). Automatic Identification and Classification of Legal Terms in Romanian Law Texts. In *Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR 2019)*, 39–49.
- Goyal, Archana, Vishal Gupta, and Manish Kumar. (2018). Recent Named Entity Recognition and Classification Techniques: A Systematic Review. *Computer Science Review*, 29: 21–43.
- Johnson, Ian, and Alastair Macphail. (2000). IATE-Inter-Agency Terminology Exchange: Development of a Single Central Terminology Database for the Institutions and Agencies of the European Union. In *Workshop on Terminology Resources and Computation*.
- Kumar, Tapesh, Mehul Mahrishi, and Gaurav Meena. (2022). A Comprehensive Review of Recent Automatic Speech Summarization and Keyword Identification Techniques. *Artificial Intelligence in Industrial Applications*, 111–26.
- Li, Jing, Aixin Sun, Jianglei Han, and Chenliang Li. (2020). A Survey on Deep Learning for Named Entity Recognition. *IEEE Transactions on Knowledge and Data Engineering*, 34 (1): 50–70.

- Litvak, Marina, and Mark Last. (2008). Graph-Based Keyword Extraction for Single-Document Summarization. In *Coling 2008: Proceedings of the Workshop Multi-Source Multilingual Information Extraction and Summarization*, 17–24.
- Mititelu, Verginica Barbu, Dan Tufiş, and Elena Irimia. (2018). The Reference Corpus of the Contemporary Romanian Language (CoRoLa). In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*.
- Păiş, Vasile. (2020). Multiple Annotation Pipelines Inside the RELATE Platform. In *The 15th International Conference on Linguistic Resources and Tools for Natural Language Processing*, 65–75. <https://profs.info.uaic.ro/~consilr/wp-content/uploads/2021/03/volum-ConsILR-v-4-final-revizuit.pdf#page=73>.
- Păiş, Vasile, Radu Ion, and Dan Tufiş. (2020). A Processing Platform Relating Data and Tools for Romanian Language. In *Proceedings of the 1st International Workshop on Language Technology Platforms*, 81–88. Marseille, France: European Language Resources Association. <https://www.aclweb.org/anthology/2020.iwltpl-1.13>.
- Păiş, Vasile, Elena Irimia, Radu Ion, Dan Tufiş, Maria Mitrofan, Verginica Barbu Mititelu, Andrei-Marius Avram, and Eric CUREA. (2021). Romanian Text Anonymization Experiments from the CURLICAT Project. In *The 17th Edition of the International Conference on Linguistic Resources and Tools for Natural Language Processing - CONSILR*.
- Păiş, Vasile, Dan Tufiş, and Radu Ion. (2019). Integration of Romanian NLP Tools into the RELATE Platform. In *International Conference on Linguistic Resources and Tools for Natural Language Processing*, 181–92. [https://profs.info.uaic.ro/~consilr/2019/wp-content/uploads/2020/01/ConsILR2019\\_final\\_BTT-60-ex-B5.pdf#page=189](https://profs.info.uaic.ro/~consilr/2019/wp-content/uploads/2020/01/ConsILR2019_final_BTT-60-ex-B5.pdf#page=189).
- Spiekermann, Sarah. (2012). The Challenges of Privacy by Design. *Communications of the ACM*, 55 (7): 38–40.
- Straka, Milan, and Jana Straková. (2017). Tokenizing, Pos Tagging, Lemmatizing and Parsing Ud 2.0 with Udpipes. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 88–99.
- Tufiş, Dan. (2009). Paradigmatic Morphology and Subjectivity Mark-up in the RoWordNet Lexical Ontology. In *Intelligent Systems and Technologies*, 161–79. Springer.
- Tufiş, Dan, Verginica Barbu Mititelu, Elena Irimia, Vasile Păiş, Radu Ion, Nils Diewald, Maria Mitrofan, and Onofrei Mihaela. (2019). Little Strokes Fell Great Oaks. Creating CoRoLa, the Reference Corpus of Contemporary Romanian. *Revue Roumaine de Linguistique*, 64 (3): 227–40.
- Tufiş, Dan, Maria Mitrofan, Vasile Păiş, Radu Ion, and Andrei Coman. (2020). Collection and Annotation of the Romanian Legal Corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 2773–77. Marseille, France: European Language Resources Association. <https://www.aclweb.org/anthology/2020.lrec-1.337>.
- Váradi, Tamás, Svetla Koeva, Martin Yamalov, Marko Tadić, Bálint Sass, Bartłomiej Nitoń, Maciej Ogrodniczuk, et al. (2020). The MARCELL Legislative Corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 3761–68. Marseille, France: European Language Resources Association. <https://www.aclweb.org/anthology/2020.lrec-1.464>.
- Váradi, Tamás, Bence Nyéki, Svetla Koeva, Marko Tadić, Vanja Štefanec, Maciej Ogrodniczuk, Bartłomiej Nitoń, et al. (2022). Introducing the CURLICAT Corpora: Seven-Language Domain Specific Annotated Corpora from Curated Sources. In *Proceedings of the Language Resources and Evaluation Conference*, 100–108. Marseille, France: European Language Resources Association. <https://aclanthology.org/2022.lrec-1.11>.
- Váradi, Tamás, Marko Tadić, Svetla Koeva, Maciej Ogrodniczuk, Dan Tufiş, Radovan Garabík, Simon Krek, and Andraž Repar. (2022). Curated Multilingual Language Resources for CEF AT (CURLICAT): Overall View. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, 339–40.



# FINE-GRAINED LEGAL REFERENCES IN THE LEGALNERO CORPUS

DARIA COSTEA, ANDREI MOLDOVAN, AND VASILE PĂIȘ

*Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, Bucharest, Romania*

## Abstract

The LegalNERo corpus contains gold annotations for named entity recognition in the Romanian legal domain. These include references to legal documents. For the purposes of this work, we manually annotated the legal references into fine-grained entity classes. As a result, we released a new version of the corpus and we trained named entity recognition models using the new fine-grained classes. In addition, we release an algorithm for automatically classifying expressions denoting legal references into one of the fine-grained classes used in LegalNERo.

## 1 Introduction

The legal industry faces special difficulties when it comes to natural language processing. The structure of legal documents and the usage of domain-specific language are responsible for this. In recent years, technology relating to legal papers has received more attention. This is reflected by the volume of recent scientific publications, the Natural Legal Language Processing (NLLP) workshop, and other international projects addressing natural language processing for the legal sector (Aletras et al., 2019; Aletras et al., 2020).

The process of locating text spans that make reference to people, places, or objects is known as named entity recognition, or NER. The CoNLL2003 shared task on language-independent NER (Tjong et al., 2003) introduced one of the annotation techniques that is used in a large number of publications and relates to names of people, organizations, and locations. This annotation scheme can be used to annotate legal documents as well, allowing existing systems to attempt to do so (with or without being adapted to legal text). To improve the information extraction capabilities of text processing algorithms specifically created for the legal domain, domain-specific entities are typically incorporated. Dozier et al. (2010) suggest including entities for lawyers, judges, courts, and jurisdictions for dealing with depositions, pleadings, and trial-specific materials. For the purpose of evaluating legal contracts, Glaser et al. (2018) suggested including the entity's date, monetary value, legal reference, and "other". Leitner et al. (2017) further proposed seven coarse-grained entity classes that can be expanded into 19 fine-grained legal named entity classes.

Considering the Romanian language, the "Multilingual Resources for CEF.AT in the legal domain" (MARCELL) initiative involved the creation of a large, compact, and validated domain-specific corpus. It provides monolingual corpora in the languages of the seven participants, including the Romanian language. The corpora are extracted from national legislation (laws, decrees, rules, etc) (Váradi et al., 2020). The Romanian corpus (Tufiş et al., 2020) was preprocessed, separated at the sentence level, tokenized, lemmatized, and annotated with POS tags. Using a general-purpose tool, named entities were recognized (Păiș, 2019). The only entity types supported by this tool, which was created for use with Romanian as a whole, were organizations, locations, people, and time expressions. No legal texts were used in the training of the tool.

The lack of a Romanian language NER tool in the legal domain justified the creation of the LegalNERo corpus (Păiș et al., 2021). It contains legal documents manually annotated with named entities (NE) and expressions, including persons, locations, organizations, time expres-

sions, and legal references. The legal references class is a coarse-grained class, including references to different types of legal documents. For the purposes of this work, we transformed this annotation into 15 fine-grained classes: Law, Ordinance, Publication, Decree, Decision, Treaty, Report, Order, Regulation, Directive, EmergencyOrdinance, Norm, Convention, Code, and Other. As a result, we released a new version of the corpus, freely available for research purposes on the Zenodo platform<sup>1</sup>. In addition, we constructed an algorithm allowing the classifying of a legal reference expression into one of the fine-grained classes. This is useful when using a NER model trained with the coarse-grained legal reference class, such as (Păiș et al., 2021). Finally, we train new NER models on the newly released version of the corpus, allowing for direct prediction of the fine-grained legal reference classes.

This paper is organized as follows: in Section 2, we present related work; in Section 3, the annotations are described; in Section 4, a rule-based fine-grained classification algorithm is introduced; and in Section 5, we introduce a name entity recognition model trained on the fine-grained annotations. Finally, we conclude in Section 6. We make the algorithm public<sup>2</sup> with all the associated resources.

## 2 Related Work

Legal NER is a critical task in extracting key information from legal documents, such as dates, references to various types of legal documents, locations, organizations, and individuals. According to Zhong and Tang (2020), once the NEs have been identified and classified, they can be used in workflows to perform various functions such as document anonymization or case summarization.

Dozier et al. (2010) conducted one of the pioneering studies in this field. The authors used statistical models, context rules, and a lookup list of NEs to examine legal NEs in US depositions, pleadings, case law, and other legal documents. They also created different taggers for NEs such as document type of jurisdiction, achieving an F1 score of 0.92 for NEs in the jurisdiction class. This work served as the foundation for Cardellino et al. (2017) to create a tool for identifying, categorizing, and linking legal NEs. They trained and evaluated various systems (Stanford NER, a Support Vector Machine, and a neural network (NN)) on Wikipedia and European Court of Human Rights decisions, obtaining an F1 score of 0.56 using Stanford NER.

Glaser et al. (2018) investigated the applicability of named entity recognition systems to legal contracts. Person, organization, location, date, money value, reference, and other are the proposed entity classes. The "reference" entity is based on the work of Landthaler et al. (2016), which takes into account references to legal norms.

Leitner et al. (2017) presented a German legal named entity corpus consisting of seven coarse-grained classes that can be further subdivided into 19 fine-grained classes. A "person" entity in this case can be a regular person, a judge, or a lawyer. A legal reference entity, similarly, can be expanded into law, ordinance, or European legal norm.

Acosta et al. (2019) describe an approach to Named Entity Recognition (NER) in German language documents from the legal domain. For this purpose, a dataset consisting of German court decisions was developed. The source texts were manually annotated with 19 semantic classes: person, judge, lawyer, country, city, street, landscape, organization, company, institution, court, brand, law, ordinance, European legal norm, regulation, contract, court decision, and legal literature. The dataset consists of approx. 67,000 sentences and contains 54,000 annotated entities. The 19 fine-grained classes were automatically generalized to seven more coarse-grained classes (person, location, organization, legal norm, case-by-case regulation, court decision, and legal literature). Thus, the dataset includes two annotation variants, i.e., coarse- and fine-grained. For the

<sup>1</sup> <https://zenodo.org/record/7025333>

<sup>2</sup> <https://github.com/racai-ai/LegalDomainNER>

task of NER, Conditional Random Fields (CRFs) and bidirectional Long-Short Term Memory Networks (BiLSTMs) were applied to the dataset as state-of-the-art models. Three different models were developed for each of these two model families and tested with the coarse- and fine-grained annotations. The BiLSTM models achieve the best performance with a 95.46% F1 score for the fine-grained classes and 95.95% for the coarse-grained ones. The CRF models reach a maximum of 93.23% for the fine-grained classes and 93.22% for the coarse-grained ones. Their work was carried out under the umbrella of the European project LYNX, which develops a semantic platform that enables the development of various document processing and analysis applications for the legal domain.

### 3 Annotations

LegalNERo is a manually annotated corpus for named entity recognition in the Romanian legal domain. The initial annotation scheme provided gold annotations for organizations, locations, persons, time, and legal resources mentioned in legal documents (legal references). The LegalNERo corpus is available in different formats: span-based, token-based, and RDF (combining span and token annotations, useful for Linguistic Linked Open Data (LLOD) applications).

The transformation of the legal reference class into fine-grained NE classes was performed by two annotators, under the supervision of a senior researcher at the Research Institute for Artificial Intelligence "Mihai Drăgănescu" of the Romanian Academy. For this purpose, all the legal reference annotations were extracted into a separate file and each of the annotators received half of them. After completing their own annotation, the file was checked by the other annotator. Finally, the resulting classification was checked again and any expressions that were unclear were discussed and annotated during annotation meetings.

The newly introduced fine-grained entity classes are:

- *Law* consists of 3 types of law: constitutional law, common law, and organic law. Law means any provision of law with binding force by coercion by public authorities. Example: "Legea nr. 104 / 2015" ("Law no. 104/2015").
- *Ordinance* are normative acts that supplement the legislative function of the Parliament. Example: "Ordonanța Guvernului nr. 22 / 2007" ("Government ordinance no. 22 / 2007").
- *Publication* includes catalogs, the official monitor, and registers. Examples: "MONITORUL OFICIAL"("Official Monitor"), "Registrul național al organizațiilor de îmbunătățiri funciare cu nr. 644 / 11.06.2019" ("National register of land improvement organizations with no. 644 / 11.06.2019").
- *Decree* is usually issued by a head of state in accordance with certain procedures (usually laid down in a constitution). The decree is also defined as the act by which general mandatory provisions are established or by which certain legal situations are regulated. We include here "decree law" ("decret lege"). Example: "DECRET nr. 858 din 25 noiembrie 2019" ("DECREE no. 858 of November 25, 2019").
- *Decision* are of two types. One is government decisions, which establish how laws will be implemented or other organizational aspects related to their implementation. The second one is a binding legal document valid for one or more countries/companies/individuals. The affected party shall be notified and the decision shall take effect upon receipt of the notice. Decisions do not need to be translated into national law. Examples: "HOTĂRÂRE nr. 13 din 17 ianuarie 2019"("DECISION no. 13 of January 17, 2019"), "Deciziei nr. 541 / 2014 / UE" ("Decision no. 541 / 2014 / EU").
- *Treaty* sets out the objectives of the European Union, the rules governing the EU institutions, the decision-making process, and the relationship between the EU and its Member

States. They are negotiated and agreed upon by all member states and then ratified by the national parliament, sometimes after a referendum. Also, there can be treaties between two or more states that are not in European Union. Example: "Tratatului de instituire a Comunității Europene" ("The Treaty establishing the European Community" Decision no. 541 / 2014 / EU").

- *Reports* are written documents describing the findings of an organization or some individual. Example: "Referatul Direcției management și structuri sanitare nr. SP 4.603 din 5.04.2019" ("Management and sanitary structures Directorate report no. SP 4.603 of 04/05/2019").
- *Order* is a prescriptive act issued by a central administration on the basis of and only in the execution of laws, decisions, and governmental regulations. Example: "Ordinului ministrului sănătății nr. 368 / 2017" ("Order of the Minister of Health No. 368 / 2017").
- *Regulations* are legal acts that are automatically and uniformly applied in all Member States as soon as they come into force without needing to translate them into national law. They are fully binding on all EU countries. Example: "Regulamentului de punere în aplicare (UE) nr. 2017 / 373" ("Implementing Regulation (EU) No. 2017 / 373").
- *Directive* requires Member States to achieve a certain outcome but does not specify the means by which they may do so. Member States must take measures to translate directives into their national legislation in order to achieve the goals they have set for themselves. The conversion into national law must take place before the deadline specified for the adoption of the directive. Example: "Directivei 2005 / 36 / CE" ("Directive 2005 / 36 / EC").
- *Emergency Ordinance* is a normative law enacted by the government in extraordinary situations, the regulation cannot be delayed. This type of normative act constitutes a form in which the executive power in Romania (the government) is vested with legislative power, in this case, legislative power, which normally only parliament has the power to pass. Example: "Ordonanța de urgență a Guvernului nr. 23 / 2008" ("Government Emergency Ordinance No. 23 / 2008").
- *Norm* is established or sanctioned by public authorities, established by the state in the form of specific laws. Example: "Normei Autorității de Supraveghere Financiară nr. 4 / 2018" ("Regulations of the Financial Supervision Authority no. 4 / 2018").
- *Convention* is an accord between states or nations, which resembles a treaty: ordinarily applied to agreements prior to an execution of an official treaty or which serve as its foundation; or to international agreements for the regulation of international affairs of common interest not within the ambit of commercial transactions or politics, such as international postage. Example: "Convenția internațională din 1974 pentru ocrotirea vieții omenești pe mare" ("1974 International Convention for the Safety of Life at Sea").
- *Code* is a collection of written laws gathered together, usually covering specific subject matter. Example: "Codului deontologic al farmacistului" ("Pharmacist's ethical code").
- *Other* – this class contains any legal references that do not fit into one of the above classes. It includes legal references such as opinions, licenses, letters, processes, protocols, procedures, authorizations, and archives. Examples: "Avizul conform al Ministerului Finanțelor Publice" ("Opinions of the Ministry of Public Finance"), "Contractului-cadru de vânzare-cumpărare a gazelor naturale" ("The framework contract for the sale and purchase of natural gas").

Statistics computed using the fine-grained annotations are given in Table 1. Furthermore, Figure 1 presents the distribution of the classes, considering the total number of instances associated with each class, while Figure 2 presents the distribution considering only the unique instances (certain legal references appear multiple times in the LegalNERo corpus).

*Table 1. Statistics of the fine-grained legal reference classes within the LegalNERo corpus*

| <b>Class</b>       | <b># Total</b> | <b># Unique</b> |
|--------------------|----------------|-----------------|
| Regulation         | 283            | 180             |
| Decree             | 10             | 5               |
| Ordinance          | 111            | 84              |
| Publication        | 57             | 23              |
| Code               | 80             | 16              |
| Law                | 890            | 229             |
| Convention         | 44             | 9               |
| Decision           | 682            | 308             |
| Directive          | 19             | 14              |
| Norm               | 40             | 29              |
| Order              | 654            | 440             |
| EmergencyOrdinance | 240            | 82              |
| Report             | 102            | 101             |
| Treaty             | 16             | 4               |
| Other              | 157            | 98              |
| <b>TOTAL</b>       | <b>3,385</b>   | <b>1,622</b>    |

Following the annotation process, the fine-grained classes were incorporated into the corpus. This process used the same approach followed when releasing the original corpus and resulted in multiple annotation formats. A span-based annotation, using the BRAT file format, contains the text position of the expression (start position and end position), the fine-grained class, and the expression itself. Furthermore, a token-based annotation, using the CoNLL-U Plus file format, contains IOB token annotations.

#### 4 Fine-grained Classification Algorithm

We developed an algorithm that classifies expressions denoting legal references into one of the fine-grained classes. The algorithm employs a set of keywords and handcrafted rules between them. The keywords are represented by words that were found often in the expressions. Then a function transforms the input expression (given in CoNLL-U format) and gives the answer according to one of the fine-grained legal reference types. Some of the NE classes have a significant keyword, such as "Decizia" ("Decision") which makes it easy for the model to detect, while others do not. For this reason, we employed a topological sort that orders the keywords on the set of rules. Each rule is expressed by two keywords, the first of which is more important than the second. A keyword is more important than another if both appear in the same expression and the answer is the first one. This enables us to develop an order of relevancy. The algorithm is described in Listing 1.

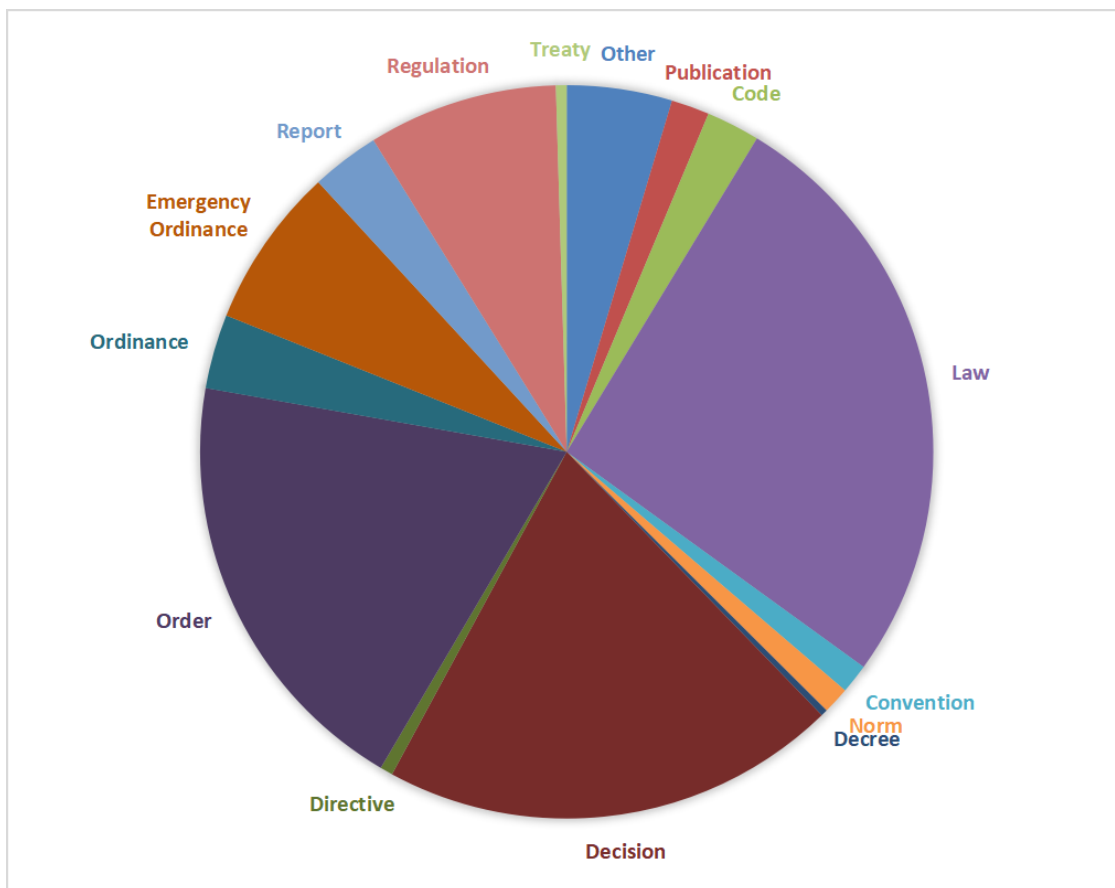


Figure 1. Distribution of the fine-grained entity classes, considering their total number

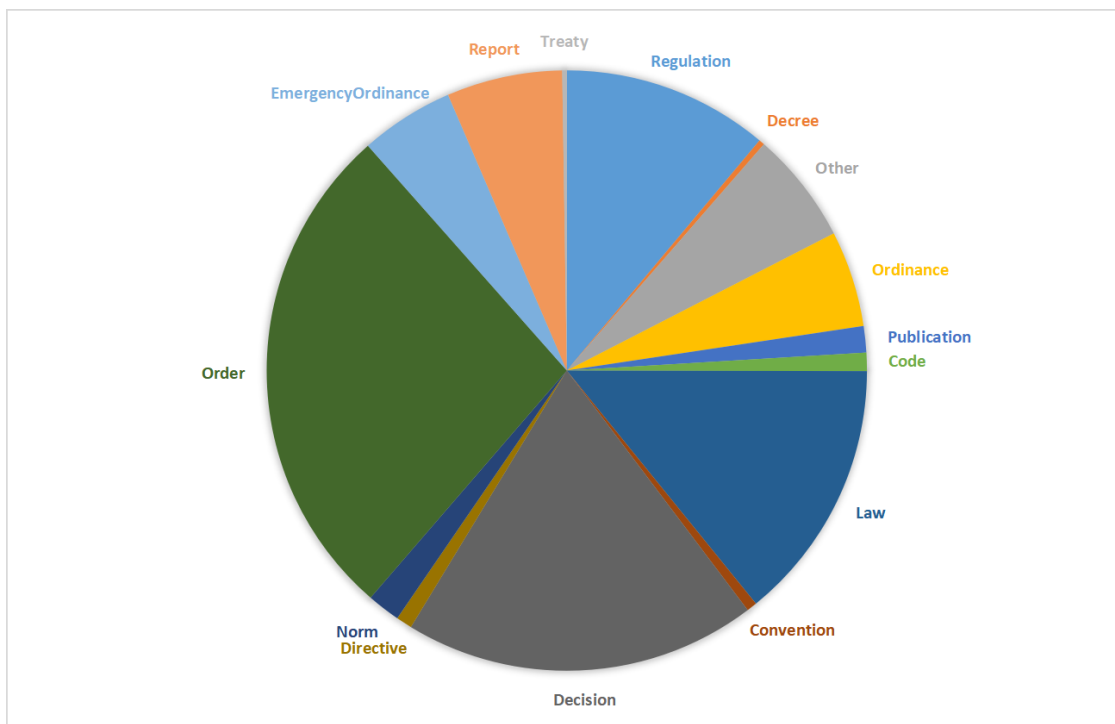


Figure 2. Distribution of the fine-grained entity classes, considering only unique instances

Input: CONLL-U file with the expression to be annotated, containing word form and lemma  
Output: Type=fine-grained legal reference type  
Resources: a txt file with set of rules and a txt file with keywords  
1.  $K = \text{read}(\text{keywords})$   
2.  $R = \text{read}(\text{set of rules})$   
3.  $C = \text{read}(\text{CONLL-U})$   
4.  $\text{topological sort}()$   
return the topological sort of the expressions using the set of rules from the predefined input. Every rule is considered an edge in the directed graph.  
5. Keep only the keywords from the expression ( $K \cap C$ ).  
Intersection takes into account the Levenshtein based similarity between word form/lemma and a keyword  

$$\frac{2 * \text{len}(\text{word}) - \text{Levenshtein}(\text{word}, \text{keyword})}{\text{len}(\text{word}) + \text{len}(\text{keyword})} \geq 0.82$$
  
6. Return the associated type from the lowest rank in the topological sort from the  $K \cap C$

*Listing 1. Fine-grained legal reference classification algorithm*

The algorithm was implemented in Python and the rules and keywords were stored in txt files. The first step is using the topological sort of the keywords. The second step is returning a list with the keywords that are in the expression read from the CONLL-U file. After that, the algorithm returns the word with the lowest rank in the topological sort. The implemented algorithm was tested on the manually annotated data set and achieved an accuracy (computed using Equation 1) of 98.46%. From the total of 1,622 expressions, there were 1,597 correctly identified.

$$\text{Accuracy} = \frac{\text{CorrectlyPredictedTypes}}{\text{TotalExpressions}} \quad (1)$$

## 5 Named Entity Recognition Model

We employed the newly created version of the LegalNERo corpus for training a NER model, considering all the available NE classes, including the fine-grained legal references. We employed a contextual embeddings layer, provided by the multilingual XLM-RoBERTa model (Conneau et al., 2020). This model includes support for the Romanian language. The contextual representation was then passed through a layer inspired by the biological process of lateral inhibition, referring to the capacity of excited neurons to reduce the activity of their neighbors (Cohen, 2011). This approach was introduced by (Păiș, 2022) and allows the network to obtain better representations in difficult cases. It has been used also for improving the results of a Romanian biomedical NER system (Mitrofan and Păiș, 2022). The lateral inhibition layer employs the Heaviside function (Berg, 1937) in the forward pass. In the backward pass, the non-differentiable function is approximated with a sigmoid function with a scaling parameter. This approximation technique is also known as surrogate gradient learning (Neftci et al., 2019). The system employs an additional linear layer and a classification head for actually predicting the NE classes. We trained the system for 80 epochs, considering a learning rate of  $2e-5$ , a batch size of 128, a sequence length of 128 tokens, and applying a dropout of 0.1 to the lateral inhibition layer and the linear layer. We finally saved the best-performing model, which was obtained in epoch 30.

For training the model, the corpus was split into 70% training, 15% validation, and 15% testing data. The split was randomly created at the file level while trying to ensure the corresponding 70-15-15% of each entity class is available in the splits. The resulting split was also included in the LegalNERo corpus to allow for future comparison with improved NER models.

The results of the test split are given in Table 2. As expected, classes that are less represented in the corpus ("Directive", "Publication", "Decree", "Convention") are very difficult to predict, thus lowering the overall score. Thus, given the current number of NE classes and the NER results, we consider that a NER system making use only of the generic legal reference class, combined with the algorithm described in Section 4, performs better. The system described in (Păiş et al., 2021) provides an overall macro F1 score of 0.8390 (which is a 5% increase compared to the current results) with a corresponding F1 of 0.9050 for the legal reference class<sup>3</sup>.

Table 2. Evaluation results, considering all fine-grained classes, on the test split

| Class              | Precision | Recall | F1     |
|--------------------|-----------|--------|--------|
| TIME               | 0.8255    | 0.9008 | 0.8615 |
| ORG                | 0.7689    | 0.7943 | 0.7814 |
| Norm               | 0.5000    | 0.2308 | 0.3158 |
| Decree             | 0.3333    | 0.2000 | 0.2500 |
| Order              | 0.9423    | 0.9703 | 0.9561 |
| Law                | 0.9237    | 0.9478 | 0.9356 |
| PER                | 0.8278    | 0.8418 | 0.8347 |
| LOC                | 0.5985    | 0.7022 | 0.6462 |
| Ordinance          | 0.7500    | 0.8400 | 0.7925 |
| EmergencyOrdinance | 1.0000    | 1.0000 | 1.0000 |
| Regulation         | 0.7037    | 0.5938 | 0.6441 |
| Other              | 0.0606    | 0.0909 | 0.0727 |
| Decision           | 0.9101    | 0.9101 | 0.9101 |
| Code               | 1.0000    | 0.6000 | 0.7500 |
| Treaty             | 1.0000    | 0.4286 | 0.6000 |
| Report             | 0.5172    | 0.7895 | 0.6250 |
| Directive          | 0.0000    | 0.0000 | 0.0000 |
| Publication        | 1.0000    | 0.1429 | 0.2500 |
| Convention         | 0.5000    | 0.4545 | 0.4762 |
| micro avg          | 0.7673    | 0.8031 | 0.7848 |
| macro avg          | 0.7750    | 0.8031 | 0.7852 |

## 6 Conclusion

Domain-specific NER is still a challenging task within the realm of natural language processing, due to the reduced number of available gold corpora. Our work produced a new level of annotations within the LegalNERo legal-domain corpus, consisting of fine-grained legal references. We constructed a NER system making use of state-of-the-art technology, including contextualized word embedding representations. However, the model performs poorly on some of the fine-grained entity classes, due to the reduced number of examples associated with them. For this reason, we also constructed an algorithm allowing automatic mapping from a legal reference expres-

<sup>3</sup> From the experiments described in the referenced paper we noted only the one with the best F1 score for the legal reference class on the test set.

sion to one of the fine-grained classes. This rule-based approach can be used with a NER model trained without knowledge of the fine-grained classes, such as the one released by (Păiș et al., 2021). Improving the NER performance requires more data, especially for the less-represented NE classes. Thus, future work will include annotating new documents containing the under-represented classes.

## References

- Acosta, Maribel, Philippe Cudré-Mauroux, Maria Maleshova, Tassilo Pellegrini, Harald Sack, and Yore Sure-Vetter. (2019). Semantic Systems. The Power of AI and Knowledge Graphs. In *Internationales Rechtsinformatik Symposium*.
- Aletras, Nikolaos, Androutsopoulos Ion, Leslie Barrett, Adam Meyers, and Daniel Preotiuc-Pietro. (2020). *Proceedings of the Natural Legal Language Processing*. Processing Workshop 2020.
- Aletras, Nikolaos, Elliott Ash, Leslie Barrett, Daniel Chen, Adam Meyers, Daniel Preotiuc-Pietro, David Rosenberg, and Amanda Stent. (2019). *Proceedings of the Natural Legal Language Processing*. Association for Computational Linguistics, Minneapolis, Minnesota.
- Berg, T. A. A. (1937). Heaviside’s Operational Calculus as Applied to Engineering and Physics. By e. J. Berg. 2nd Edition. Pp. Xv, 258. 18s. Electrical Engineering Texts. (McGraw-Hill). *The Mathematical Gazette*, 21 (245): 309–10. <https://doi.org/10.1017/S0025557200058289>.
- Cardellino, Cristian, Milagro Teruel, Alonso Laura Alemany, and Serena Villata. (2017). A Low-Cost, High-Coverage Legal Named Entity Recognizer, Classifier and Linker. In *Proceedings of the 16th Edition of the International Conference on Artificial Intelligence and Law*, 9–18.
- Cohen, Ronald A. (2011). Lateral Inhibition. In Jeffrey S. Kreutzer, John DeLuca, and Bruce Caplan (Eds.), *Encyclopedia of Clinical Neuropsychology* (1436–37). New York, NY: Springer New York. [https://doi.org/10.1007/978-0-387-79948-3\\_1379](https://doi.org/10.1007/978-0-387-79948-3_1379).
- Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. (2020). Unsupervised Cross-Lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8440–51. Online: Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-main.747>.
- Dozier, Christopher, Ravikumar Kondadadi, Marc Light, Arun Vachher, Sriharsha Veeramachaneni, and Ramdew Wudali. (2010). *Named Entity Recognition and Resolution in Legal Text*. Springer Berlin Heidelberg, 27–43.
- Glaser, Ingo, Bernhard Walt, and Florian Matthes. (2018). Named Entity Recognition, Extraction, and Linking in German Legal Contracts. In *Internationales Rechtsinformatik Symposium*, 325–34.
- Landthaler, Jörg, Bernhard Walzl, and Florian Matthes. (2016). Unveiling References in Legal Texts-Implicit Versus Explicit Network Structures. In *Internationales Rechtsinformatik Symposium*, 8: 71–80.
- Leitner, Elena, Georg Rehm, and Julian Moreno-Schneider. (2017). *The Power of AI and Knowledge Graphs*. Springer International Publishing, 272–87.
- Mitrofan, Maria, and Vasile Păiș. (2022). Improving Romanian BioNER Using a Biologically Inspired System. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, 316–22. Dublin, Ireland: Association for Computational Linguistics. <https://aclanthology.org/2022.bionlp-1.30>.
- Neftci, Emre O., Hesham Mostafa, and Friedemann Zenke. (2019). Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks. *IEEE Signal Processing Magazine*, 36 (6): 51–63. <https://doi.org/10.1109/MSP.2019.2931595>.
- Păiș, Vasile, Maria Mitrofan, Carol Luca Gasan, Vlad Coneschi, and Alexandru Ianov. (2021). Named Entity Recognition in the Romanian Legal Domain. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, 9–18. Punta Cana, Dominican Republic: Association for Computational Linguistics. <https://aclanthology.org/2021.nllp-1.2>.
- Păiș, Vasile. (2022). RACAI at SemEval-2022 Task 11: Complex Named Entity Recognition Using a Lateral Inhibition Mechanism. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, 1562–69. Seattle, United States: Association for Computational Linguistics. <https://aclanthology.org/2022.semeval-1.215>.

- Păiș, Vasile, Maria Mitrofan, Carol Luca Gasan, Alexandru Ianov, Corvin Ghiță, Vlad Silviu Coneschi, and Andrei Onuț. (2021). Romanian Named Entity Recognition in the Legal domain (LegalNERo). *Zenodo*. <https://doi.org/10.5281/zenodo.4772094>.
- Tjong, Kim, Erik Sang, and Fien Meulder. (2003). Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*, 142–47.
- Tufiș, Dan, Maria Mitrofan, Vasile Păiș, Radu Ion, and Andrei Coman. (2020). Collection and Annotation of the Romanian Legal Corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 2766–70. Marseille, France: European Language Resources Association. <https://www.aclweb.org/anthology/2020.lrec-1.337/>.
- Váradi, Tamás, Svetla Koeva, Martin Yamalov, Marko Tadić, Bálint Sass, Bartłomiej Nitoń, Maciej Ogrodniczuk, et al. (2020). The MARCELL Legislative Corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 3754–61. Marseille, France: European Language Resources Association. <https://www.aclweb.org/anthology/2020.lrec-1.464/>.
- Zhong, Qing, and Yan Tang. (2020). An Attention-Based Bilstm-Crf for Chinese Named Entity Recognition. In *IEEE 5th International Conference on Cloud Computing and Big Data Analytics*, 550–55.

## **CHAPTER 2**

### **LANGUAGE PROCESSING TOOLS**



# AI-BASED SOLUTIONS FOR TEXT LANGUAGE IDENTIFICATION

RADU A. CIORA<sup>1</sup> AND DANIELA GÎFU<sup>2,3</sup>

<sup>1</sup>*Faculty of Mechanical and Electrical Engineering, University of Petroșani,  
Romania, raduciora@gmail.com*

<sup>2</sup>*Institute of Computer Science, Romanian Academy, 2 Codrescu St, Iasi, Romania*

<sup>3</sup>*Alexandru Ioan Cuza University, Faculty of Computer Science, 16 Berthelot St.,  
Iași, Romania, daniela.gifu@iit.academiaromana-is.ro*

## Abstract

The problem of determining the online content language has gained increasing interest in an era when user-generated online content, more specifically – social media posting – plays a key role in the distribution chain of various types of news stories. In fact, this kind of content suffers from a lack of professional gatekeepers to analyze and validate it. As a result, we need enhanced tools that can help us speed up the process of detection of linguistic aspects of online content. This paper presents a novel research method intended to identify the language of short texts, based on AI algorithms considering a Twitter collection, related to COVID-19 subject. The aim of this work is to optimally determine the main language in which a text is written on social networks, considering the variability of the language and its abbreviated forms that are specific to online communication. Therefore, to find the best solution for language detection, a comparative analysis between state-of-the-art language detection algorithms for short texts was undertaken. Moreover, the results of this analysis were used for the implementation of a new language detection model, focused, primarily, on the detection of the Romanian language of the gathered tweets.

**Keywords:** language detection, Twitter collection, lexical units, artificial intelligence, binary classification.

## 1. Introduction

The task of language identification (LI), also called language detection, consists of determining the natural language that a text or part of it is written in (Jauhiainen *et al.*, 2018; Lui & Baldwin, 2012). The problem of language identification has gained increasing importance nowadays when user-generated content (UGC) on social media is a driving force in the dissemination of news stories (Boididou *et al.*, 2018). Moreover, due to urbanization, globalization, and human mobility, this kind of content suffers from a lack of professional gatekeepers able to analyze it (Afify *et al.*, 2019). As a consequence, this superdiversity extends beyond traits such as ethnicity and country of origin (Vertovec, 2007). Therefore, the need for new tools that can speed up the process of linguistics content identification is imperative.

The aim of this work is to analyze existing algorithms that determine the main language in which a text is written, considering the social networks' particularities like the variability of the language and the abbreviated forms specific to online communication behavior. Therefore, in order to develop a language identification algorithm, a comparative analysis between state-of-the-art language identification algorithms for short texts was undertaken.

The legitimate question of this survey is: *How to optimally determine the language of a piece of text on social media?*

In this article, we pursued this idea further by examining social media, generally based on the urban superdiversity (Väisänen *et al.*, 2022), through language use. Our approach aims to test and assess the results of various language detection algorithms available on the shelf and come up

with solutions for improvement in the language detection domain, with specific applicability to the detection of the Romanian language.

The research results were used in the implementation of a new language detection model, focused primarily on the detection of the Romanian language from tweets. We focused on the Romanian language having in view the limited openness and low penetration barrier of the content generated through social media platforms that is written in Romanian.

The paper is structured as follows: Section 2 presents a short background of several approaches to identify the language of a written text in order to have a realistic starting point about the topic in question. Section 3 describes a dataset focused on the Romanian language and the phases being performed in this method, including seven algorithms we encountered in the literature in order to propose a new language detection model. Section 4 presents the results of this survey, before drawing some conclusions in the last section.

## 2. Related Research

The size of the public information available online, on the internet has grown exponentially because of the pandemic (Wong *et al.*, 2021). COVID-19 (SARS-CoV-2) pandemic brought along with the virus, another so-called infection - the Internet information-sharing virus (Mhlanga, 2022). Moreover, the focus on information retrieval and processing has shifted towards online content like microblogging, microservices, RSS feeds, etc. (Esteva *et al.*, 2021). As opposed to traditional press and other communication means that used to be regional or country-wide limited, the online medium created the premises for an interconnected society, which is not geolocation tight, but its borderless and boundless properties make it almost impossible to manage and predict. Therefore, data filtering (Herold *et al.*, 2021) is a primordial task for any data processing in such a complex environment.

Selecting proper features for language identification (Zampieri *et al.*, 2017) is probably the most important step for bootstrapping the identification process. Depending on the algorithm utilized, the feature set comprises the following parts.

- *Text*, or the document under investigation, is the original piece of written information, in plain language.
- *Short texts* – according to literature, a short text length is under 100 characters.
- *Labels* – are tagging elements attached to a piece of text – like positive/negative tags – used in sentiment analysis methods.
- *Dataset* – is the corpus of texts that is labeled correctly, representing the ground truth from which machine learning algorithms learn to tag information.
- *Text classification* – is a machine learning technique that can be used to organize, structure, and categorize any open-ended texts by assigning them a set of predefined labels and/or categories.
- *Language Identification* (LangID) – is a special kind of text classification where the labels are the language in which the text is written.
- *Script* – a set of alphabets used in the writing of a text.

*How to determine the language of a piece of text?* In order to respond to the question, below, we present an analysis of existing state-of-the-art approaches toward language detection from texts.

### 2.1. Polyglot

In the online multilingual context, language detection systems often have to deal with texts containing inclusions of multiple other languages in form of phrases, words, or even parts of

words (Romsdorfer & Pfister, 2007). In such multilingual cultural settings, *Polyglot*<sup>1</sup> is a wrapper written in Python for Compact Language Detector 2 (CLD2) designed by Google. It features a Naïve Bayesian classifier, using three different token algorithms based on different types of features. For Unicode texts like Greek and Thai that map one-to-one to detected languages, the script defines the result. For texts written with the aid of 80,000+ characters like the Han script and other large vocabulary languages, single letters (unigrams) provide the scoring. For all other texts, sequences of four letters (quad-grams) are scored.

## 2.2. *Langdetect*

*Langdetect*,<sup>2</sup> a Python implementation of Language-Detection (Choi & Wang, 2021; Shuyo, 2010), is based on pre-generated language profiles from Wikipedia abstract XML<sup>3</sup>. Language detection of a text is achieved by using a Naïve Bayesian filter. It calculates the language probability from spelling features – an approach called Naïve Bayes with the character n-gram. Their improvements over other approaches include noise filtering and character normalisation.

## 2.3. *Guesslang*

*Guesslang*,<sup>4</sup> a Python library for programming language detection of a given text within a Python program, uses a deep learning TensorFlow model built with 1,900,000 unique source code files. It is a Deep Neural Network classifier combined with a Linear classifier and heuristics that rely on the language character set and trigrams. This approach is extremely quick in execution as it requires lower computations, but with good performance detection rates. It was designed by Jacob R Rideout for K Desktop Environment (KDE)<sup>5</sup> and is an integral part of the KDE distribution package.

## 2.4. *FastText*

*FastText*<sup>6</sup> is a library for text classification (Joulin *et al.*, 2017) designed by Facebook AI Research labs. In this approach, text classification is achieved by representing sentences as Bag-of-Words (BoW) and by training a linear classifier. It is based on a technique called Continuous Bag-of-Words (CBow), which enhances the BoW approach, by using a continuously distributed representation of the context (Mikolov *et al.*, 2013). This approach is often on par with deep learning classifiers in terms of accuracy and many orders of magnitude faster for training and evaluation.

## 2.5. *SpaCy*

*SpaCy*<sup>7</sup> is a free, open-source library based on RoBERTa (Liu *et al.*, 2019) for advanced Natural Language Processing (NLP) written in Python. Like most deep learning algorithms, it is designed specifically for production use and helps you build applications that process and “understand” large volumes of text. In this study, we used the pretrained approach with an a priori given corpus, but it can also be trained on load, with a user-defined dataset.

## 2.6. *Pyld3*

*Pyld3*<sup>8</sup> is another neural network model for language identification (Ooms, 2022), developed by Google. It extracts character n-grams from the input text and computes the fraction of times each

<sup>1</sup> <https://docs.roperosci.org/cld2/>

<sup>2</sup> <https://pypi.org/project/langdetect/>

<sup>3</sup> [https://en.wikipedia.org/wiki/Wikipedia:Database\\_download](https://en.wikipedia.org/wiki/Wikipedia:Database_download)

<sup>4</sup> [https://guesslang.readthedocs.io/\\_/downloads/en/latest/pdf/](https://guesslang.readthedocs.io/_/downloads/en/latest/pdf/)

<sup>5</sup> <https://kde.org/>

<sup>6</sup> <https://github.com/facebookresearch/fastText>

<sup>7</sup> <https://spacy.io/>

<sup>8</sup> <https://github.com/google/cld3/> - accessed on Oct. 12, 2022.

of them appears. Then the embeddings of each n-gram type are averaged according to the fractions, and then they are concatenated to produce the embedding layer. The remaining components of the network are a hidden rectified linear layer and a SoftMax layer.

### 3. Dataset and Method

This section starts by describing the main source of information that was used for assessing the language detection models' performances – specifically a collection of tweets. Furthermore, the architecture of the novel language detection system is presented describing the stages it implements.

#### 3.1. Dataset

This study is based on a large-scale COVID-19 Twitter chatter dataset that was collected between January 1, 2020, and September 4, 2022, and is actively maintained by Banda (Banda *et al.*, 2021). This dataset contains over 1.3 billion tweets and growing. From this collection of tweets related to COVID-19, we extracted a Twitter collection of 433357 Tweets that were tagged by Twitter as tweets written in the Romanian language. Even though Twitter introduced a language detection and a language tag decorator for its tweets in 2013 (Roomann-Kurrik, 2013), its accuracy is low. Therefore, a tweet language detection analysis is necessary so that a proper Romanian language dataset is achieved.

We present a comparative analysis between multiple options for pre-processing our Twitter collection, which we implemented and tested. Banda's public dataset comprises only tweet IDs. Therefore, a tweet hydration process had to be implemented in the first place. The Tweeter API v2<sup>9</sup> was used in this hydration process. Secondly, the raw results of the hydration process needed to be cleaned and then verified.

#### 3.2. Method

Even though we are interested in raw tweets, the collected information from online is far from being usable directly, even if it looks human-readable at first sight. Therefore, cleaning and filtering processes were applied to the initial collection of data, in order to get rid of unnecessary information. The cleaning process consisted of the removal of formatting rules like HTML and JSON tags. Moreover, in the cleaning process, special language characters, like diacritics were replaced with non-diacritic equivalents. The filtering process comprised removing non-essential syntax.

The developed architecture of the RoLanD (*Romanian Language Detection*) method is depicted in Fig. 1, where the circles represent the data at various stages and the arrows represent the processes the data went through from one stage to another. The tweets' hydration process together with the cleaning and filtering processes described in previous sections belong to the data pre-processing stage.

The Language detection process involved a neural network model for language classification. Note that LI systems fall short when working with short texts or in domains with unconventional spelling, such as Twitter or other social media (Jaech *et al.*, 2016).

The language inference code extracts character n-grams from the input texts and creates a tf\*idf matrix. The n-gram size is in the interval [2,4]. For each n-gram, a hashed id is assigned and put into a vector. An average of occurrences for each n-gram is calculated. These averaged values are put into the embedding layer of the algorithm. The hidden layer contains a rectified linear activation function. The final layer is the SoftMax function which maximizes the predictions based on the highest prediction values.

<sup>9</sup> <https://developer.twitter.com/en/support/twitter-api/v2>

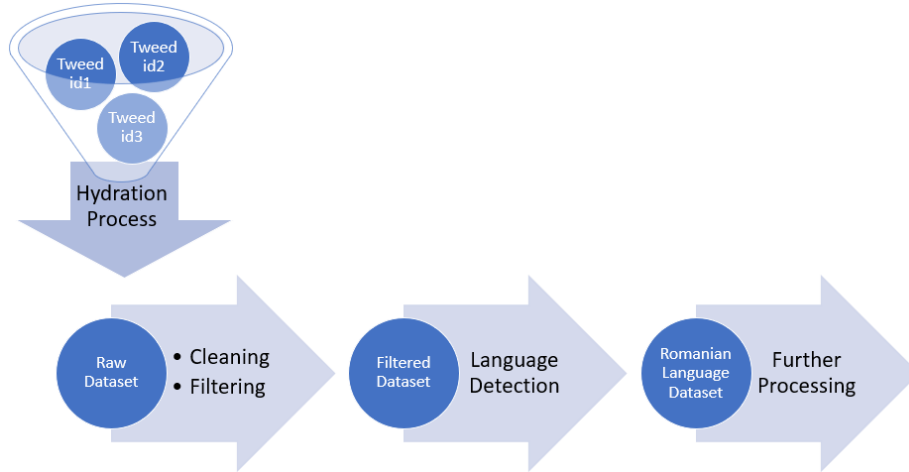


Figure 1. RoLanD Language Detection System Architecture

The algorithm travels the network once and returns the result to the language detection function. The further processing stage is thought of as a process that can be attached to the existing implementation as a loosely coupled process that would benefit from our results – the dataset.

#### 4. Results

After running the seven models over our newly created COVID-19 Romanian Tweeter Chatter dataset, the metrics obtained were put in Table 1.

Table 1. Text Language Identification Algorithms' Performance

| Model          | Precision (%) | Recall (%)   | F1 Score (%) | Accuracy (%) |
|----------------|---------------|--------------|--------------|--------------|
| Polyglot       | 95.41         | 33.13        | 49.18        | 72.41        |
| Langdetect     | 95.39         | 42.98        | 59.26        | 88.07        |
| Guess-language | 95.53         | 33.10        | 49.16        | 98.45        |
| FastText       | 95.59         | 25.53        | 40.29        | <b>98.82</b> |
| SpaCy          | 95.48         | 43.01        | 59.31        | 97.96        |
| Pycld3         | 95.62         | 22.17        | 36.00        | 98.98        |
| <b>RoLanD</b>  | <b>96.60</b>  | <b>74.73</b> | <b>84.27</b> | 97.37        |

As depicted in Table 1, RoLanD, our model, has the best F1 score by a large margin, and thus it proves that our classifier model surpasses existing approaches. Furthermore, even though accuracy could be improved, with FastText having the most accurate classification, it is a rule of thumb that says that the weighted average of the F1 score is to be used in comparing classification models, rather than global accuracy.

#### 5. Conclusions and Future Work

We conclude that one method for language detection cannot be a panacea in all scenarios. Therefore, just one classical machine learning technique is not enough for various tasks, instead, a hybrid approach that combines classic techniques looks like a more realistic approach.

Future improvements in our approach can imply the use of a neural network in order to optimally determine the best combination of algorithms and feature sets for a given task, based on the imposed constraints.

In future work, as we saw that Twitter does not identify tweets' language correctly, we also intend to hydrate the non-Romanian tagged tweets and then run the language detection algorithm RoLanD again, so that we obtain a larger consistent dataset.

### References

- Afify, Esraa & Sharaf Eldin, Ahmed & Khedr, Ayman & Kamal Alsheref, Fahad. (2019). User-Generated Content (UGC) Credibility on Social-Media Using Sentiment Classification. *Information Bulletin*, Faculty of Computers and Information, Helwan University, Vol 1, 1(1): 1-19. DOI: 10.21608/fcihib.2019.107506
- Armand, Joulin & Grave, Edouard & Bojanowski, Piotr & Mikolov, Tomas. (2017). Bag of Tricks for Efficient Text Classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, Vol. 2, Short Papers, 427–431, Valencia, Spain.
- Arne, Roomann-Kurrik. (2013). Introducing New Metadata for Tweets. [https://blog.twitter.com/developer/en\\_us/a/2013/introducing-new-metadata-for-tweets](https://blog.twitter.com/developer/en_us/a/2013/introducing-new-metadata-for-tweets)
- Banda, Juan M. & Tekumalla, Ramya & Wang, Guanyu & Yu, Jingyuan & Liu, Tuo & Ding, Yuning & Artemova, Katya & Tutubalina, Elena & Chowell, Gerardo. (2021). A Large-Scale COVID-19 Twitter Chatter Dataset for Open Scientific Research - An International Collaboration. *Epidemiologia*. 2(3):315-324. DOI: 10.3390/epidemiologia2030024.
- Boididou, Christina & Papadopoulos, Symeon & Zampoglou, Markos & Apostolidis, Lazaros & Papadopoulou, Olga & Kompatsiaris, Ioannis. (2018). Detection and Visualisation of Misleading Content on Twitter. *International Journal of Multimedia Information Retrieval*, Vol. 7, 71-86. DOI: 10.1007/s13735-017-0143-x.
- Choi, Keunwoo & Wang, Yuxuan. (2021). Listen, Read, and Identify: Multimodal Singing Language Identification of Music. In *Proc. of the 22nd Int. Society for Music Information Retrieval Conference* - <https://archives.ismir.net/ismir2021/paper/000014.pdf>
- Esteva, Andre & Kale, Anuprit & Paulus, Romain & Hashimoto, Kazuma & Yin, Wenpeng & Radev, Dragomir & Socher, Richard. (2021). COVID-19 Information Retrieval with Deep-Learning Based Semantic Search, Question Answering, and Abstractive Summarization. *npj Digital Medicine* 4:68. DOI: 10.1038/s41746-021-00437-0.
- Herold, Christian & Rosenfahl, Jan & Vanvinckenroye, Joris & Ney, Hermann. (2021). Data Filtering using Cross-Lingual Word Embeddings. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 162-172.
- Jaech, Aaron & Mulcaire, George & Hathi, Shobhit & Ostendorf, Mari & Smith, Noah. (2016). A Neural Model for Language Identification in Code-Switched Tweets. In *Proceedings of the Second Workshop on Computational Approaches to Code Switching*, 60–64, Austin, TX.
- Jauhiainen, Tommi & Lui, Marco & Zampieri, Marcos & Baldwin, Timothy & Linden, Krister. (2019). Automatic Language Identification in Texts: A Survey. *Journal of Artificial Intelligence Research*, Vol. 65, 675-782. DOI: 10.1613/jair.1.11675.
- Liu, Yinhan and Ott, Myle and Goyal, Naman and Du, Jingfei and Joshi, Mandar and Chen, Danqi and Levy, Omer and Lewis, Mike and Zettlemoyer, Luke and Stoyanov, Veselin (2019). RoBERTa: A Robustly Optimised BERT Pretraining Approach. DOI: 10.48550/arXiv.1907.11692.
- Lui, Marco & Baldwin, Timothy (2012). langid.py: An Off-the-shelf Language Identification Tool. *Proceedings of the ACL 2012 System Demonstrations*, Jeju Island, Korea, Association for Computational Linguistics, 25-30.
- Mhlanga, David. (2022). The Role of Artificial Intelligence and Machine Learning Amid the COVID-19 Pandemic: What Lessons Are We Learning on 4IR and the Sustainable Development Goals. *International Journal of Environmental Research and Public Health*, 19, 1879. DOI: 10.3390/ijerph19031879.
- Mikolov, Tomas & Chen, Kai & Corrado, G.s & Dean, Jeffrey. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of Workshop at ICLR*. 2013. DOI: 10.48550/arXiv.1301.3781.
- Moillic, Jade & Fawaz, Hassan Ismail. (2022). Language Identification for Very Short Texts: A Review, <https://medium.com/besedo-engineering/language-identification-for-very-short-texts-a-review-c9f2756773ad>

- Ooms, Jeroen (2022). cld2: Google's Compact Language Detector 2. <https://docs.ropensci.org/cld2/> (docs) <https://github.com/ropensci/cld2> (devel) <https://github.com/cld2owners/cld2> (upstream).
- Ooms, Jeroen (2022). cld3: Google's Compact Language Detector 3. <https://docs.ropensci.org/cld3/>
- Romsdorfer, Harald & Pfister, Beat. (2007). Text Analysis and Language Identification for Polyglot Text-to-Speech Synthesis. *Speech Communication*, 49(9):697-724. DOI: 10.1016/j.specom.2007.04.006
- Shuyo, Nakatani (2010). Language Detection Library for Java. <http://code.google.com/p/language-detection/>.
- Väisänen, Tuomas & Järv, Olle & Toivonen, Tuuli & Hiippala, Tuomo. (2022). Mapping Urban Linguistic Diversity with Social Media and Population Register Data. *Computers, Environment and Urban Systems*, Vol. 97. DOI: 10.1016/j.compenvurbsys.2022.101857.
- Vertovec, Steven (2007). Super-diversity and its Implications. *Ethnic and Racial Studies*, 30(6): 1024-1054, DOI: 10.1080/01419870701599465.
- Wong, Adrian & Ho, Serene & Olusanya, Olusegun & Antonini, Marta Velia & Lyness, David. (2021). The Use of Social Media and Online Communications in Times of Pandemic COVID-19. *Journal of the Intensive Care Society*, 22(3):255-260. DOI:10.1177/1751143720966280.
- Zampieri, Marcos & Ciobanu, Alina Maria & Dinu, Liviu. (2017). Native Language Identification on Text and Speech. In *Proceedings of the 12th Workshop on Innovative Use of NLP for Building Educational Applications*, 398-404, Copenhagen, Denmark.



# CORRECTING POSTAL ADDRESSES WRITTEN IN ANY LANGUAGE

GEORGIANA DRUȚĂ<sup>1</sup>, ADRIAN IFTENE<sup>1</sup>,  
AND DANIELA DÂNCIULESCU<sup>2</sup>

<sup>1</sup>*Faculty of Computer Science, “Alexandru Ioan Cuza” University, General Berthelot Street, 16, Iasi, Romania*

<sup>2</sup>*Faculty of Sciences, University of Craiova, Alexandru Ioan Cuza Street, 13, Craiova, Romania*

## Abstract

In recent years, applications that allow searching for words or their translation on the Internet are more and more advanced. Depending on the context, they realize that we have written a wrong word and suggest its correct form, and give us the results with this corrected form. In the case of postal addresses, correcting and obtaining a form that corresponds to a real address is of major importance for postmen or for distributors of products ordered on the Internet. The good performance of their activity depends on the quality of such a component and this can influence whether or not the ordered products reach the person who ordered and paid for them. The purpose of this work is to help users who frequently shop online, but also courier and distribution companies. The proposed solution consists in signaling the insertion of a wrong address and auto-completion with the correct address.

## 1 Introduction

Entering an invalid address in an online form intended for online shopping or for creating a new account is a problem often encountered these days when many of our activities have moved to the online environment. The problem is that most of the time you only notice this after you place the order or when you are called to confirm the delivery address. In Romania, it is not such a serious problem to fill out a field in a form incorrectly, but *what would happen when you fill out a delivery form in the United States of America where you are not called to verify the address, but the package is left in front of the door?*

An address can be written incorrectly in many ways: disregarding the meaning of the address fields (inverting the county with the city), typos (fuzziness), and addresses with codes (“IS” instead of “Iași”), addresses with acronyms (“Jud. Iași” instead of “Iași”), addresses with alternative names, addresses with missing fields, non-existent addresses (addresses with a city that is not found in the specified county) or conflicting addresses (addresses with two or more counties). These are just a few examples of invalid addresses. The challenge is even greater when an invalid address can be corrected in several ways. And to decrease the degree of complexity, the algorithm should correct addresses on a large scale, for all countries, and for any language.

## 2 Named Entity Recognition and Classification

Named entity recognition is a sub-problem of information extraction and involves structured processing or unstructured document processing and identifying expressions that refer to peoples, places, organizations, companies (Cimiano and Volker, 2005), and addresses (Yassine et al., 2020). Named entity recognition involves two activities. Firstly identifying proper names in the text and secondly classifying these names into a list of predefined interest categories such as *names of people, organizations* (companies, government, committees, etc.), *locations* (villages,

cities, counties, countries, mountains, rivers, etc.), *date* and *time* expressions, etc. For humans, named entity recognition is pretty intuitive because many entity names are proper names and most of them start with capital letters and can be easily recognized, but making an app for that is much harder. A possible solution could be to use dictionaries or Wikipedia (Iftene and Balahur-Dobrescu, 2008), but it is not an ideal solution because proper nouns are created continuously. Assuming we had the ability to enter all proper nouns into the dictionary, we would have a very hard time deciding the meaning of those words. Ambiguity is the most common problem in this field because a word can have several meanings depending on the context. To illustrate this problem, we will take two examples: “Paris meets her friends for her birthday.” and “Last weekend I was in Paris.”. From the presented contexts, it appears that the first example refers to a person and the second – to a location. The purpose of named entity recognition is to extract and classify names into specific categories by observing word meanings.

In recent years, the automatic recognition of name-type entities and their classification has become one of the most popular research fields that have a considerable number of studies on the development of such systems. They can be classified into three classes of algorithms, namely, (1) *hand-crafted rule-based* named entity recognition, (2) *machine learning-based* named entity recognition, and (3) *hybrid* entity recognition (Goyal, 2021). *Handcrafted rule-based named entity recognition* uses a lot of human-made rules (Hamza et al., 2003; Nadkarni et al., 2011; Iftene et al., 2011). In general, systems consist of a set of patterns that use *grammatical rules* (e.g., part of speech), *syntactic rules* (e.g., word precedence), and *orthographic features* (e.g., capitalization) in combination with a *dictionary*. The main disadvantages of this approach are the lack of portability and robustness, plus the high cost of slightly modified data. *Machine learning-based named entity recognition* is based on converting the identification problem into a classification problem, using the (Nadkarni et al., 2011) statistical model. This approach looks for patterns and relationships in the text to make a model using statistical models as well as learning algorithms. The systems identify and classify nouns into particular classes, such as *names of persons*, *locations*, or other entities. Machine learning can be of two types: (1) *supervised* and (2) *unsupervised*. *Supervised machine learning* involves using a program that can learn to classify a given set of tags that are based on the same features<sup>1</sup>. This approach requires the preparation of training data to build a statistical model, but it cannot achieve good performance without a large amount of training data. *Unsupervised machine learning* is a learning model where an unsupervised model learns without feedback<sup>1</sup>. Unlike the rule-based method, these types of machine learning can be easily ported to different domains or languages. *Hybrid entity recognition* is an approach that combines rule-based and machine-learning methods<sup>1</sup> (Păiș et al., 2019). Thus, new methods are created that have the best points of each of the two methods. Although this type of approach can achieve a better result than other approaches, it has a disadvantage, namely, the same weakness of the based method, the need to change the data domain.

## 2.1 Approaches and Solutions to Similar Problems

### Detecting Location Entities Mentioned in Twitter Messages

Unlike the name entity recognition task, this task goes much deeper: classifying locations detected in Twitter<sup>2</sup> messages into *city*, *province/state*, and *country names* to map to physical locations (Inkpen et al., 2017). The task approach consists of two stages. In the first step, conditional random field (CRF) models are trained that are able to detect the locations mentioned in the messages. They train three classifiers: one for cities, one for provinces/states, and one for countries, with various feature sets. In the second stage, the remaining ambiguities are resolved, i.e., cases where

<sup>1</sup> <https://www.coursera.org/articles/types-of-machine-learning>

<sup>2</sup> <https://twitter.com/>

there are several places with the same name. For this problem, a set of heuristics capable of choosing the correct physical location in these cases was proposed. The model thus created will allow a social network monitoring system to visualize the places mentioned in Twitter messages on a world map or to calculate statistics about locations.

### Improving Named Entity Disambiguation Using Wikipedia Entity Binding

Named entity disambiguation (NED) is the task of mapping words of interest, such as the names of people, companies, and locations, from a text document to their corresponding unique entities in a knowledge base (Parravicini, 2019). In this case, the target knowledge base is Wikipedia<sup>3</sup>. The most accurate solution to this problem was to use the concept of named entity congruence, or the relationship between named entities in the same context. Thus, using Wikipedia2Vec's 300d pre-trained entity embeddings, the proposed entity congruence method produced a gain of 4.2% to achieve a final accuracy of 76.3% on the AIDA CoNLL test dataset - YAGO (Seaton, 2020).

## 3 Proposed Architecture

The proposed architecture is based on two components, the back-end, and the front-end components.

### 3.1 The Back-end Component

#### Database

We chose to create a non-relational database using the files offered by geonames.org, a database that contains a tree for each country. Besides this, we used a mechanism provided by Java, the serialization of objects. After a serialized object has been written to a file, it can be read from the file and deserialized, meaning the type information and bytes representing the object and its data can be used to recreate the object in memory. Thus, we excluded the problem of the complexity of the insertion and deletion operations in this data structure.

#### The Chosen Data Structure

To make the search process more efficient, we decided to use a tree data structure for a good ranking of data. In the initial application, we created a 6-level tree. *Root* represents the whole world, *level 1* - continents, *level 2* - countries, *level 3* - cities, *level 4* - localities, *level 5* - streets, and *level 6* (leaves) - postal code. It should be noted that some levels may have different meanings for other countries. For example, for Romania, level 4 would represent localities, but for America it represents districts. The main disadvantages of the initial data structure were the complexity of searching for data in the tree.

Considering these disadvantages and the efficiency of searching for data in the tree, we have rethought the structure used. We decided to create a tree with 3 levels for each country: *level 1* - country, *level 2* - cities, and *level 3* - localities. Streets and zip codes are missing from the current tree because the Geonames database only has a few streets for each location and zip codes do not appear. The problem is very complex even without these fields (we will illustrate these aspects in one of the following sections). All the trees are saved in a set to avoid unnecessary data duplication. Obviously, the complexity of this structure is much better because there are far fewer nodes.

#### Adding Information to the Chosen Data Structure

To add data to the data structure for each country, we used the database provided by GeoNames<sup>4</sup>. The database we used contains archives with the data for each country, as well as text files or ar-

<sup>3</sup> <https://ro.wikipedia.org/>

<sup>4</sup> <https://download.geonames.org/export/dump/>

chives with *countries*, *alternative names*, *cities*, *postal codes*, etc. To create the final data structure, we did various types of “join” between these files, and we filtered or added information.

For added efficiency, we have created separate files of type *X.txt*, where *X* is the country code. The reason is that the number of lines in these files is greatly reduced, having a much better complexity. Thus, the search for a particular country's data is done in a file with only that country's locations and not all countries. *How do we identify a location if it is a country, county, or town with only the information in the files?* A location is *country*, if the feature class is “A” and the feature code is “PCLI”; *county*, if the feature class is “A” and the feature code, is “ADM1”; and *locality*, if the feature class is “A” or feature class is “P” and feature code is not “PCLI” and feature is not “ADM1”.

### 3.2 The Front-end Module

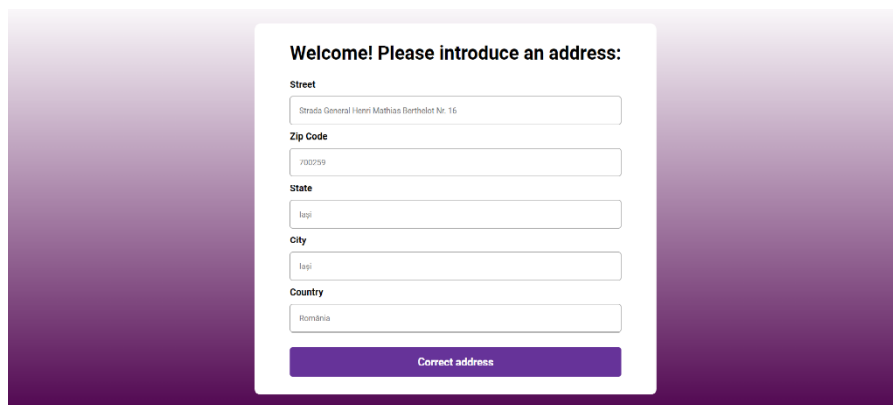
The front-end module is made using the React<sup>5</sup> framework. React (also known as React.js or ReactJS) is a free and open-source JavaScript library for building user interfaces based on UI components. It is maintained by Meta and a community of individual developers and companies. A React application is composed of modules, components, and services. The application contains three pages: the application presentation (introduction) page, the page with the form, and the page containing the top three addresses.

#### Introduction Page

It is the first component that any user interacts with. Its role is to provide information about the application and to give a clearer picture of the role of this application. On this page, the description contains details about the form fields, what kind of wrong addresses the algorithm can correct with high accuracy, and how the whole application flow can be achieved. On this page, there is only one button, the “Let’s start” button, which when pressed will redirect the user to the page with the form.

#### Form Page

This page contains a form for the user to fill out and a button with the message “Correct address” (see Fig. 1). The displayed form contains data in each placeholder field, serving as an example of what a correct address looks like. The user must fill in at least one field to be able to navigate to the next page where the chart is located, otherwise, they will get an error staying on the same page.



The screenshot shows a web form titled "Welcome! Please introduce an address:". It contains five input fields stacked vertically: "Street" (placeholder: "Strada General Henri Mathias Berthelot Nr. 16"), "Zip Code" (placeholder: "7000759"), "State" (placeholder: "Iasi"), "City" (placeholder: "Iasi"), and "Country" (placeholder: "Romania"). Below these fields is a purple button labeled "Correct address". The form is centered on a dark purple gradient background.

Figure 1. Form Page

<sup>5</sup> <https://reactjs.org/>

### Page with the List of Corrected Addresses

This page contains the top three best addresses after correcting the address submitted using the form (see Fig. 2). Also, this page contains a “Correct another address” button which when pressed will redirect to the form page.

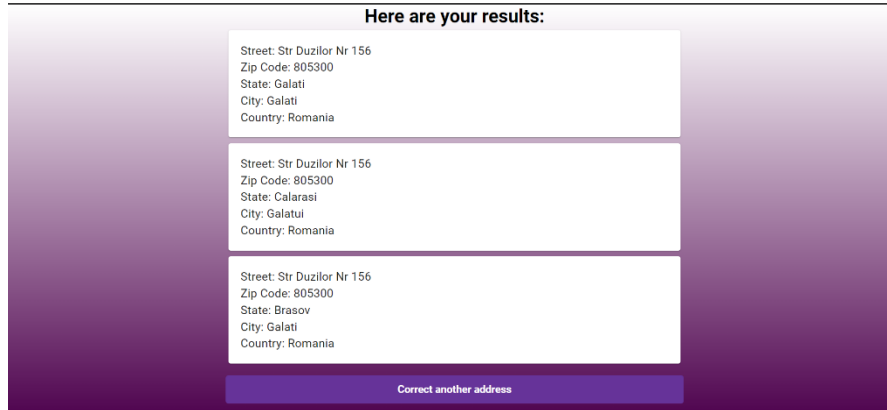


Figure 2. Top Addresses

### 3.3 Communication between Modules

In order to achieve communication between the two modules, we created an API at the back-end level using the Spring Boot<sup>6</sup> framework. The front-end module communicates with the user and the back-end module. Communication with the back-end module is done by sending HTTP requests to the API provided by the back-end. There is only one endpoint in the application, where the user is redirected by the front-end module after pressing the “Correct address” button on the form page. The application will in turn redirect to the page with the list of corrected addresses after receiving a list of three addresses.

## 4 Steps in the Process of Correcting an Address

In this section, we will present the algorithm built for correcting an address (see Fig. 3). In the next section dedicated to the tests, we will exemplify the improvements we made to the initial algorithm and their impact on the results we obtain.

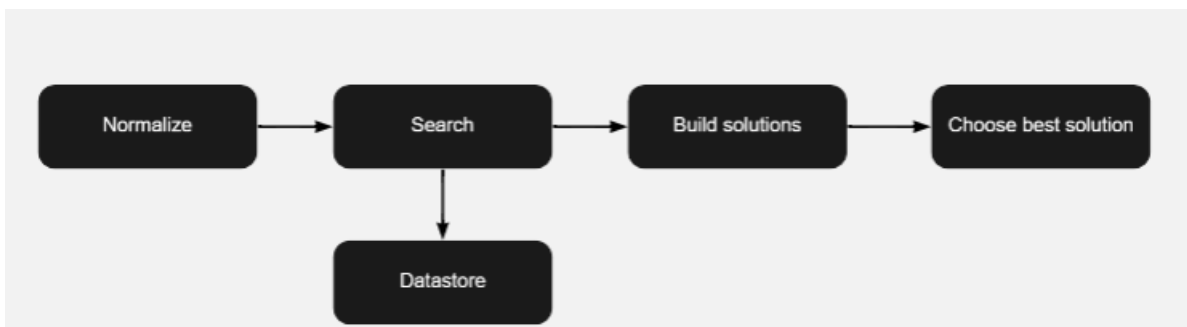


Figure 3. Steps in the Process of Correcting an Address

### 4.1 Conventions Considered

Since each country comes with its own specifications, in order to be able to correct postal addresses globally, it was necessary to choose a global convention for writing postal addresses. For example, Romania is divided into counties, cities, and towns (municipalities and villages),

<sup>6</sup> <https://spring.io/projects/spring-boot>

whereas Germany has administrative regions and districts. The convention chosen to exclude no country was *street, zip code, state, city, country*. Following the address correction algorithm, the address will only contain correct data in the city, state, and country fields because the database does not contain street and zip code data, which is why the street and zip code fields will remain unchanged.

#### 4.2 Data Normalization

With each country having its own specifications, we chose to bring each address to the simplest possible form before correcting it, a canonical form. For example, for Romania an address with the chosen convention may look like this: “Strada Mircea cel Bătrân, 237510, Olt, Vișina Nouă, Romania” (En: “Mircea cel Bătrân Street, 237510, Olt, Vișina Noua, Romania”), for Germany it could be “51 Zigeunereckweg, 74391, Baden-Württemberg, Erligheim, Bundesrepublik Deutschland” (En: “51 Zigeunereckweg, 74391, Baden-Württemberg, Erligheim, Federal Republic of Germany”) and for Japan “栃木県宇都宮市上横町 491-1218, 53638339, 大阪府, 栄町, 日本” (En: “491-1218 Kamiyokocho, Utsunomiya, Osaka, Sakaemachi, Japan”). Each received address is reduced to a canonical form: lowercase, replacing diacritics, replacing special characters (e.g. “!@€£¢\$¥%\*’()-.=:,+? {} []”), removing multiple spaces, removing spaces from the beginning and end of the address, replacing the “&” character with “and” and replacing the “o” character with “o” (“N” will become “No”). The final step in the normalization process is to convert addresses that contain at least one non-ASCII character in one of the fields to addresses that contain only ASCII characters. Checking whether the characters are ASCII or not was done with the help of the Guava<sup>7</sup> library. If the address does not contain only ASCII characters, then it will be searched in the database for the equivalent address with only ASCII characters. The search will be made in each alternate name list of the location with the name in the field, or in each alternate name list that contains the name in the current field. Thus, the new addresses will be able to be read and understood by any user who knows English. After the normalization process, the above addresses will have the following form: for Romania – “strada mircea cel batran, 237510, olt, visina noua, Romania”, for Germany – “51 zigeunereckweg, 74391, baden wurttemberg, erligheim, bundesrepublik Deutschland” and for Japan – “栃木県宇都宮市上横町 491-1218, 53638339, osaka fu, sakaemachi, japan” (in the case of Japan the street remains unchanged because the database used contains only state, city and country).

#### 4.3 Search Algorithm

To search for the value from a field in the address, it is necessary to go through all the trees. This operation has a low complexity because the database contains 252 trees and traversing a tree is done starting from the root node from top to bottom (if we know the country). To optimize this process, we used Guava’s multimap which allows having multiple keys with the same value, but (*key, value*) pairs have to be unique. The fact of having multiple keys with the same value is very important because we can have several localities with the same names from different countries (for example, Iași is the country seat of Iași County, as well as a locality in Brașov County). The multimap contains pairs of the form (*key, value*), where the *key* represents the name of the location (the primary name or one of the alternate names) and the *value* represents a pair of values. The pair contained by the value of the multimap is of the form (*key, value*), where the *key* is the principal name of the location with the name in the key of the multimap, and the *value* is the parent node for the same location. To avoid redundant operations, we used Guava’s SetMultimap to avoid duplicate (*key, value*) pairs. The advantages of this data structure are: inserting data without validating if the (*key, value*) pair already exists and saving storage space.

<sup>7</sup> <https://github.com/google/guava>

Thus, the code `“childNameParentMultimap.get(“podoleni”).stream().findFirst()”` will return `“Optional[podolenii de jos = AbstractLocation{geoNameId = 675809, name = ‘iasi’, asciiName = ‘iasi’, alternateNames = [iasi, iasi, iasi, judetul iasi, judetul iasi, ro213], code = ‘RO’, subRegions = [AbstractLocation{geoNameId = 685036, name = ‘beresti’, asciiName = ‘beresti’, alternateNames = [], code = ‘RO’, subRegions = [],...}]}”`. The output data shown for this code block is much more detailed, the list of subregions for Iasi has 598 localities, but to save space we decided to remember only the first element for a better understanding of the data structure. From the example above we deduce that “podoleni” is the alternative name for the locality “podolenii de sus” and belongs to the county “iași”. Having this information: the main name of the locality and its parent node, we can also find out the county of which the locality with the name “podoleni” belongs by accessing the parent node from the returned node: `“childNameParentMultimap.get(“podoleni”).stream().findFirst().get().getValue().get CountryRoot()”`. Full traversal of all trees is avoided because `“childNameParentMultimap.get(name)”` returns a set of pairs of the form *(location parent name, parent location)*. With this improvement, the costs have been greatly reduced.

#### 4.4 Data Structure

As mentioned before, the data structure contains a collection of trees with one tree for each country in the world. Each tree has 3 levels that represent the hierarchical order in an address: country, state, and city (the meaning of the levels from top to bottom, starting from the root node). The size of the data structure is very large, but nevertheless, a more detailed data structure provides even better precision. Therefore, when creating the data structure, we added other new alternative names to the list of alternative names provided by the Geonames<sup>8</sup> web application. New alternate names are variations of the location name and each alternate name. The variations consist of the most frequent spelling mistakes such as dropping prepositions, omitting writing with two equal characters next to each other, and dropping one or more vowels. In this way, we cover multiple invalid postal addresses. With such a data structure, the accuracy of the algorithm increases considerably.

##### Elimination of Prepositions

We considered any word less than three in length to be a preposition because we could not find a worldwide database of prepositions. Because of this, collisions occur. In tests to measure accuracy, some words removed from the address are not prepositions and the correction algorithm does not return the expected address. For example, for the address “Strada Sfântul Lazar No. 28, 77177, Ilfov, Stefanestii de Sus, Romania” the algorithm for generating test addresses without prepositions will return the address “Strada Sfântul Lazar No. 28, 77177, Ilfov, Stefanestii, Romania”. After the correction, the address returned by the algorithm is “Strada Sfântul Lazar No. 28, 77177, Ilfov, Stefanestii Noi, Romania”, although the user would have expected the address “Strada Sfântul Lazar No. 28, 77177, Ilfov, Stefanestii de Sus, Romania”. The function returns a list of all variations of a name without one or more prepositions. For example for “Stefanestii de Sus” it will return the list: [“Stefanestii”, “Stefanestii de”, “Stefanestii Sus”].

##### Removing Duplicate Characters

For the test address “51 Zigeunereckweg, 74391, Baden-Württemberg, Erligheim, Federal Republic of Germany” the deduplication algorithm will return the address “51 Zigeunereckweg, 74391, Baden-Württemberg, Erligheim, Federal Republic of Germany” with “Württemberg” with a single “t”. The best match search algorithm will return the original address, “51 Zigeunereckweg, 74391, Baden-Württemberg, Erligheim, Federal Republic of Germany”, as expected.

<sup>8</sup> <https://download.geonames.org>

For a more complex breakdown, we created a function that removes all duplicates from a given string and a function that creates a list of strings without one or more duplicate characters. For example for “Babbenhausen” the function will return the list: [“Babennhausen”, “Babbenhausen”, “Babenhausen”].

### Elimination of Vowels

For example, for the address “Strada Sfantul Lazar No. 28, 77177, Ilfov, Stefanestii de Sus, Romania” the test generation algorithm for addresses without one or more vowels will return the address “Strada Sfantul Lazar No. 28, 77177, Ilfv, Stefanest de Sus, Romania”. After the correction, the address returned by the algorithm is “Strada Sfantul Lazar No. 28, 77177, Ilfov, Stefanestii de Sus, Romania”, this is due to the insertion of all name variations without one or more vowels in the database. For example, for the name “Roma” in the list of alternative names will be added: “Roma”, “Rom”, “Rma”, and “Rm”. By covering these cases of postal address misspellings, the accuracy of the correction algorithm increases significantly.

## 4.5 Creating Solutions

### Data Interpretation

The first step in finding solutions is to interpret the data in the given address. For this, we looped through each field and searched for all substrings in it in the multimap. For the best possible accuracy, we chose to generate all substrings so as not to omit information. For example, for “Iasi Copou”, the list of substrings [“Iasi Copou”, “Iasi”, “Copou”] will be generated, of which only “Iasi” and “Copou” are found in the database. If we had only searched for the entire value in the field, “Iasi Copou”, we would not have had any valid information. By data interpretation, we mean the categorization of data into the city, state, or country. Thus, we created a shape map “Map < String, Map < String, Set < String >>> mapWithFieldsValue”, where the *key* of the map represents the name of the field (*street*, *zipCode*, *state*, *city*, *country*) and the *value* represents the classification of the data in the field. The map in the map value represents the category name (*state*, *city*, *country*) and a set of field values that fall into the category with the name in the key. To further understand the importance of this data structure, we will take an example of a valid address and create the map described in this subsection. For the address: “Str. Sfantul Lazar No. 28, 247752, Vâlcea, Voineasa, Romania”, the map created will be “{zipCode={country=[], city=[], state=[]}, country={country=[romania], city=[], state=[]}, city={country=[], city=[voineasa], state=[]}, street={country=[], city=[], state=[]}, state={country=[], city=[valcea], state=[valcea]}}”. For the street and zip code fields there are no existing substrings in the database, in other words, the street and zip code fields do not contain state, city or/and country. The country field contains “Romania” and, after identifying it in the database, it is classified as a country, which is true. In the city field, identify “Voineasa” as a city. And last but not least, in the state field, identify “Vâlcea” as state and city because it is a county as well as a county seat. The interpretation of data is done by searching the database for the corresponding node with the name in the field or the node with the name in the field as an alternative name. The checking data type (*city*, *state*, or *country*) is done using the Java function “instanceof”, so if the node is an instance of the City class, we infer that it is from the city category, if it is an instance of the State class, we can infer that it is from the state category, similarly for the country.

### Data Deduction

For even better accuracy in creating solutions and finding the best match, we chose to infer new information from the data received at the stage where the map data structure is created. Thus, for the address above, the new value in the map data structure will be: “{zipCode = {country=[], city=[], state=[]}, country = {country=[romania], city=[], state=[]}, city = {country=[roma-

nia\*], city = [voineasa], state = [valcea\*, olt\*]}, street = {country= [], city=[], state=[]}, state = {country = [romania\*], city=[valcea], state = [valcea, brasov\*, alba\*]} }".

For example, for the city field containing "Voineasa", "Voineasa" was categorized as a city and more information could be inferred (data ending with "\*" is inferred data) such as the county in which there is located the locality of Voineasa (this locality can be found in several counties), as well as the country. These inferences could only be made by finding all elements in the Set-Multimap with the "willful" key and extracting the data of interest (the name of the parent location of the current location and/or the name of the parent location of the parent of the current location as appropriate).

### Creating Solutions

Having this map data structure with data categorization and inference of other data based on the received data, we can create all possible solutions for the received address. Creating solutions requires reinterpreting data from the map data structure and assigning confidence scores to find the best address.

### Reinterpreting the Data and Obtaining the Confidence Score

We have created three intermediate lists that contain each category-ordered data so that the deduced data is in the last positions. For a country, the list will be: [romania, romania\*], for states: [valcea, valcea\*, olt\*, brasov\*, alba\*] and for city: [voineasa, valcea]. Based on this map data structure, we created three lists of pair values. We have created a list for the state, city, and country categories. The key of the pair represents the name of the location and the value represents the number of points assigned. Thus, for the map above and the intermediate lists we will have the minimum lists for the country list: [[romania, 6]], for the state list: [[valcea, 10], [olt\*, 5], [brasov\*, 4], [alba\*, 3]], and for the city list: [[voineasa, 9], [valcea, 1]]. The minimal form of the list involves removing from the list the pair for which the score is less than the score of another pair, where its key is equal to the key of the second pair (the comparison is made on the name of the location without "\*"). For example, for the list [[valcea, 10], [valcea\*, 8], [olt\*, 5], [brasov\*, 4], [alba\*, 3]], the minimum list will be [[valcea, 10], [olt\*, 5], [brasov\*, 4], [alba\*, 3]]. The score was calculated according to the type of data: information from the address or inferred information, respecting the meaning of the field or not, and, last but not least, the category of data.

We also took into account the data category because we considered that a city is more priority than a state, the reasoning behind it is that by adding a village or a smaller locality that probably few people know about, there is a high chance that it is just a mistake in completing the fields. In the list for the state field: Vâlcea has 5 points for respect of the field, 5 points for priority in the intermediate list, the Olt has 5 points for respect of the field, 3 for priority in the intermediate list, and -3 because it is inferred information. In the city list: Voineasa has 5 points for respecting the field, 3 points for being a small town, and 1 point for priority in the intermediate list, and Vâlcea has 1 point only for priority in the intermediate list. The list for the country field contains a single element [[romania, 6]] because there is only one country on the map and it is assigned a score of 6 because it has 5 points because "romania" is found in the correct "country" field and 1 more point that is the priority in the intermediate list for the country.

### 4.6 Choosing the Best Solution

Having the lists above, for the country list: [[romania, 6]], for the state's list: [[valcea, 10], [olt\*, 5], [brasov\*, 4], [alba\*, 3]] and for the city list: [[voineasa, 9], [valcea, 1]], we generated all possible addresses of the form state, city, country by calculating the score in advance and saving in a list all pairs of the form (address, score), where the address is valid. An address is valid if there is a match for the country-state-city concatenation in the database.

In order to make validation more efficient, we went through the tree from bottom to top. So, for the current city, we checked if its parent has the same first name as the state in the current address, if so, we continued the check for the country. If the country in the address matches the principal name of the parent state in the database, then the address is valid. After repeating this process for each address in the address list, the list of valid addresses is sorted in ascending order by score.

### Result

Thus, the ordered list of valid addresses for the above address is: [{Street: Str. Saint Lazarus No. 28, Zip code: 247752, State: valcea, City: voineasa, Country: romania}, 25], [{Street: Str. Saint Lazarus No. 28, Zip code: 247752, State: olt, City: voineasa, Country: romania}, 20], [{Street: Str. Saint Lazarus No. 28, Zip code: 247752, State: valcea, City: valcea, Country: romania}, 17], [{Street: Str. Saint Lazarus No. 28, Zip code: 247752, State: brasov, City: valcea, Country: romania}, 11], [{Street: Str. Saint Lazarus No. 28, Zip code: 247752, State: alba, City: valcea, Country: romania}, 10]]. So, after running the algorithm, the corrected resulting address is: “Street: Str. Sfantul Lazar No. 28, Zip code: 247752, State: valcea, City: voineasa, Country: romania” and corresponds to the address given to the algorithm because the initial address was already correct.

## 5 Accuracy and Performance Tests

To check the progress, we created test files for Romania, Germany, and Japan. Each file contains 100 correct addresses. we closely checked the accuracy of the algorithm and the duration of address correction during the development of the algorithm. Then, we created algorithms to generate, based on the file with correct addresses, variations of invalid addresses for cases where the meaning of the fields is not respected, addresses with all the data in one field, omitting the state, city, and/or country from an address, typos, multiple cities, states and/or countries. In the early stages, we created postal address files for each country, as well as algorithms to generate incorrect addresses, to keep a close eye on our progress. The objective we set was to reach an accuracy of 80-90%. In the section dedicated to the tests on Romania, we will detail each case of writing a wrong address, the changes made and the precision of the algorithm for that case, so that later we will only mention the resulting precision, possibly other new changes made.

### 5.1 Addresses from Romania

To check the accuracy and latency of the algorithm for addresses in Romania, we created a text file with 100 postal addresses for Romania and corrected the already correct addresses to see how many addresses corrected by the algorithm coincide with the original ones, generated by the web application “<https://generate.plus>”. In the original algorithm, where the database contained one tree which contained subtrees for each country of 5 levels, the time to correct 100 corrected addresses is about 3 minutes and 50 seconds (230 seconds). After changing the structure of the database, having a set of all trees for each country, each tree has 3 levels, and improving the correction algorithms, the time to correct the same addresses dropped to about 11 seconds. Regarding accuracy, it decreased significantly as we covered more cases of misspelling a postal address.

### Multiple Data in a Single Field

The same algorithm on the first run on tests with two values in the state field had a price of 100%, and on tests with two values in the city field, it had an accuracy of 93%. The drop in accuracy was due to the fact that the entire field input was the most prioritized followed by a sublist ordered in ascending order of word count, with the order in the list influencing the score given. After the descending ordering of the possible names in the city field, the precision for the case where we have two cities in the city field increased to 100%, remaining unchanged for the other two cases. Accuracy also remained unchanged for cases where the values of two or more fields

were in a single field, at 99%. Accuracy also remained the same for cases where two or more fields were filled in incorrectly.

### Fields Filled Wrong

For the case where the state is in the street field, the initial accuracy was 0%, but after the changes made to the algorithm, the accuracy for each case reached 99% (collisions occur). The changes involved the extraction of data from the street and zip code fields, and their interpretation. For example, for the address “7.252 Brennan Springs Braşov, 507.207, Vâlcea, Romania”, there are two solutions: (1) 7.252 Brennan Springs, 507.207, Braşov, Valcea, Romania, (2) 7.252 Brennan Springs, 507.207, Valcea, Râmnicu Valcea, Romania. The solution obtained by the algorithm is the address with the highest score, namely the second address. In the present case, the score of the second address is higher because Vâlcea is in the state field and it really is the state, in the city field the county residence is put.

### Blank Fields

The next set of tests was the one where we had no data in one of the fields: city, state or country. For the case where the country field was empty, the accuracy remained unchanged. For the test set with no states, the accuracy dropped to 74%, which was normal because most of the time localities do not have unique names and the algorithm will return the first state in the database for that locality in the city field. For the case where the city is missing, the precision dropped drastically, reaching 3%. Of course, a state has tens of thousands of localities and the chances of guessing the locality from the initial address without additional data about it are very small, almost non-existent. Most of the time the corrected address for this case has the same city state or county residence.

### Addresses with Alternate Names

To generate addresses where at least one alternate name appears, we created a multimap. Each (*key, value*) type pair in the multimap represents the location name and a list of alternative names. For example, the location Bucharest has several alternative names: [bucaresto, bucharesti municipuliu, bukareszt, buekres, bu jia lei si te, bucharest, boukourest, bukarest, bucaresta, bukarest, bucarest, bukarestas, bukulesyuti, bwqrst, ro321, bucharest, bukrest, boekarest, bukureshh, bkharst, bucuresti, bukarest, bukureshti, bwkharst, buh, bukaresuto, bukres, bukurest, bukureshta, bukuresht] (this is the normalized list of alternate names). For this location alone, the multimap will contain 30 (*key, value*) pairs. Each key represents an alternate name from the list above, and each value will be in this list. After the first run of the algorithm on the new test data with alternate names in the city field, the accuracy dropped to 9%. To also cover the cases with alternative names, we had to rethink the helper multimap data structure for traversing the tree from the bottom up.

The change made was to change the values in the multimap. Originally the multimap contained as values the parent node in the tree for the location with the name in the key, and now the values are in pairs. Each pair represents the primary name of the location with the name in the key and the parent node in the tree for the location with the name in the key. After this modification, the accuracy increased to 57%. Another thing that caused the low precision was that the list of alternate names also contained null or empty values. After removing the null and empty values, the accuracy increased to 92%.

### Addresses with Typos

The accuracy of the algorithm was also tested with addresses that contained the most common typos such as missing vowels, writing addresses without duplicate letters, and addresses without prepositions. We considered prepositions of any word with a length of fewer than three letters because each country has its own prepositions, not finding a list of them. So the resulting accuracies

were in this order: 92%, 97%, and 93%. For the case of removing duplicate characters, the precision remained as good because in Romania there are few or even no locations that contain two consecutive equal characters. Therefore, on another set of test data for another country, the accuracy would probably have been much lower. Also, in the case of removing prepositions, collisions can be much higher depending on the country. For example, for the normalized address: “6328 jay landing, 547484, mures, iod, romania”, the word “iod” will be removed because the algorithm cannot figure out the meaning of the word, it only knows its length. The address returned after correction will be “6328 jay landing, 547484, mures, mures, romania”). We mentioned that the addresses generated with typos given for testing vary from one run to another. Their form varies because all possible forms are generated for cases where duplicate characters are removed, one or more vowels are missing, and one or more prepositions are missing, and then an element is randomly drawn from these generated lists.

### **Levenshtein’s Distance Metric**

Wanting to improve the accuracy of the algorithm for fuzziness cases, we implemented the Levenshtein distance metric to find words as close as possible to the words in the given address. Initially, the algorithm corrected addresses without prepositions in less than a second with 93% accuracy. After adding Levenshtein’s 95% word similarity metric and some improvements, the best correction time on the same data set was 509 seconds (8.48 minutes), and the accuracy remained unchanged, by 93%. The time complexity of the algorithm decreased drastically because, for each word in the multimap, the similarity rate with a word or a string of words in the address is calculated. The similarity rate assumes the minimum number of single-character edits (i.e., insertions, deletions, or substitutions) required to change one word to another. We dropped this metric considering the fact that the database on which the algorithm runs contains only the data for Romania and has a very low time complexity and unchanged precision.

### **Accuracy after Canonical Form Modification**

The initial accuracy was 96% on the correct addresses in Romania and it dropped to 80% after adding the translation of the given addresses in the readable version by any user (ASCII characters only, in the case of Romanian addresses without diacritics, for Japan it would be the translation “トチギケン” in “Tochigi”). We improved the accuracy of the algorithm by checking whether the corrected address matches the original address in the correct address file or the translated address, thus, the accuracy increased from 80% to 93%, and the correction time increased from 5 seconds to 21 seconds. Another improvement of the algorithm is that we have turned the address into a readable one (containing only ASCII characters, an address written in English) only if at least one of the fields contains a non-ASCII character. Thus, it returned to the original accuracy, 96%, in 7 seconds.

### **Final Values**

In the experiments carried out, the accuracy was between 6 and 100. The low accuracy is due to the cases where the address does not contain the city. For example, for the address “str. sf. lazar, 88787, iasi, podolenii de jos, romania” after removing the city, the new address will look like this: “str. sf. lazar, 88787, iasi, , romania”. After correcting the address without city, the new corrected address will be “str. sf. lazar, 88787, iasi, iasi, romania” because the algorithm has no other information. All addresses without a city will have the city state or county seat as the city.

## **5.2 Addresses from Japan**

For addresses from Japan written in Japanese/Chinese, we decided to translate them into English. Each name is replaced by the corresponding name written in ASCII, a name is taken from the database. After adding this change the time to correct addresses for Romania increased from 3 sec-

onds to 7 seconds. We chose to keep this change despite the increased patch time. The main motivation is to correct addresses written in any language, and the second reason is to accurately measure accuracy. Following address correction, the returned address was not written in the original language and would have been considered wrong even if it was the same address written in another language. As a convention, we have chosen that all addresses received or returned by the algorithm are written in a language that can be understood by anyone, namely English. The accuracy of the algorithm decreased in the Japanese data set. The reasons can be multiple: loss of information by translating words into ASCII writing, missing data from the database, and incorrect generation of correct addresses by the web application used.

### 5.3 Addresses from Germany

For certain postal addresses in Germany, it was necessary to translate them into their English equivalent, they have certain special characters such as ö, ü, ä, and ß. The accuracy of the algorithm for the test set for Germany is lower than the accuracy for the data sets for Romania and Japan. Possible reasons could be loss of information due to address translation, generation of wrong addresses by the web application used, or lack of certain information from the data structure used.

## 6 Conclusions

In the present work, an application was presented that aims to offer users (both those who make purchases and service providers) help in correcting the address when they place or distribute orders. The average accuracy for correcting addresses within Romania is 89%, for Japan it is 78% and for Germany, it is 75%, with the mention that due to cases with typos, the specified accuracies are not the same for each run. These accuracies are indicative because they are determined by the data flow. For other test files, the accuracy would improve or not, but the speed of the algorithm would remain the same, about 11 seconds for correcting 100 addresses.

In the future, we propose to go in two directions (1) after correcting the address given through the form, the application should direct the user to a page containing the top three best matches on a map containing a pin for each of them, (2) we want to integrate the application with other online shopping platforms such as Emag, Altex, About You, Aliexpress, and many others, thus proving useful for a wider range of users around the world.

## References

- Philipp Cimiano and Johanna Volker. (2005). Towards large-scale, open-domain and ontology-based named entity classification. *International Conference Recent Advances in Natural Language Processing*, pages 166–172.
- Chirag Goyal. (2021). Step by step guide to master NLP – named entity recognition, Part 10. *Data Science Blogathon*.
- Oana Hamza, Kalina Bontcheva, Diana Maynard, Valentin Tablan, and Hamish Cunningham. (2003). Named entity recognition in Romanian using GATE. In *RANLP 2003 Workshop on information Extraction for Slavonic and other central and eastern European languages*.
- Adrian Iftene and Alexandra Balahur-Dobrescu. (2008). Named entity relation mining using Wikipedia. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08), 28-30 May, Marrakech, Morocco*.
- Adrian Iftene, Diana Trandabăţ, Mihai Toader, and Marius Corici. (2011). Named entity recognition for Romanian. In *Proceedings of the 3th Conference on Knowledge Engineering: Principles and Techniques Conference (KEPT2011). Studia Universitatis, Babes Bolyai, Cluj-Napoca, Romania, July 4-6, 2011*, 2:19–24.
- Diana Inkpen, Ji Liu, Atefeh Farzindar, Farzaneh Kazemi, and Diman Ghazi. (2017). Location detection and disambiguation from Twitter messages. *Journal of Intelligent Information Systems*, 49:237–253.

- Prakash M. Nadkarni, Lucila Ohno-Machado, and Wendy W. Chapman. (2011). Natural language processing: an introduction. *Journal of the American Medical Informatics Association*, 18(5):544–551.
- Alberto Parravicini. (2019). Named entity disambiguation with knowledge graphs. *Blogs Oracle AI*.
- Vasile Păiș, Dan Tufiș, and Radu Ion. (2019). Integration of Romanian NLP tools into the relate platform. In *Proceedings of the International Conference on Linguistic Resources and Tools for Processing Romanian Language – CONSILR 2019*, pages 181–192.
- Will Seaton. (2020). Improving named entity disambiguation using entity relatedness within Wikipedia. *Towards Data Science*.
- Marouane Yassine, David Beauchemin, Francois Laviolette, and Luc Lamontagne. (2020). Leveraging Subword Embeddings for Multinational Address Parsing. *arXiv*.

# SPLITTING HASHTAGS IN ROMANIAN MICRO-BLOGGING TEXTS

ROXANA MICU, CAROL LUCA GASAN, AND VASILE PĂIŞ

*Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, Bucharest, Romania*

## Abstract

Hashtags are commonly used in social media texts to facilitate message grouping and searching. They can be formed from single words, but usually contain multiple words or expressions glued together into a single hashtag. When used inside sentences, the words that are part of the hashtag also become part of the sentences. This poses unique challenges for processing micro-blogging text since a valid text processing pipeline must take into account splitting the hashtags into the constituent words. In this paper, we introduce such an algorithm for splitting hashtags, adapted to the Romanian language. To improve the performance, we integrate resources extracted from the Representative Corpus of Contemporary Romanian Language (CoRoLa).

## 1 Introduction

Social media is a collective term for websites and applications that focus on communication, community-based input, interaction, content-sharing, and collaboration. Social media platforms allow users to have conversations, share information, and create web content. There are many forms of social media, including blogs, micro-blogs, wikis, social networking sites, photo-sharing sites, instant messaging, video-sharing sites, podcasts, widgets, virtual worlds, and more.

Microblogging content represents short text messages, sometimes containing embedded multimedia content, designed for quick interaction between social media users. Typical microblogging platforms include Twitter, Reddit, and Gab. However, microblogging content can be found on other social media platforms as well. A microblog also allows direct interaction by means of replies or comments.

A hashtag is a word or keyword phrase, usually found within social media posts, that is preceded by a hash symbol (#). By searching for keywords or particular hashtags, a user can find the specific topics they are looking for. Hashtags help to draw attention to posts and encourage interaction. When using a phrase as a hashtag, it is written without spaces, such as "#usinghashtags-forever". It can include numbers but not symbols (apart from underscores) or punctuation. The hashtag can be placed at the beginning, middle, or end of the social media post or comment, and it allows better indexing by the social media network. The hashtag can be either separated from regular text (for example added at the end) or made part of the text (words inside a sentence are transformed into hashtags). With this strategy, people who do not follow a user can still find their posts by looking at related hashtags, following links from other users.

Our interest in splitting Romanian hashtags arises from the creation of MicroBloggingNERo<sup>1</sup> (Păiş et al. 2022), a manually annotated corpus for named entity recognition in Romanian micro-blogging texts. The corpus contains a number of named entities that are represented entirely by hashtags or expressions containing hashtags. This is only a part of the challenges faced with annotating the corpus, as described in Păiş et al. (2022). Furthermore, during automatic processing of the corpus, for basic operations such as lemmatization and part-of-speech tagging, we observed that hashtags were not handled properly, due to their internal multi-word structure.

---

<sup>1</sup> <https://doi.org/10.5281/zenodo.6905234>

In terms of our contribution, we introduce an algorithm that splits each hashtag into the words that compose it (for example, "#everythingiscool" is going to be split into "everything" + "is" + "cool"). We provide resources extracted from the Representative Corpus of Contemporary Romanian Language (CoRoLa) used for improving the algorithm's splitting capabilities and the ability to identify different entities included in the hashtag's content (such as "bucharest" in "#lovebucharestRO"). This paper is structured as follows: Section 2 presents related work, then Section 3 presents the method used for hashtags splitting, with the algorithm in Section 3.1, and additional language resources in Section 3.2. Section 4 presents the experiments performed with the proposed algorithm, and we conclude in Section 5.

## 2 Related Work

Çelebi and Özgür (2016) analyze three baseline methods for hashtag splitting: HMM (Hidden Markov Models), Norvig's Naive Bayes, and Microsoft's Word Breaker. The training data was automatically generated from the SNAP Stanford Twitter dataset (Leskovec and Krevl, 2014), which presents the idea of a fourth baseline method named SNAP n-grams. Word Breaker provided the best results, having an F1-score of approximately 84.6% on the Stanford dataset.

Reuter and Kalita (2016) propose methods of attributing scores to a hashtag's split in English such as Maximum Known Matching (MKM), Maximum Unknown Matching (MUM), Two grams (2GM), and POS tagging – AH (tag with ARK, assign probabilities with HMM). For the base algorithm, two methods were tested: the pipeline method and the hill-climbing method. The former proved to yield better results, with an F-score of 82.2% (compared to the 76.5% obtained with the latter method). Evaluated on a dataset with around 1,000 manually segmented hashtags pulled from Rovereto's Twitter 1-gram data, the model had the following precision for each of the named scoring methods: MUM – 90.1%, MUM – 91.6%, 2GM – 92.1% and AH – 92.1%.

Another representative dataset that consists of segmented hashtags in English is HashSet (Kodali et al., 2022). It contains hashtags that are split both manually and automatically (by using the camel case cues). All data was obtained using the Twitter API.

Somewhat similar to the task of splitting hashtags is the compound splitting problem in different languages, such as the one for German presented by Weller-Di Marco (2017). This paper offers a resource-poor approach that relies on a basic frequency-based approach and a form-to-lemma mapping to approximate morphological operations. Evaluated based on SMOR (a comprehensive German finite-state morphology covering inflection, derivation, and compounding) and with GermaNet (Henrich and Hinrichs, 2010) as the gold dataset, the model has a precision of 93.93%.

Several micro-blogging datasets were created for the Romanian language with the purpose of achieving various tasks (especially classification), however without considering the hashtag splitting issue. For example, ROFF (Manolescu and Çöltekin, 2021) is a Romanian Twitter dataset for offensive language. It contains 5,000 micro-blogging posts annotated by volunteer annotators. The reported agreement for the first level annotation ("is the tweet offensive or not?") is 86.70% (Cohen's Kappa  $k = 0.52$ ), for the second level ("is the tweet targeted or not?") – 58.30% ( $k = 0.17$ ), and for the third level ("who is the target of the tweet?") – 43.20% ( $k = 0.20$ ).

Maddela et al. (2019) introduce a dataset of 12,594 English hashtags split into individual segments and proposes a set of approaches for hashtag segmentation by framing it as a pairwise ranking problem between candidate segmentations. They propose a multi-task learning approach that jointly learns segment ranking and single- vs. multi-token hashtag classification.

Rodrigues et al. (2021) develop a zero-shot hashtag segmentation framework and demonstrates accuracy improvement of multilingual sentiment analysis pipelines. The framework combines two publicly available and pre-trained LMs, GPT-2 and BERT, and uses beam search and re-ranking. It establishes the new state-of-the-art results for several hashtag segmentation datasets. Halgamuge et al. (2021) propose an algorithm to segment hashtags, prioritizing the optimization

of computation time. They create candidates according to a given corpus, containing 1-gram and 2-gram data. Each candidate's probability is calculated either using the Viterbi algorithm (for the 2-gram) or the Naïve Bayes algorithm (for the 1-gram).

### 3 Method

For the purposes of this work, we introduce an algorithm enhanced with specific capabilities for the Romanian language, as described in Section 3.1. Subtle differences such as the presence of diacritics call for methods of adapting the input to fit the language. The algorithm works similarly to an approach of separating compound or concatenated words. In order to obtain data for training, 1,213 hashtags from the MicroBloggingNERo dataset were manually segmented, 508 of which were fully lowercase and 90 of them fully uppercase.

#### 3.1 Hashtag Splitting Algorithm

The algorithm looks to solve the main issue of dividing a string of characters into smaller sub-strings that represent words in Romanian. Mathematically, this approach can be described in the following way: considering a hashtag  $H = c_1, c_2, \dots, c_n$ , where  $c_i$  is a UTF-8 character, the algorithm will produce the sub-strings  $W_1 = c_1, c_2, \dots, c_{n_1}$ ,  $W_2 = c_{n_1+1}, c_{n_1+2}, \dots, c_{n_2}, \dots, W_m = c_{n_{m-1}+1}, c_{n_{m-1}+2}, \dots, c_n$  where  $W_1, W_2, \dots, W_m$  are words in Romanian. If a word  $W_j$  is not in Romanian, the algorithm will divide it into a string of characters  $W_j = c_i$ , or a combination of characters found in the resources of the algorithm (for example, it splits "#feather\_perfection" into "feather" + " " + "p" + "e" + "r" + "f" + "e" + "c" + "t" + "i" + "o" + "n"). The main function returns a set of words, having the following format:  $\{(W_1, c_1, c_{n_1}); (W_2, c_{n_1+1}, c_{n_2}); \dots (W_m, c_{n_{m-1}+1}, c_n)\}$ .

The result is useful for other natural language processing tasks that make use of word information present inside hashtags, such as named entity recognition in micro-blogging texts. While categorizing corpora of text into entities and sub-entities, it is crucial not to neglect entities found within hashtags, as they are as important as the ones found before them. For instance, splitting "#orasulbucuresti" (Bucharest city) helps to identify "bucuresti" as a spatial entity. In addition, entities that contain hashtags within them, such as "Consiliul Local #Brasov" ("#Brasov Local Council") may benefit from this approach. By using the proposed algorithm, the sentence changes into "Consiliul Local Brasov", which is an organizational entity.

The Viterbi algorithm (Viterbi 1967) is used as a basis for our approach. It is a dynamic programming algorithm that works by branching out on all possible paths of a sequence and then analyzing them to find the most probable one. By using lists, the Viterbi algorithm implementation stores the probability of each path it generates. The algorithm calculates the probability that a sequence  $W_k$  is found in the corpus. The probability that  $W_k$  is in the  $S = W_1, \dots, W_k, \dots, W_m$  path is calculated by multiplying  $W_k$ 's probability with each previous word's probability ( $P(W_j \in S)$ , where  $j$  takes values from 1 to  $k-1$ ) individually. This way, the algorithm determines the probability of a sub-sequence being a valid word. Therefore, once the algorithm is reaching its end, it only has to analyze the paths created after the second to last step.

Given that no previous attempts were reported for segmenting hashtags in the Romanian language, we aimed to establish a baseline, relying on the Viterbi algorithm instead of other more complex methods. Furthermore, the work of (Sheng et al., 2021) shows that initial Viterbi-based segmentation can be further refined with additional functions. Certain aspects of the algorithm can be customized at runtime by using parameters. Most of them focus on processing the text input and frequency files. For example, they can indicate whether the user wants to modify the text sample, like transforming all of the sample's letters into their lowercase equivalent or cleaning up the hashtag by removing the diacritics. Their values are checked at the beginning of the function and the changes are made before the Viterbi algorithm is used. In this regard, the first step of

working towards splitting a hashtag is normalizing the text input, which is realized by removing any Romanian diacritics or uppercase letters, if desired. When it comes to diacritics, not every user utilizes them, and sequences of text are not guaranteed to conform to the correct spelling or casing format (i.e., some users might write messages in all caps to convey important/urgent ideas).

The algorithm utilizes known words from a supplied resource file. In this case, it reads each word's probability using a tab-separated (TSV) file of Romanian frequency words extracted from the CoRoLa corpus (see Section 3.2). Therefore, adapting the algorithm to work with code-mixed situations (messages written using a combination of languages, including Romanian) requires extending the frequency list with additional foreign-language words.

The algorithm is given in Listing 1 and was implemented using Python with its result encoded internally using dictionaries. The final output is in the JSON format with the keys "token", "start" and "end", as seen in the following example: "#EENCanHelp" turns into [{"token": "EEN", "start": 1, "end": 3}, {"token": "Can", "start": 4, "end": 6}, {"token": "Help", "start": 7, "end": 10}]. The algorithm admits arguments taken from the command line as parameters that pertain to the processing of the text or frequency file. Finally, even if the string might not contain a hashtag symbol, the indexing of the positions is flexible, as the argument "hashtag" keeps track of its presence. Let's take the string "#SigurantaOnline" (#OnlineSafety), for example. In the dictionary returned by the function, the word "Siguranta" is between the first and ninth positions, therefore the start and end positions will be 1 and 9. However, if the hashtag is removed, the start/end positions become 0 and 8.

The algorithm's complexity is  $O(n * m)$ , where  $n$  is the size of the inputted hashtag and  $m$  is the maximum size of a word found in the corpus, which gets automatically calculated using the frequency file. Methods of parallelization have been proposed for different parts of the Viterbi algorithm, but our approach is serial, without additional optimizations. Considering the fact that the maximum hashtag size in the MicroBloggingNERo corpus is 38 and considering a maximum word of the same length, given the complexity  $O(n * m)$ , the number of cycles is limited to a maximum of 1,444.

```

Input: Hashtag  $H = c_1, c_2, \dots, c_n$ , normalization parameters
Output:  $\{(W_1, c_1, c_{n_1}); (W_2, c_{n_1+1}, c_{n_2}); \dots (W_m, c_{n_{m-1}+1}, c_n)\}$ 
1. Normalize text according to normalization parameters (lowercase, diacritics)
2. For each sequence  $W_k = c_i, \dots, c_j$ , where  $1 \leq i, j \leq n$ 
2.1. Compute probability  $P(W_k) = \frac{N(W_k)}{\text{total}}$ , where  $N(W_k)$  is the number of  $W_k$  occurrences in the corpus and  $\text{total}$  is the number of words in the corpus. If  $W_k$  has no occurrences,  $N(W_k)$  is 0, therefore the probability  $P(W_k)$  is also set to 0. ( $N(W_k)$  is loaded from the external resource.)
3. Compute the optimal path, which contains the word sequences with the highest probabilities  $P(W_k)$ 
4. For each path  $S = W_1, \dots, W_k, \dots, W_m$ 
4.1. Compute the probability  $P(W_k \in S) = (\text{for } j = 1..k-1, P(W_j \in S) * P(W_k))$ 
    
```

Listing 1. Hashtag splitting algorithm

### 3.2 CoRoLa-based Resources

The Reference Corpus for Contemporary Romanian Language (CoRoLa) (Tufiş et al., 2019; Cristea et al., 2019) was constructed as a priority project of the Romanian Academy, between 2014 and 2017, with smaller additions and corrections in recent years. It contains both written texts and oral recordings. Its aim is to cover major functional language styles (legal, scientific, journalistic, imaginative, memoirs, administrative), in four domains (arts and culture, nature, so-

ciety, science) and in 71 sub-domains while taking into account intellectual property rights (IPR). With over 1 billion word tokens (written and spoken), CoRoLa<sup>2</sup> is one of the largest fully IPR-cleared Reference Corpus in the world. CoRoLa is searchable via three interfaces (two for the written part and one for the oral part) and it is supported by the KorAP corpus management platform, developed at the Institute for the German Language in Mannheim (Bański et al., 2014; Diewald et al., 2016).

We employed the written part of the CoRoLa corpus to extract frequency lists for use in the hashtag-splitting algorithm. We constructed 24 frequency lists: 12 based on the word forms and 12 using the lemmas. For each one, we considered the minimum number of apparitions in the corpus (all, at least 5 times, at least 10 times) and different forms of writing (as appears in the corpus, lowercase, with diacritics replaced with ASCII equivalents). The generated frequency lists with the associated number of entries are available in Table 1. The generated files (using a tab-separated format) are publicly available for download from the Zenodo platform<sup>3</sup>.

*Table 1. CoRoLa frequency lists with the associated number of entries*

| Type  | Lowercase | Diacritics | Min Freq | # Entries |
|-------|-----------|------------|----------|-----------|
| Lemma | Yes       | Replaced   | 1        | 1,375,725 |
| Lemma | Yes       | Yes        | 1        | 1,457,518 |
| Lemma | No        | Replaced   | 1        | 1,562,523 |
| Lemma | No        | Yes        | 1        | 1,635,250 |
| Lemma | Yes       | Replaced   | 10       | 227,590   |
| Lemma | Yes       | Yes        | 10       | 235,234   |
| Lemma | No        | Replaced   | 10       | 242,325   |
| Lemma | No        | Yes        | 10       | 248,593   |
| Lemma | Yes       | Replaced   | 5        | 351,596   |
| Lemma | Yes       | Yes        | 5        | 365,463   |
| Lemma | No        | Replaced   | 5        | 380,751   |
| Lemma | No        | Yes        | 5        | 392,053   |
| Word  | Yes       | Replaced   | 1        | 1,685,410 |
| Word  | Yes       | Yes        | 1        | 1,813,746 |
| Word  | No        | Replaced   | 1        | 2,112,107 |
| Word  | No        | Yes        | 1        | 2,260,992 |
| Word  | Yes       | Replaced   | 10       | 358,577   |
| Word  | Yes       | Yes        | 10       | 381,715   |
| Word  | No        | Replaced   | 10       | 447,538   |
| Word  | No        | Yes        | 10       | 473,087   |
| Word  | Yes       | Replaced   | 5        | 517,630   |
| Word  | Yes       | Yes        | 5        | 553,031   |
| Word  | No        | Replaced   | 5        | 650,971   |

<sup>2</sup> <https://corola.racai.ro>

<sup>3</sup> <https://doi.org/10.5281/zenodo.7091534>

| Type | Lowercase | Diacritics | Min Freq | # Entries |
|------|-----------|------------|----------|-----------|
| Word | No        | Yes        | 5        | 690,676   |

Table 2. Calculated accuracy based on CoRoLa frequency lists

| Resource                                                        | Correctly Predicted Hashtags | Accuracy |
|-----------------------------------------------------------------|------------------------------|----------|
| word freq corola ro.tsv<br>(all words in corpus)                | 943                          | 77.74 %  |
| word freq corola ro<br>gte5.tsv<br>(frequency >= 5 in corpus)   | 920                          | 75.84 %  |
| word freq corola ro<br>gte10.tsv<br>(frequency >= 10 in corpus) | 899                          | 74.11 %  |

Table 3. Sample errors

| Hashtag            | Correct Output      | Actual Result                |
|--------------------|---------------------|------------------------------|
| #Hamid-Reza        | Hamid-Reza          | Hamid - R e z a              |
| #Inglés            | Inglés              | In gl é s                    |
| #1milion           | 1 milion            | 1 m i l i o n                |
| #Worlfsburg        | Worlfsburg          | Worl fs burg                 |
| #DescubreAlAsesino | Descubre Al Asesino | Des cub reAlA se si no       |
| #ne_vedem_la_vot   | ne vedem la vot     | ne _ v e d e m _ l a _ v o t |
| #RETROSPECTIVĂ2021 | RETROSPECTIVĂ 2021  | RETROSPECTIVĂ 2 0 2 1        |

## 4 Method

To evaluate the proposed algorithm, as described in Section 3, we extracted the available hashtags from the MicroBloggingNERo corpus and manually created a list with the unique hashtags and their corresponding splits.

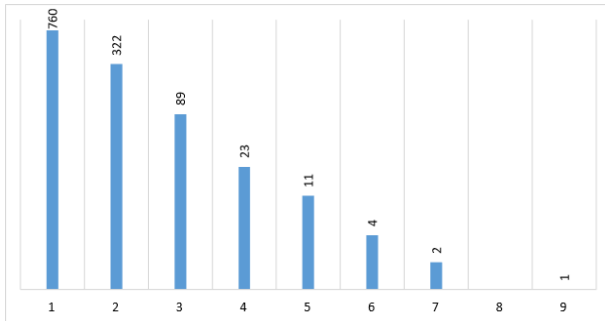


Figure 1. The distribution of the number of words per hashtag

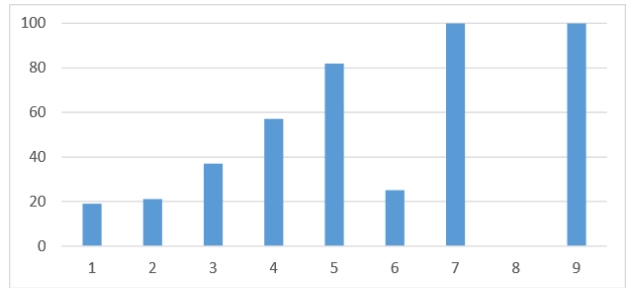


Figure 2. The percentage of errors in splitting hashtags into a certain number of words

A statistic is given in Fig. 1 considering the distribution of the number of words per hashtag. It can be easily observed that while the number of words per hashtag increases, the frequency of the respective hashtags decreases. As seen in Fig. 1, there can also be hashtags composed of many words, for example, #TheBestForFoodServiceForABetterLife is split the following way: “The” + “Best” + “For” + “Food” + “Service” + “For” + “A” + “Better” + “Life”, being the only one from

the dataset formed from 9 words (the dataset also contains hashtags in English because of the code-mixed language commonly used in social media texts). Hashtags such as #ofertă\_specială (#special\_offer) are considered one-word because of the ‘\_’ sign which marks a single structure, however "#ofertăspecială" would be split into two words.

Figure 2 presents the percentages of errors during the automated splitting of hashtags with a certain number of composing words. It can be inferred on a general level that the errors’ values rise in accordance with the number of composing words that make up the hashtags.

We executed the algorithm over the entire list, using the different resources presented in Section 3.2 with characteristics given in Table 1. We employed only the word-level resources (while the lemma-based resources were created for possible future research). For evaluation purposes, we computed the accuracy (see Equation 1) of the automatically produced splits. This expresses the fraction of the correctly predicted splits from the total number of unique hashtags present in the corpus. Results are given in Table 2.

$$Accuracy = \frac{CorrectSplits}{TotalUniqueHashtags} \quad (1)$$

As indicated by the accuracy values, the best results were achieved using the resource word\_freq\_corola\_ro.tsv. Intuitively, this happened because the file presents the most words out of all three. Thus, the algorithm is able to split hashtags that also contain unusual and/or uncommon words.

Following the execution of the algorithm, we looked into the special cases where the algorithm failed. Examples of such cases are given in Table 3. An analysis of the errors indicates that the algorithm failed because the frequency file exclusively contains words in Romanian. As such, when the hashtags contain words from languages other than Romanian, numerals (such as years), or any other types of special characters, the algorithm does not find their counterparts in the frequency file. For the failed cases with Romanian words, the main source of errors is represented by the presence of numbers or incorrectly spelled words. For example, "#RETROSPECTIVĂ2021" (#2021RETROSPECTIVE) failed because it contains the year 2021, which is not present in the frequency file. Another source of errors is the presence of underscores in certain hashtags. However, we did not take into account any explicitly introduced splitting characters, because we analyzed the text considering the non-uniform behavior of users.

We further explored the impact of including the algorithm’s result while annotating text using different basic language annotation resource kits (BLARK). For this purpose, we randomly extracted 20 sentences from the MicroBloggingNERo corpus containing hashtags, that were automatically annotated with different BLARK pipelines for the Romanian language, as evaluated by (Păiș et al., 2021). None of the existing pipelines (UDPipe (Straka et al., 2016), NLP Cube (Boroș et al., 2018), TTL (Ion, 2007), TreeTagger (Schmid, 1994), RNNTagger (Schmid, 2019), Stanza (Qi et al., 2020)), given Romanian language models, are aware of hashtags. Instead, the entire hashtag is considered a single token. This makes it impossible to properly annotate the content of the hashtag. In Table 4, an example can be found of processing one sentence ("Am avut un #crăciundepoveste în #Bucureștiulmeu.") using the TEPROLIN web service (Ion, 2018). Internally, it makes use of NLP Cube for part-of-speech tagging and dependency parsing and calls an external NER system (Păiș, 2019) for named entity recognition. In addition, we executed a term annotation process (Coman et al., 2019; Tufiș et al., 2020), using the IATE terminology. All these processes were executed inside the RELATE platform (Păiș et al., 2019; Păiș et al., 2020). In Table 5, we used the same sentence, but with the hashtags split ("Am avut un crăciun de poveste în Bucureștiul meu."). Here, in the initial text, "crăciun" appeared fully lowercase. In the second case, the NER and IATE annotation processes were able to identify additional information in the provided text.

## 5 Conclusion

Hashtags are commonly used in social media texts. Thus, it is important to be able to process them properly for different natural language processing tasks. Sometimes hashtags may contain named entities, terms, or other information which may be missed unless the internal structure of the hashtag is investigated. In this paper, we introduced an algorithm for splitting Romanian hashtags into constituent words. Furthermore, we proved that CoRoLa-based resources play a key role in improving the algorithm’s capabilities. Finally, we consider that basic micro-blogging text operations can be improved by the addition of this algorithm.

We make the algorithm public<sup>4</sup> with all the associated resources. Furthermore, the algorithm is also available on the RELATE platform<sup>5</sup>. Currently, the algorithm is not included in any Romanian language processing pipeline. Future work will explore the possibility of integrating the algorithm into a BLARK pipeline and additional splitting based on explicitly defined delimitation characters (such as underscores). Additionally, other textual cues (such as lowercase-uppercase transitions) may be explored to identify other means of splitting the hashtags. Furthermore, the separation into individual letters for unknown tokens may not be the best solution and a concatenated string may instead provide better results. Nevertheless, there are specific cases (such as people’s initials) where providing individual letters is the only viable result. Therefore, additional resources such as lists of names with initials may be explored.

Table 4. Example annotation using TEPROLIN without hashtag splitting

| ID | FORM             | LEMMA            | UPOS  | XPOS   | NE    | NP   | IATE |
|----|------------------|------------------|-------|--------|-------|------|------|
| 1  | Am               | avea             | AUX   | Va-1   | O     | O    | —    |
| 2  | avut             | avea             | VERB  | Vmp-sm | O     | O    | —    |
| 3  | un               | un               | DET   | Timsr  | O     | B-NP | —    |
| 4  | crăciundepoveste | crăciundepoveste | NOUN  | Ncfsrn | O     | I-NP | —    |
| 5  | în               | în               | ADP   | Spsa   | O     | O    | —    |
| 6  | Bucureştiulmeu   | Bucureştiulmeu   | PROPN | Np     | B-LOC | B-NP | —    |
| 7  | .                | .                | PUNCT | PERIOD | O     | O    | —    |

Table 5. Example annotation with hashtags splitting

| ID | FORM        | LEMMA     | UPOS  | XPOS    | NE     | NP   | IATE      |
|----|-------------|-----------|-------|---------|--------|------|-----------|
| 1  | Am          | avea      | AUX   | Va-1    | O      | O    | —         |
| 2  | avut        | avea      | VERB  | Vmp-sm  | O      | O    | —         |
| 3  | un          | un        | DET   | Timsr   | O      | B-NP | —         |
| 4  | crăciun     | crăciun   | NOUN  | Ncms-n  | B-TIME | I-NP | —         |
| 5  | de          | de        | ADP   | Spsa    | O      | O    | —         |
| 6  | poveste     | poveste   | NOUN  | Ncfsrn  | O      | B-NP | —         |
| 7  | în          | în        | ADP   | Spsa    | O      | O    | —         |
| 8  | Bucureştiul | Bucureşti | PROPN | Npmsry  | B-LOC  | B-NP | 1:1891531 |
| 9  | meu         | meu       | PRON  | Ps1ms-s | O      | I-NP | —         |

<sup>4</sup> <https://github.com/racai-ai/MicroBloggingRo>

<sup>5</sup> [https://relate.racai.ro/index.php?path=social\\_media/hashtags](https://relate.racai.ro/index.php?path=social_media/hashtags)

| ID | FORM | LEMMA | UPOS  | XPOS   | NE | NP | IATE |
|----|------|-------|-------|--------|----|----|------|
| 10 | .    | .     | PUNCT | PERIOD | O  | O  | —    |

### References

- Bański, Piotr, Nils Diewald, Michael Hanl, Marc Kupietz, and Andreas Witt. (2014). Access Control by Query Rewriting: The Case of KorAP. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, 3817–22. Reykjavik, Iceland: European Language Resources Association (ELRA). [http://www.lrec-conf.org/proceedings/lrec2014/pdf/743\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2014/pdf/743_Paper.pdf).
- Boroş, Tiberiu, Ştefan Daniel Dumitrescu, and Ruxandra Burtică. (2018). NLP-Cube: End-to-End Raw Text Processing with Neural Networks. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, 171–79. Brussels, Belgium: Association for Computational Linguistics. <https://doi.org/10.18653/v1/K18-2017>.
- Çelebi, Arda, and Arzucan Özgür. (2016). Segmenting Hashtags Using Automatically Created Training Data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 2981–85. Portorož, Slovenia: European Language Resources Association (ELRA). <https://aclanthology.org/L16-1476>.
- Coman, Andrei, Maria Mitrofan, and Dan Tufiş. (2019). Automatic Identification and Classification of Legal Terms in Romanian Law Texts. In *International Conference on Linguistic Resources and Tools for Natural Language Processing*.
- Cristea, Dan, Nils Diewald, Gabriela Haja, Cătălina Mărânduc, Verginica Barbu Mititelu, and Mihaela Onofrei. (2019). How to Find a Shining Needle in the Haystack. Querying CoRoLa: Solutions and Perspectives. *RRL*, no. 3: 279–92.
- Diewald, Nils, Michael Hanl, Eliza Margaretha, Joachim Bingel, Marc Kupietz, Piotr Bański, and Andreas Witt. (2016). KorAP Architecture — Diving in the Deep Sea of Corpus Data. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 3586–91. Portorož, Slovenia: European Language Resources Association (ELRA). <https://aclanthology.org/L16-1569>.
- Halgamuge, Malka N., Huseyin Caliskan, and Azeem Mohammad. (2021). Computation Time Optimization on Hashtag Segmentation for Social Media Data. In *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 1–6. <https://doi.org/10.1109/WCNC49053.2021.9417569>.
- Henrich, Verena, and Erhard Hinrichs. (2010). GernEdiT - the GermaNet Editing Tool. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA). [http://www.lrec-conf.org/proceedings/lrec2010/pdf/264\\_Paper.pdf](http://www.lrec-conf.org/proceedings/lrec2010/pdf/264_Paper.pdf).
- Ion, Radu. (2007). *Word Sense Disambiguation Methods Applied to English and Romanian*. PhD thesis, Romanian Academy.
- Ion, Radu. (2018). TEPROLIN: An Extensible, Online Text Preprocessing Platform for Romanian. In *The 13th International Conference on Linguistic Resources and Tools for Natural Language Processing*, 69–76. <https://profs.info.uaic.ro/~consilr/2019/wp-content/uploads/2019/06/volum-ConsILR-2018-1.pdf#page=77>.
- Kodali, Prashant, Akshala Bhatnagar, Naman Ahuja, Manish Shrivastava, and Ponnurangam Kumaraguru. (2022). *HashSet - A Dataset for Hashtag Segmentation*. CoRR abs/2201.06741. <https://arxiv.org/abs/2201.06741>.
- Leskovec, Jure, and Andrej Krevl. (2014). *SNAP Datasets: Stanford Large Network Dataset Collection*. <http://snap.stanford.edu/data>.
- Maddela, Mounica, Wei Xu, and Daniel Preoţiuc-Pietro. (2019). Multi-Task Pairwise Neural Ranking for Hashtag Segmentation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2538–49. Florence, Italy: Association for Computational Linguistics. <https://doi.org/10.18653/v1/P19-1242>.
- Manolescu, Mihai, and Çağrı Çöltekin. (2021). ROFF - a Romanian Twitter Dataset for Offensive Language. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, 895–900. Held Online: INCOMA Ltd. <https://aclanthology.org/2021.ranlp-1.102>.
- Păiş, Vasile, Radu Ion, Andrei-Marius Avram, Maria Mitrofan, and Dan Tufiş. (2021). In-Depth Evaluation of Romanian Natural Language Processing Pipelines. *Romanian Journal of Information Science and Technology (ROMJIST)*, 24 (4): 384–401. <https://www.romjist.ro/full-texts/paper700.pdf>.
- Păiş, Vasile. (2019). *Contributions to Semantic Processing of Texts; Identification of Entities and Relations Between Textual Units; Case Study on Romanian Language*. PhD thesis, Romanian Academy.

- Păiș, Vasile, Radu Ion, and Dan Tufiș. (2020). A Processing Platform Relating Data and Tools for Romanian Language. In *Proceedings of the 1st International Workshop on Language Technology Platforms*, 81–88. Marseille, France: European Language Resources Association. <https://www.aclweb.org/anthology/2020.iwlt-1.13>.
- Păiș, Vasile, Maria Mitrofan, Verginica Barbu Mititelu, Elena Irimia, Roxana Micu, and Carol Luca Gasan. (2022). Challenges in Creating a Representative Corpus of Romanian Micro-Blogging Text. In *Proceedings of the Workshop on Challenges in the Management of Large Corpora (CMLC-10)*, 1–7. Marseille, France: European Language Resources Association. <https://aclanthology.org/2022.cmlc10-1.1>.
- Păiș, Vasile, Maria Mitrofan, Verginica Barbu-Mititelu, Elena Irimia, Carol Luca Gasan, Roxana Micu, Laura Marin, Maria Dicusar, Bianca Florea, and Ana Badila. (2022). Romanian micro-blogging named entity recognition (MicroBloggingNERo). *Zenodo*. <https://doi.org/10.5281/zenodo.6905235>.
- Păiș, Vasile, Dan Tufiș, and Radu Ion. (2019). Integration of Romanian NLP Tools into the RELATE Platform. In *International Conference on Linguistic Resources and Tools for Natural Language Processing*, 181–92. [https://profs.info.uaic.ro/~consilr/2019/wp-content/uploads/2020/01/ConsILR2019\\_final\\_BTT-60-ex-B5.pdf#page=189](https://profs.info.uaic.ro/~consilr/2019/wp-content/uploads/2020/01/ConsILR2019_final_BTT-60-ex-B5.pdf#page=189).
- Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. (2020). Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, 101–8. Online: Association for Computational Linguistics. <https://doi.org/10.18653/v1/2020.acl-demos.14>.
- Reuter, Jack, Jhonata Pereira-Martins, and Jugal Kalita. (2016). Segmenting Twitter Hashtags. *International Journal on Natural Language Computing*, 5 (August): 23–36. <https://doi.org/10.5121/ijnlc.2016.5402>.
- Rodrigues, Ruan Chaves, Marcelo Akira Inuzuka, Juliana Resplande Sant’Anna Gomes, Aquila Santos Rocha, Iacer Calixto, and Hugo Alexandre Dantas do Nascimento. (2021). *Zero-Shot Hashtag Segmentation for Multilingual Sentiment Analysis*. ArXiv abs/2112.03213.
- Schmid, Helmut. (1994). Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of the International Conference on New Methods in Language Processing*. Manchester, UK.
- Schmid, Helmut. (2019). Deep Learning-Based Morphological Taggers and Lemmatizers for Annotating Historical Texts. In *Proceedings of the 3rd International Conference on Digital Access to Textual Cultural Heritage*, 133–37. DATeCH2019. New York, NY, USA: Association for Computing Machinery. <https://doi.org/10.1145/3322905.3322915>.
- Sheng, Longshuai, Ce Li, and Yihan Tian. (2021). Coarse-to-Fine Loss Based on Viterbi Algorithm for Weakly Supervised Action Segmentation. In *2021 International Conference on Signal Processing and Machine Learning (CONF-SPML)*, 1–6. <https://doi.org/10.1109/CONF-SPML54095.2021.00009>.
- Straka, Milan, Jan Hajič, and Jana Straková. (2016). UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, 4290–97. Portorož, Slovenia: European Language Resources Association (ELRA). <https://aclanthology.org/L16-1680>.
- Tufiș, Dan, Verginica Barbu Mititelu, Elena Irimia, Vasile Păiș, Radu Ion, Nils Diewald, Maria Mitrofan, and Onofrei Mihaela. (2019). Little Strokes Fell Great Oaks. Creating CoRoLa, the Reference Corpus of Contemporary Romanian. *Revue Roumaine de Linguistique*, 64 (3): 227–40.
- Tufiș, Dan, Maria Mitrofan, Vasile Păiș, Radu Ion, and Andrei Coman. (2020). Collection and Annotation of the Romanian Legal Corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, 2773–77. Marseille, France: European Language Resources Association. <https://aclanthology.org/2020.lrec-1.337>.
- Viterbi, A. (1967). Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. *IEEE Transactions on Information Theory*, 13 (2): 260–69. <https://doi.org/10.1109/TIT.1967.1054010>.
- Weller-Di Marco, Marion. (2017). Simple Compound Splitting for German. In *Proceedings of the 13th Workshop on Multiword Expressions (MWE 2017)*, 161–66. Valencia, Spain: Association for Computational Linguistics. <https://doi.org/10.18653/v1/W17-1722>.

# PERSONALITY PREDICTION BASED ON SOCIAL MEDIA POSTS

BIANCA IONELA STRĂTIANU<sup>1</sup> AND ADRIAN IFTENE<sup>1</sup>

<sup>1</sup>*Faculty of Computer Science, “Alexandru Ioan Cuza”, General Berthelot Street,  
16, Iasi, Romania*

## Abstract

Personality is an essential component in our lives, influencing the way we behave, speak, and respond, but also the activities we do. Usually, people turn to a psychologist or a personality test to get this information. At the moment we display most of our life on social networks, so the posts that the user makes can be analyzed in order to obtain personal information about them. This paper presents a way to extract text from social media posts for personality recognition.

## 1 Introduction

Personality is “the combination of characteristics or qualities that form an individual’s distinctive character”<sup>1</sup> (Corr and Matthews, 2009). There are several methods to investigate human personality, among which we mention observation, experiment, conversation, psychological investigation, physiognomy study, anamnesis, sociometric test, graphology, case study, etc. We live in a world where we expose our personal lives and thoughts on social media. The main questions we are trying to answer in this paper are: “*Do we express our personality in the posts we have on our social media accounts? Is it possible to identify a person’s personality based on his posts on social networks?*”.

Personality is closely related to the preferences and behavior of the user, so based on it we can improve posts, advertisements, and preferred types of music, but also recommendations from other users who have the same type of personality or with whom they could get along.

### 1.1 The Personality Types

We have different perspectives and theories for personality types (Matthews and Gillian, 1999). Donnellan and Robins (2010) use resilient, overcontrolled, and undercontrolled personality types, while Eysenck (1994) uses biological foundations. We chose to use the Myers-Briggs classification (Myers and Myers, 1980). Briggs and Myers created the MBTI<sup>2</sup> (Myers-Briggs Type Indicator) during World War II to be able to help women find a suitable job. They were inspired by the book (Jung, 1971). It assigns four categories: (1) *extraverted* (E), *introverted* (I); (2) *sensing* (S), *intuition* (N); (3) *thinking* (T), *feeling* (F); and (4) *judging* (J), *perceiving* (P). Thus, one letter is taken from each category resulting in the 16 personality types: ESTJ, ENTJ, ESFJ, ENFJ, ISTJ, ISFJ, INTJ, INFJ, ESTP, ESFP, ENTP, ENFP, ISTP, ISFP, INTP, and INFP.

The four dimensions talk about preferences, each with two branches, thus extraversion/introversion answers the question “*Where do you get your energy from?*”, sensation/intuition “*What type of information do you prefer to use?*”, thinking/feeling “*What method do you use to make decisions?*”, judgment/perception “*How do you see the world around you?*”.

*Extraversion* applies to a person who acts before thinking, they get their energy from the outside, from the people they communicate with, and they have a wide range of interests. *Introversion* is the opposite of extroversion, meaning people tend to think before they act, prefer to spend

---

<sup>1</sup> [https://www.oxfordlearnersdictionaries.com/definition/american\\_english/personality](https://www.oxfordlearnersdictionaries.com/definition/american_english/personality)

<sup>2</sup> <https://www.verywellmind.com/the-myers-briggs-type-indicator-2795583>

time reflecting and get their energy from within (meditation/concentration). The feeling applies to those who prefer facts, look at details, focus on the here and now, and collect information about the world. Intuition applies to those looking for new ideas, looks at the big picture, and makes sure things work in theory. People who like to apply logical reasoning, use cause-effect analysis, decide using impersonal criteria, and focus on tasks tend to be more inclined towards the thinking function. Intuition makes people understand other people's points of view, focus on details, offer praise, and also make decisions based on personal values. People who like to make plans, act in a controlled manner, prefer to schedule activities are more inclined towards the judgmental type, and people who are more flexible, prefer to be spontaneous, and want to have more options are those who lean towards the perception.

### 1.2 Previous Approaches

Although the field is relatively new, various research has been done on this topic, the vast majority of them targeting only a certain data set or only certain algorithms. In the paper (Cui and Qi, 2017), various personality prediction algorithms are presented, such as Naive Bayes, SVM (Support Vector Machines) (Gandhi, 2018), and Deep Learning (Burns and Brush, 2021). The results for different test data are ranging from 17% to 38%. The input dataset was taken from Kaggle<sup>3</sup> and processed: conversion to lowercase, punctuation removal, word lemmatization, identifying special texts (URL, numbers, emoji), and, finally, assigning to each lemma its frequency.

The article (Keh and Cheng, 2019) presents algorithms such as Logistic Regression, LSTM (Long Short-Term Memory) on different data sets taken from Twitter, and Reddit, but also a forum<sup>4</sup> created specifically for personality discussions where each user has their type profiled. For the extracted data, punctuation marks were deleted and converted to lowercase letters and the personality type explicitly written was replaced with a specific word. The models created in this paper have an accuracy of 19% to 47%.

In the paper (Salsabila and Setiawan, 2021), the authors take data from Twitter and use two different methods: the first is based on SVM, and the second is a combination of SVM with BERT as the semantic approach. The results obtained show an accuracy of around 80%.

### 1.3 Contributions

This paper uses text classification from social media posts (Instagram, Twitter) to predict user personality. The language used for the extracted texts is English, and the classification is performed by implementing Naive Bayes, k-NN (k - Nearest Neighbors), and XGBoost algorithms (Cuşmuliuc et al., 2018; Cuşmuliuc et al., 2021). The datasets we trained the algorithms on are taken from Kaggle. We analyzed and removed impurities from the data set (null words, connecting words, but also punctuation marks).

The contributions are that we combined the two data sets (from Kaggle and Reddit - see section 2.3 for more details) to have better accuracy of the algorithm, reaching an increase of about 4% for the test data compared to the original results. We trained several algorithms, which had different results, more or less well. Thus, from the first algorithm which had an accuracy of approximately 20%, we reached a model that has an average accuracy of 78.25%. To add practical utility to our research work, we built a system where a user can enter his or other people's name on social media accounts to find out their personalities. In order to help user to understand the results, we have inserted in the web page posts of people with the same type of personality, but also the 100 most common words from all the input data on which we trained the algorithms.

The idea of this project is to present to the user an image of how he can be perceived on social media based on the posts he makes. The user can also enter a certain text to see how it will be

<sup>3</sup> <https://www.kaggle.com/datasets/datasnaek/mbti-type>

<sup>4</sup> <https://www.personalitycafe.com/>

perceived as a future post. He can also see the personality of friends, acquaintances, or different people to see if they can get along or have certain points in common because personality influences character and behavior.

## 2 Proposed Solution

The purpose of the developed app is to identify the personality type of a user based on the texts on social networks or only on the basis of a certain selected text. Thus, as input, we will have a text or the name of the user from social networks, and as output, we will have the personality type and details about it.

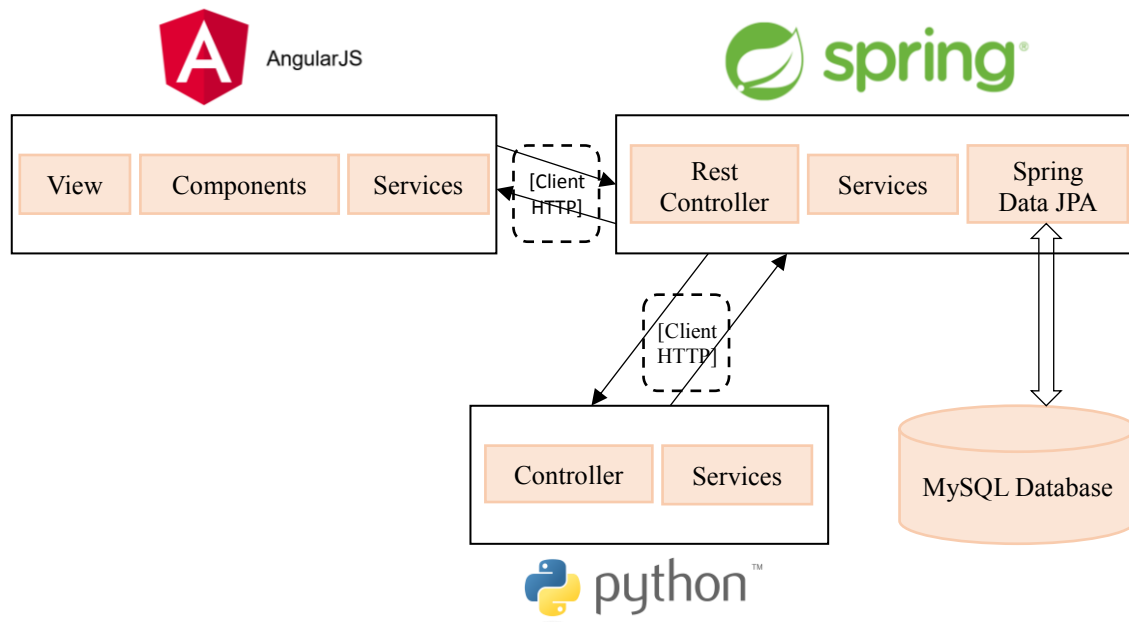


Figure 1. Application Architecture

Figure 1 shows the application architecture and how the application works. The client sends HTTP requests to the server, and it communicates with the MySQL database and the server written in Java. The server communicates with the component connected to the database, with another component that calls the ML (Machine Learning) model written in Python and with the user interface through Angular. For this, using the user's name on social networks extracts the posts or text sent by the user so that it can be analyzed. The result is returned to the back end so that all information about it can be searched in the database. A more detailed structure of how the application interacts with the user is described below.

### 2.1 The Back-end

The back-end module is divided into two parts: (1) the classification part made with the help of a machine learning algorithm and (2) the part that communicates with the front-end module. We made a REST<sup>5</sup> model which we divided into 3 packages: (1) "controller" (which displays the endpoints to the outside), (2) "models" (take the state of the model and save it to the database) and (3) "service" (with which we can change the state of the models, allowing to delete, update, insert) (Baboi et al., 2019). We also have a "repository" package to facilitate interaction with the database, having some predefined methods. In order to have a more organized structure, we chose to create two tables in the MYSQL database:

<sup>5</sup> <https://www.codecademy.com/article/what-is-rest>

1. The first one is for famous people with the same personality, where we store their name, a picture, and their type.
2. The second one is where we have similar posts, a description of the personality type, but also the picture with the 100 most common words.

For the algorithm part, we chose to save the built model to reduce the execution time, and for extracting the posts we chose to analyze them from the most recent to the oldest within a hundred posts if they exist. To communicate with the Java back-end we built a controller that takes the user's name from social networks to be able to extract his posts and returns the personality type.

## 2.2 The Front-end

We have three more important pages: (1) Home where the user finds details about personality in general; (2) Input page where the data is entered; and (3) Result page where we can see the results. After the user enters the name of the social media accounts they want to analyze, the result will come in the form of the obtained personality type, celebrities who have this type of personality<sup>6</sup>, and similar posts. The description for each type is based on information from Myers-Briggs Type Indicator<sup>7</sup> (Cherry, 2022). For the well-known personalities who have the same personality, we chose four people for each type and initially entered into the database both their names and a picture so that they could be recognized as easily as possible.

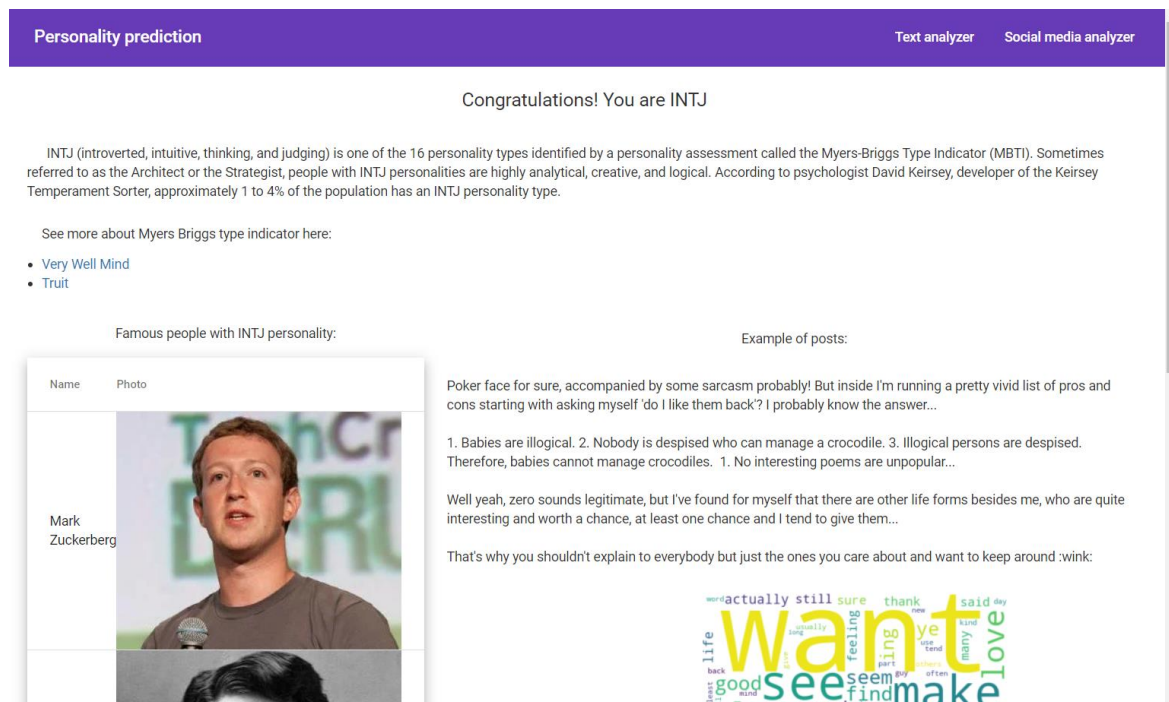


Figure 2. Front-end - The Result Page

In Fig. 2, we can see the page where the user can see the results obtained. In order for him to have as much confidence as possible in the returned result, we chose to show him posts from the data set on which the algorithm was trained that were characterized by the same personality. For all the posts in the datasets of a certain category, we managed to extract the 100 most used words so that the user can understand why his text was characterized in that way. We also chose to create a separate page for entering only a certain text so that the user can figure out exactly which

<sup>6</sup> <https://www.thefamouspeople.com/personality-type.php>

<sup>7</sup> <https://www.verywellmind.com/the-myers-briggs-type-indicator-2795583>

text or words determined a certain type of personality or he could enter the text of the intended following post to see how it will probably be perceived by his followers.

### 2.3 Datasets

The first dataset used was taken from Kaggle<sup>8</sup>. It was built in 2007 as part of an experiment on Facebook, in which users filled out a questionnaire aimed at finding out their personalities. They agreed that the app could also save the posts they had on their personal profile. Then a connection was discovered between the text of the posts and the personality obtained from the answers to the questions. It contains 8,675 lines, meaning all that many users, each having two columns, the first representing the personality type and the second column being the person's posts. On each line, we consider the last 50 posts of a user.

The second dataset was built in 2015 by Gjurković and Šnajder (Gjurkovic and Šnajder, 2018) with the help of Reddit posts of over 9,000 users. They contain more data such as the author of the post, his id, and things that did not help us in building the algorithm, so we reduced this set to just the personality type and the text of the user's posts and comments.

Although we had different data sets with different numbers of columns, in the final data set we combined only the data we need, which only contains the personality type and the posts. In the final data, we have certain personality types that have more entries (INTP, INFP, INTJ, INFJ) because they are the most predominant (Mallari and Pelayo, 2017). Our algorithm on the final combined data had a prediction about 4% better than the best results obtained on the separate data sets. The data were divided into training data and test data (distribution being 80% and 20%).

### 2.4 Data Processing

All the data we extracted for training the algorithm was processed with the help of various libraries. We did the following processing steps (Iftene et al., 2020): (1) we removed the URL addresses, (2) we deleted the punctuation marks, (3) we turned the uppercase letters into lowercase letters, (4) we removed all the linking words. In the end, we ended up with a data set of length 16 (for all 16 personality types) having only words that could help us in building the future algorithm. The libraries used for all processes on the input data are:

- “re”<sup>9</sup> used to remove URLs (re.sub(r'http.\*?(?=\s)', '', ...));
- “nltk.corpus”<sup>10</sup> to remove connecting words, also called stop words;
- “nltk.tokenize”<sup>11</sup> to split sentences into words;
- “string”<sup>12</sup> to remove punctuation.

After the first step of data processing, from the more than 17,000 lines from the union of the two data sets, we reached only 16 data points (corresponding to the 16 personality types), because we made a new list, with only 16 attributes, and each entry we added it to the appropriate row.

Table 1. The Processing Steps

|              |                                                                                                                                                                                                                                                                                                                                                                        |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Initial Text | https://www.youtube.com/watch?v=iz7lE1g4XM4    sportscenter not top ten plays https://www.youtube.com/watch?v=uCdfze1etec pranks     What has been the most life-changing experience in your life?    http://www.youtube.com/watch?v=vXZeYwwRDw8 http://www.youtube.com /watch?v=u8ejam5DP3E     On repeat for most of today.     May the PerC Experience immerse you. |
|--------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

<sup>8</sup> <https://www.kaggle.com/datasets/datasnaek/mbti-type>

<sup>9</sup> <https://docs.python.org/3/library/re.html>

<sup>10</sup> <https://www.nltk.org/api/nltk.corpus.html>

<sup>11</sup> <https://www.nltk.org/api/nltk.tokenize.html>

<sup>12</sup> <https://docs.python.org/3/library/string.html>

|                                       |                                                                                                                                                                                                                                                        |
|---------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Text after removing URLs              | sportscenter not top ten plays pranks    What has been the most life-changing experience in your life?    On repeat for most of today.   May the PerC Experience immerse you.                                                                          |
| Text after removing punctuation marks | ['sportscenter', 'not', 'top', 'ten', 'plays', 'pranks', 'What', 'has', 'been', 'the', 'most', 'lifechanging', 'experience', 'in', 'your', 'life', 'On', 'repeat', 'for', 'most', 'of', 'today', 'May', 'the', 'PerC', 'Experience', 'immerse', 'you'] |
| Text after removing capitalization    | ['sportscenter', 'not', 'top', 'ten', 'plays', 'pranks', 'what', 'has', 'been', 'the', 'most', 'lifechanging', 'experience', 'in', 'your', 'life', 'on', 'repeat', 'for', 'most', 'of', 'today', 'may', 'the', 'perc', 'experience', 'immerse', 'you'] |
| Text after removing stop words        | ['sportscenter', 'top', 'ten', 'plays', 'pranks', 'lifechanging', 'experience', 'life', 'repeat', 'today', 'may', 'perc', 'experience', 'immerse']                                                                                                     |

In the table above, we can see all the processing done on an initial sentence, step by step. In the end, we will have a list of words, which will later be processed by the algorithm in order to obtain a common denominator between the personality type and the lists thus obtained.

In Fig. 3, we can see the 100 most common words, according to the number of occurrences for each of the 16 personality types. It can be seen that some types have some words in common with each other, this is due to the fact that they have some things in common. For example, ENFP (d) and ESFP (h) (common words: “type”, “feel”, “love”, “thank”) only have different N/S (sensation/intuition), the other three criteria being the same. It is understood why there are some common characteristics between them.



Figure 3. The Most Frequent Words.

## 2.5 Algorithm Construction

Because we want to have the most accurate results, we did experiments with several classification algorithms, including Naive Bayes, k-NN, Logistic Regression, and XGBoost.

### Naive Bayes

Naive Bayes is an algorithm based on the Bayes theorem that calculates the probability that a given instance is in a class (Ray, 2017). MNB (Naive Multinomial Bayes) (Solanki, 2020) is an algorithm created specifically for text classification (natural language processing, NLP), it calculates for a given text, the probability of each type of personality among the 16 and returns the tag with the highest chance of appearing. The algorithm uses the weight of word occurrences in the training data to predict the test data.

We have built three algorithms for Naive Bayes. The first algorithm (with an accuracy of 38%) is built using the MNB algorithm, but without processing the input data, i.e., the training data were all punctuation marks, link words, URLs, but also words with uppercase letters. In the second algorithm (accuracy 41%), the input data was processed with the steps presented above, but we used the simple Bayes Naive algorithm, not the one created specifically for text classification. In the last algorithm we used MNB, we processed the input data, but also the test data. Due to the combination of the two algorithms, the results were about 20% better.

### k-NN

k-NN (k-Nearest Neighbors) is a classifier that uses the distance between the training data and its  $k$  nearest neighbors to determine the output (Patwardhan, 2021). This  $k$  is unknown, it must be found out by trial and error because it is not a monotonically increasing or decreasing function to know if we need to decrease  $k$  or increase it.  $k$  must be an odd number because if it is an even number, we can have 50% of neighbors in one class and 50% in another class, the algorithm cannot predict which class the instance is in.

In the first phase, we did experiments for  $k = 1, 5, 7, 15, 21, 55$  for the initial data, without processing them. The results increased from 22.54% (for  $k = 1$ ), to 29.86% (for  $k = 9$ ), which was the highest value. For  $k > 9$ , the values stagnate around this rate. Because of this stagnation in the prediction rate, we decided not to continue for  $k > 55$ , also having the Naive Bayes algorithm which had about 2 times better prediction. In the second phase, we did experiments for  $k = 1, 3, 5, 7, 9, 11, 15$  for the processed data. In this case, the best 33% results were for  $k = 1$ , and for  $k > 1$  we had a decrease, reaching up to 3.5% for  $k = 15$ . Working with the processed version of the data, we have only 16 input data points, so only 16 possible neighbors, that's why we didn't try the algorithm for  $k > 16$ , and due to this fact it is understandable why the algorithm predicts better for  $k = 1$ , being the first neighbor.

### XGBoost

XGBoost (eXtreme Gradient Boosting) is a machine-learning algorithm based on decision trees that stimulates the gradient (Morde and Setty, 2019). It is considered the best algorithm for structuring small and medium-sized data. A decision tree is a structure where each edge indicates a test and each leaf indicates a class label. Gradient boosting means building a model for the data set and then building a second algorithm to correct errors encountered in the first model. Thus we build sequential models, where the new one learns from the mistakes of the previous one because the correctly predicted results have a lower weight and the wrongly predicted results have a higher weight in building the future algorithm so that we can focus on them.

We divided the input data into 4 classes I/E, N/S, F/T, J/P because we have more training data by dividing the dataset like this and the algorithm can learn on more data with a rate of better

prediction. After processing the input data accordingly, we applied stemming and lemmatization to the resulting words. With the help of stemming algorithms, we removed the number, gender, case, and person endings. To extract lemmatized words, their morphological analysis is necessary. In this case, we have better results, reaching approximately 90% for the N/S class.

### **Logistic Regression**

Logistic regression is a process of modeling independent variables and a dependent variable (Edgar and Manz, 2017). Most logistic regressions model a binary argument (0/1, True/False), but there are also multinomial regressions that model multiple arguments, such as the 16 personality types we use. It's an algorithm that learns from past mistakes, and the more relevant input data it has, the better it can predict future data.

In the case of our algorithm, for each input having the data processed according to the described model, each word in the specified input has a certain coefficient, and after applying the logistic regression, we can say whether or not the given text belongs to a certain type of personality. After training the algorithm on the initial input data set without processing it, we got a score of about 36%, which is a lot for an initial algorithm. After changing the parameters of the logistic regression, combining the two data sets, but also processing the input data, we obtained a score of 54%, which is a very big progress compared to the initial one.

### **2.6 Problems Encountered**

#### **Initial Low Algorithm Accuracy**

One of the problems encountered was the fact that the algorithms built the first time had relatively low accuracy rates, about 38% (Naive Bayes) and 20% (k-NN) which did not help us in correctly predicting the personality type of a user. Thus, over time we tried various algorithms, with various parameters, reaching an average rate of approximately 80%, providing better precision in correctly establishing the result.

#### **Differences between Personality Types**

Another problem encountered was the large difference between posts from certain personality types (e.g., INFP and ISTJ, INFJ and ESFJ). This may be because ESTJs, ESFJs, ESFPs, and ESTPs are the least common personality types in the world. Thus, to solve this problem, we chose to train and test the algorithm created for each of the four categories. To reduce this difference and to have as much training and testing data as possible, we processed the data from another experiment done by Gjurmović, Matej, and Šnajder in 2015 with the help of Reddit posts.

#### **Credibility of Results**

After finishing the algorithm and testing it, it was noticed that the application did not have enough credibility because only the result was returned, without providing details about how that result was obtained. To solve this problem, posts were introduced that were characterized by the same type of personality as the user being tested for, but also the most frequent words used by them. Thus, the person concerned may notice certain commonalities between the user's posts and those given as an example.

#### **Reducing the Execution Time**

Due to the fact that the algorithms were building the model again every time we needed to use the application, and this process was time-consuming, we decided to save the important components of it to reduce the execution times. Finally, the response time decreased from 9 minutes to a few seconds.

### 3 Use Cases

#### 3.1 Nicole Kidman

Nicole Kidman<sup>13</sup> is a film actress. In 2002, she won the Academy Award for Best Actress in the movie “The Hours”<sup>14</sup>. We chose this one to see the correctness of the created algorithm. She is an INFJ (introverted, intuitive, sensitive, and judgmental) personality, according to (Zaki, 2020).

With the help of the application, we extracted the texts from her posts on Instagram and Twitter. For each word in these posts, we applied the steps described above: we removed the URL, removed the punctuation, converted uppercase letters to lowercase letters, and removed linking words. After this processing, we applied the XGBoost algorithm, and for each obtained word we assigned a value according to the number of occurrences in the text.



Figure 4. 70 most common words for Nicole Kidman

In Fig. 4, we can see the 70 most common words from her posts. Comparing this figure with Figure 3 (k), we can see some common words that appear most often: “little”, “thank”, “first”, “love”, “long”, “give”. After applying the algorithm to the resulting words, we obtained an INFJ personality that is consistent with its type. Thanks to the created application, we could understand a person’s character just by viewing their posts on social networks. Thus, we could tell in advance if we are talking to a certain person if we can resonate with them. Or we could find a person who is opposite to us so that we can grow socially or see a different perspective on life.

#### 3.2 Anna Wintour

Anna Wintour<sup>15</sup> is a British journalist who was the editor-in-chief of Vogue. She is an INTJ (Introverted, Intuitive, Thinker, and Judge) personality, according to (Zaki, 2020).

In Fig. 5, we see the most common words from her last posts. After processing we can see some common words like those of an INTJ (Fig. 3 (a)), for example: “post”, “love”, “never”, “life”, “great”, “want”. After applying the algorithm to the texts taken from her social networks, we obtained an INTP personality (Introverted, Intuitive, Thinker and Perceptive). We can see that the result obtained is 75% correct because it is wrong on the last class analyzed (the correct would have been J, and our algorithm predicted P).

<sup>13</sup> [https://ro.wikipedia.org/wiki/Nicole\\_Kidman](https://ro.wikipedia.org/wiki/Nicole_Kidman)

<sup>14</sup> <https://www.imdb.com/title/tt0274558/>

<sup>15</sup> [https://en.wikipedia.org/wiki/Anna\\_Wintour](https://en.wikipedia.org/wiki/Anna_Wintour)



Figure 5. 70 most common words for Anna Wintour

## 4 Conclusions

The application presented in this paper aimed to show users another way in which posts on social media accounts can be interpreted. With the help of the application, we can see the personality type of a certain user or text and we can see which known personalities have the same personality type. Through the experiments we have done, we can say that, yes, in the posts that a user makes, we often have elements specific to his personality. Thus, we can identify his personality by analyzing his posts on the social networks he uses.

As can be seen in the case studies, the algorithm could be improved in the future, at least in the last class: *judgment (J)/perception (P)* because it performs slightly worse than the other three. Also, we could collect data from users based on a form containing a personality test, and following the user's answers, we could characterize him from the point of view of the 16 types of personalities.

## Acknowledgements

Data processing and analysis in this paper were supported by the Competitiveness Operational Program Romania, under project SMIS 124759 - RaaS-IS (Research as a Service Iasi).

## References

- Mihai Baboi, Adrian Iftene, and Daniela Gifu. (2019). Dynamic microservices to create scalable and fault tolerance architecture. *Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 23<sup>rd</sup> International Conference KES2019. Procedia Computer Science*, 159:1035–1044.
- Ed Burns and Kate Brush. (2021). In-depth guide to machine learning in the enterprise. *Tech Accelerator: Tech Target*.
- Kendra Cherry. (2022). How the Myers-Briggs type indicator works. *Very Well Mind* <https://www.verywellmind.com/the-myers-briggs-type-indicator-2795583>.
- Philip J. Corr and Gerald Matthews. (2009). The Cambridge handbook of personality psychology (1. publ. ed.). *Cambridge: Cambridge University Press*.
- Brandon Cui and Calvin Qi. (2017). Survey analysis of machine learning methods for natural language processing for MBTI personality type prediction. *Final Report Stanford University*.
- Ciprian Gabriel Cuşmuliuc, Lucia Georgiana Coca, and Adrian Iftene. (2018). Identifying fake news on Twitter using Naive Bayes, SVM and Random Forest distributed algorithms. *Proceedings of the 13th Edition of the International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR-2018)*, pages 177–188.
- Ciprian Gabriel Cuşmuliuc, Matei A. Amarandei, I. Pelin, Vlad I. Cociorva, and Adrian Iftene. (2021). UAICs at CheckThat! 2021: Fake news detection. *Proceedings of the Working Notes of CLEF 2021 - Conference and Labs of the Evaluation Forum, Bucharest, Romania*, September 21-24, 2021, 2936:494–507.

- Brent M. Donnellan and Richard Robins. (2010). Resilient, overcontrolled, and undercontrolled personality types: Issues and controversies. *Social and Personality Psychology Compass*, 4, 11.
- Thomas W. Edgar and David O. Manz. (2017). Chapter 4 - exploratory study. *Research Methods for Cyber Security*, pages 95–130.
- Hans Jurgen Eysenck. (1994). 7 - personality: Biological foundations. *The Neuropsychology of Individual Differences*, pages 151–207.
- Rohith Gandhi. (2018). Support vector machine - introduction to machine learning algorithms. SVM model from-scratch. *Towards Data Science*.
- Matej Gjurkovic and Jan Snajder. (2018). Reddit: A gold mine for personality prediction. Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media, NAACLHLT2018, New Orleans, Louisiana, June 6, Association for Computational Linguistics, pages 87–97.
- Adrian Iftene, Daniela Gîfu, Andrei Remus Miron, and Mihai Ștefan Dudu. (2020). A real-time system for credibility on Twitter. *Proceedings of the 12th Language Resources and Evaluation Conference (LREC 2020)*, Marseille, France, May 2020, pages 6168–6175.
- Carl Gustav Jung. (1971). Psychological types. *Princeton University Press*, 6:136–147.
- Sedrick Scott Keh and I-Tsun Cheng. (2019). Myers-Briggs personality classification and personality-specific language generation using pre-trained language models. <https://arxiv.org/pdf/1907.06333v1.pdf>.
- Shedy Dee C. Mallari and Jose Maria G. Pelayo. (2017). Myers-Briggs type indicator (MBTI) personality profiling and general weighted average (GWA) of nursing students. Assessment, Counseling, Alumni and Placement Center (ACAP).
- Gerald Matthews and Kirby Gilliland. (1999). The personality theories of h. j. Eysenck and j. a. Gray: A comparative review. *Personality and Individual Differences*, 26:583–626, 03.
- Vishal Morde and Venkat Anurag Setty. (2019). XGBoost algorithm: Long may she reign! The new queen of machine learning algorithms taking over the world. . . . *Towards Data Science*.
- Isabel B. Myers and Peter B. Myers. (1980). Gifts differing: Understanding personality type. *Mountain View, CA: Davies-Black Publishing*.
- Sai Patwardhan. (2021). Simple understanding and implementation of kNN algorithm! *Analytics Vidhya*.
- Sunil Ray. (2017). 6 easy steps to learn naive bayes algorithm with codes in Python and R. *Analytics Vidhya*.
- Ghina Salsabila and Erwin Setiawan. (2021). Semantic approach for big five personality prediction on twitter. *Journal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 5:680–687, 08.
- Gautam Solanki. (2020). Multinomial Naive Bayes explained. *Great Learning, Artificial Intelligence, Machine Learning*.
- Mahmoud Zaki. (2020). Heaven and earth. A quick introduction to human behaviours. *Printed in the Egypt*.



## **CHAPTER 3**

### **ROMANIAN LINGUISTIC RESOURCES**



# MEDICAL CORPUS FROM ROMANIAN HOSPITALS

RAREȘ ARVINTE<sup>1</sup> AND DIANA TRANDABĂȚ<sup>1</sup>

<sup>1</sup>*Alexandru Ioan Cuza University,  
Faculty of Computer Science,  
Bd. Carol I, No.11, 700506, Iași, Romania*

## Abstract

After obtaining a considerable dataset from the collaboration with Iasi Infectious Diseases Hospital, we opted for research in sepsis-related diseases, especially the relevance of secondary diagnoses in identifying associations that could allow for early identification of sepsis. In this idea, the epicrisis needed to be parsed to obtain or summarize information useful to streamline the interpretive process of a doctor. Finally, following the consultation with specialists in the field, we modified the research area towards nosocomial infections and the applicability of the different existing technologies for the development of their early detection processes.

## 1 Introduction

In this paper, we introduce a medical database containing healthcare records and doctors' recommendations for patients hospitalized with a sepsis diagnosis. Each patient has an observation form containing information about the diagnosis, the medication he received, his medical progress, and an epicrisis, *i.e.*, a summarizing description of the phenomena observed by the doctor during hospitalization. The main application of this corpus is to seek ways to detect nosocomial infections.

Applying NLP techniques to medicine is not a new topic. Various directions have been investigated so far, from analyzing medical scientific articles to dissecting doctor-patient voice records, generating medical reports or investigating drug interactions. As always, limited research is available for less-resourced languages, as also discussed in (Cherednichenko et al., 2020).

## 2 Related Work

In Zweigenbaum et al. (2001), a framework for designing a medical corpus is described. Their proposed textual corpus is carefully engineered to represent a variety of medical languages and integrate various sources. The authors show that it is in the nature of medical corpora to lack context, thus an amount of subjectivity of the medical expert cannot yet be reproduced and applied to improve the objective expert systems. To solve this, their paper proposes to enhance medical texts with information obtained from online environments through web services or by integration with external systems. In the same context, medical corpus plays an important role in offering support systems for medical decisions (Kukhtevich et al., 2020).

Another approach investigates the interpretation of the epicrisis recorded by the doctor to identify key terms and obtain datasets using specific NLP tools (Langnickel et al., 2021). The authors propose the use of a specific pattern for these patient discharge reports, which however allows doctors a certain amount of freedom.

Our aim is to use the medical dataset to detect early cases of nosocomial infections. Antibiotic-associated diarrhea and colitis have been studied soon after the widespread use of antibiotics (Bartlett, 2006). In 1978, *C. difficile* was identified as the causative pathogen in the majority of cases of antibiotic-associated colitis; many early cases were attributed to clindamycin (Bartlett, 2006) (Bartlett et al., 1978). Increasing use of penicillin and cephalosporins led to the implication

of these antibiotic classes as precipitants as well. The appearance of *C. difficile* occurs on average in 1 in 10 patients hospitalized or in treatment facilities (McFarland et al., 1989). The biggest problem is the presence of those who transmit *C. difficile* and that they are asymptomatic. These people spread spores and act as a bridge of infection for other patients (Curry et al., 2013). It has therefore been observed that the concentration of toxins in asymptomatic patients is similar to that of patients in a more unstable state of health (Pollock et al., 2019).

The paper (Li et al., 2019) discusses the prediction based on the patient's medical records and depending on his medical evolution of cases of *C. difficile*, whether or not they become medically complicated, meaning that a morbid process or event has been superimposed on the disease, thus altering symptoms and modifying its course for the worse. Judging by this, a doctor can consider a statistical evolution of diagnostics and adapt to make their treatment more efficient. They had at their disposal a number of 1144 cases registered at the University of Michigan hospitals (UM) between October 2010 and January 2013. Two experts provided the discharge summaries, while a third one was used to remedy any misunderstandings between them. It should be added that they brought into discussion the making of multiple predictions based on certain pre-established data over a period of time. In order to identify these cases, the authors looked for the patients admitted with the respective diagnosis, which can be understood as the fact that they did not contact the infection from the respective hospital. Their data also contain demographic details (such as age or sex) and data about medical history, such as whether the patient was diagnosed with diabetes in the last 90 days, the medical record at hospitalization, and the daily observation record. The obtained results based on daily evolution and treatment applied to the patient indicate that for predictions where patients are in the early stages of hospitalization, the accuracy was 79% and for patients in their third day of hospitalization, the accuracy increases to 90%. This shows that the patient's hospitalization and the actions taken by the doctors during hospitalization have a direct impact on reaching the correct diagnosis.

As a reference for the methods applied by us, (Ijaz et al., 2018) was of great help through its content rich in explanations of a multitude of algorithms. One of the methods explained is DBSCAN, which is accompanied by examples helping to detect the outlier in a diabetes and hypertension dataset. The paper explains how to choose the minimum distance between neighbors using the "knee" method, but it does not try at all, nor does it take into account, the modification of the minimum number of neighbors to decide if a point can be considered the center of a cluster. The number chosen for this decision is a minimum of five neighbors to define a cluster.

### 3 Data Analysis

Following the collaboration with the Iasi Infectious Diseases Hospital, we managed to get approval to have access to the hospital database. A long enough step followed, aiming to determine which data exactly could be useful. After a small analysis of the available data, we considered selected data from the medical records of patients hospitalized in the period 2017-2020. The dataset consists of three large CSV files.

#### 3.1 Patient Data

The first file is a dataset containing details about the patients, demographics, and their diagnoses during the hospitalization. We have thus 6823 entries, representing the total number of medical records in the data set. Each patient has a unique ID to be identifiable across different files. For each record, there are 17 columns.

The first columns represent details about the patient's hospitalization, such as where they were hospitalized, for how long, their condition at hospitalization, and their condition at check-out. We also have a column that specifies the main diagnosis, which, in our case, is sepsis. But further follows an interesting column of possibly increased importance for research, namely a list of sec-

ondary diagnoses. These secondary diagnoses are presented in the international format called ICD-10-CM, a format that helps doctors to catalog diagnostic charts for the ease of placing a patient in a specific treatment grid. This format is easy to read but requires either specialized personnel to understand the data or the use of special interpretive dictionaries.

The following columns indicate demographic parameters, such as sex, age, and environment, which could have an impact on the development of statistics about patients affected by various diseases, categorized by the environment in which they live, and the type of communities. However, all data is anonymized, and no names or personal numbers are stored in our database.

Another interesting information is found in the column dedicated to epicrisis, which is a discharge report in natural language, edited in professional terms, describing the patient's journey throughout the hospitalization. This column is intended to be used later by doctors in case of transfers between hospitals or for investigating the patient's medical history. This column can have a major impact on natural language processing tasks, especially for summarizing medical records.

Finally, we have a column called ATI, usually associated with people in very poor health, containing details about how much that person was hospitalized in the intensive care section.

### 3.2 *Medication during Hospitalization*

This dataset contains details about all the medication administered to each patient, associated with the medical record in the previous patient dataset. Here one can find 215,085 rows containing names of medicines or chemical names of the products administered, together with the recommended administration manner. This dataset also contains the recommended quantities to be administered; together with the date of administration, these data are useful to identify returning patients, but also for automatically observing certain patterns to see if the medication doses can be improved and thus reducing costs.

### 3.3 *Medical Analyses*

Similar to the medication dataset, this dataset is associated with the first patient dataset, linked by the unique identification ID of each patient's medical record. Thus, here we have 642,235 rows that represent the medical analyses performed for various patients, divided into the different medical files.

In this dataset, there are columns such as the scientific name of the analysis and the obtained results, but also a column of observations, sometimes filled-in by doctors, but mostly left empty.

This dataset is also of major importance because patient analyses are a mandatory resource in determining treatment and prescribing medications. Having both analyses and prescriptions can lead to the development of an automatic prescription generator. It could also help doctors in their early careers who are supposedly not yet experienced to act quickly and efficiently in solving a patient's medical problem.

Having this database, we decided to focus currently on nosocomial infections<sup>1</sup>. By simply observing the data, we noticed that 302 medical records are of patients who have such infections. Since these infections are in a very large percentage developed in the hospital, we can conclude that a number of them could have been prevented if hospitals were better prepared or had an early detection system.

We see this kind of system customized for each patient according to his medical history, his current medical record, and hospitalization, but also from the perspective of the long-term efficiency of the hospital in detecting and preventing such infections.

---

<sup>1</sup> Infection(s) acquired during the process of receiving health care that was not present during the time of admission

### 3.4 The Data Conversion Process

The data received in the initial CSV format was not consistent, so a process of conversion and bringing it into an easy-to-follow form was needed. We created a data conversion system that makes the medical dataset easy to read for future classification and/or clustering tasks.

Figure 1 shows the architecture followed for data conversion, which brings the data into an easy-to-read format. The Pandas module is used to format data into objects easy to use by different classification methods.

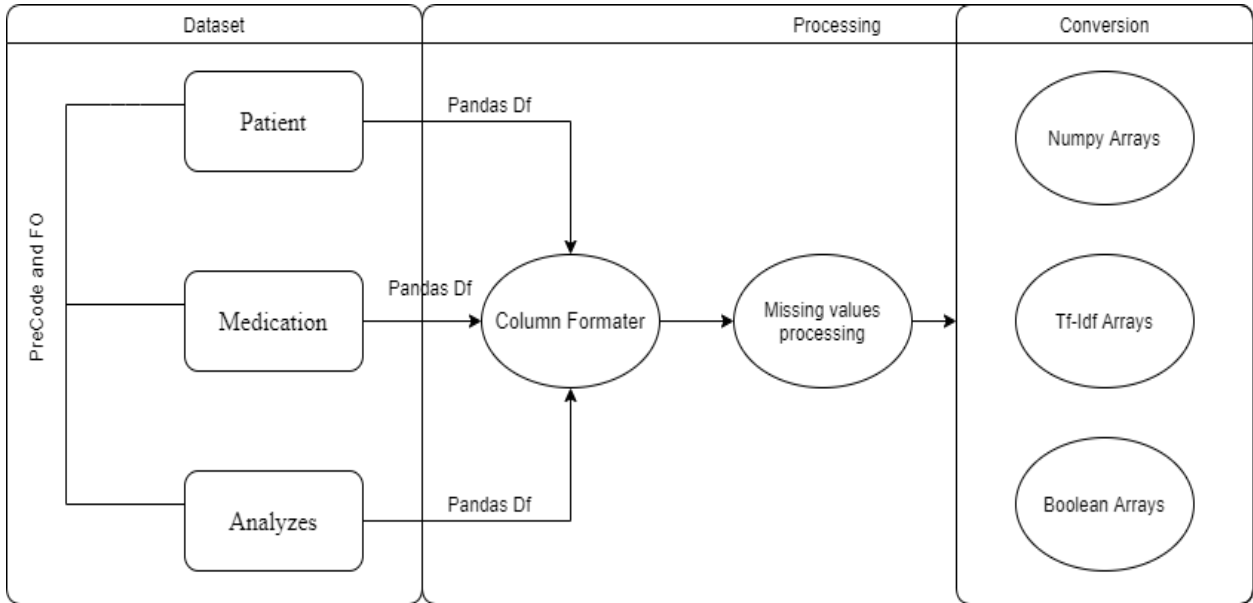


Figure 1. The data conversion process

The initial datasets were each saved in only one column, so the first step was to convert them to a more user-friendly column format by splitting each row into the column using commas as a delimiter. After this step, we noticed that some rows still remained merged, not converted to a readable version due to the fact that they also had a dividing space, and we split them also.

Some columns in the data looked pretty redundant, so we eliminated a number of columns that did not have a major impact on the analysis, namely the column of the main diagnosis (which is always sepsis, as it was the criteria for collecting data) or the columns with calendar data.

The next challenging step was converting the secondary diagnostic column, which initially presented a series of comma-separated diagnostics. The first conversion attempt was to assign each unique entry a unique identification number, but this led to faulty results. A better approach was to convert each line into an array of strings that represent secondary diagnoses, but there were difficulties in propagating the information in the form of lists so the complexity of the calculations raised exponentially. The final solution was to identify each unique diagnosis in the dataset and convert it to a new binary column, true or false, to determine if that diagnosis was identified for each patient record.

Figure 2 shows a snippet of secondary diagnoses, along with their total number of occurrences in the secondary diagnosis column.

For the column that represents the epicrisis of an entry, which was natural language text, we applied the tf-idf method to normalize it. Thus we arrived at the final dataset with 6,835 rows and 24,033 columns, where each row is associated with a registered medical file and the columns represent the extracted features. Figure 3 presents a snippet of the converted data. The obtained cleaned data will be used for different classification and statistical processes.

|      |                                                   | diag_sec | diag_sec_count |
|------|---------------------------------------------------|----------|----------------|
| 2    | R63.3 Dificultati si proasta administrare a al... |          | 3223           |
| 1    | D50.9 Anemia prin carenta de fier nespecificata   |          | 2831           |
| 20   | I10 Hipertensiunea esentiala (primara)            |          | 2202           |
| 8    | B37.0 Stomatita prin Candida                      |          | 1879           |
| 4    | E86 Hipovolemia deshidratarea                     |          | 1333           |
| ...  | ...                                               |          | ...            |
| 935  | Z86.12 Istoric personal de poliomielifita         |          | 1              |
| 937  | D69.4 Alte trombocitopenii primare                |          | 1              |
| 938  | I69.0 Sechelele hemoragiei subarahnoidale         |          | 1              |
| 942  | H49.0 Paralizia celui de-al treilea nerv [ocul... |          | 1              |
| 1496 | G93.6 Edem cerebral                               |          | 1              |

Figure 2. Unique secondary diagnoses count

| 1  | TIP_INTERNARE                | STARE_EXT | TIP_EXT  | DIAG_SECUNDARE                                           | SEX | VARSTA | MEDIU | EPICRIZA    | ATI       |
|----|------------------------------|-----------|----------|----------------------------------------------------------|-----|--------|-------|-------------|-----------|
| 2  | URGENTA                      | AMELIORAT | EXTERNAT | B27.9 Mononucleoza infectioasa nespecificata,I10 Hi      | M   | 76     | r     | Pacient in  | NO DATA   |
| 3  | TRANSFER INTER-SPITALICESC   | AMELIORAT | EXTERNAT | I27.0 Hipertensiunea pulmonara primara,I33.0 Endoc       | M   | 35     | u     | NO DATA     | NO DATA   |
| 4  | TRANSFER INTER-SPITALICESC   | AMELIORAT | EXTERNAT | E11.9 Diabet mellitus tip 2 fara complicatii,I11.9 Card  | M   | 69     | u     | NO DATA     | NO DATA   |
| 5  | TRIMITERE MEDIC SPECIALIST A | DECEDAT   | DECEDAT  | B35.1 Tinea unghiilor,B35.3 Tinea piciorului,B37.88 C    | M   | 84     | u     | NO DATA     | NO DATA   |
| 6  | URGENTA                      | AMELIORAT | EXTERNAT | B96.88 Alti agenti bacterieni si nespecificati drept cau | M   | 61     | r     | Pacient in  | NO DATA   |
| 7  | URGENTA                      | AMELIORAT | EXTERNAT | D50.9 Anemia prin carenta de fier nespecificata,E04.2    | F   | 80     | r     | NO DATA     | NO DATA   |
| 8  | TRIMITERE MEDIC SPECIALIST A | AMELIORAT | EXTERNAT | B96.2 Escherichia coli [E. coli] cauza unor boli clasate | F   | 79     | u     | NO DATA     | NO DATA   |
| 9  | TRIMITERE MEDIC SPECIALIST A | AMELIORAT | EXTERNAT | B96.2 Escherichia coli [E. coli] cauza unor boli clasate | F   | 89     | r     | Pacienta ir | NO DATA   |
| 10 | URGENTA                      | AMELIORAT | EXTERNAT | D64.8 Alte anemii specificate,D69.6 Trombocitopenia      | M   | 44     | u     | Bolnav ext  | NO DATA   |
| 11 | TRIMITERE MEDIC SPECIALIST A | AMELIORAT | EXTERNAT | D50.9 Anemia prin carenta de fier nespecificata,D53.4    | M   | 32     | u     | Bolnav in   | NO DATA   |
| 12 | URGENTA                      | AMELIORAT | EXTERNAT | D50.9 Anemia prin carenta de fier nespecificata,J02.9    | F   | 5      | u     | Copil in va | NO DATA   |
| 13 | URGENTA                      | AMELIORAT | EXTERNAT | I10 Hipertensiunea esentiala (primara),J18.1 Pneumoi     | F   | 77     | r     | Pacient in  | NO DATA   |
| 14 | URGENTA                      | VINDECAT  | EXTERNAT | A04.7 Enterocolita prin Clostridium difficile,B37.0 Sto  | F   | 36     | r     | Pacienta 3  | NO DATA   |
| 15 | TRIMITERE MEDIC SPECIALIST A | AMELIORAT | EXTERNAT | B37.0 Stomatita prin Candida,B96.2 Escherichia coli [    | F   | 65     | u     | NO DATA     | NO DATA   |
| 16 | URGENTA                      | AMELIORAT | EXTERNAT | B00.1 Dermita veziculara datorita virusului herpetic,D   | F   | 83     | r     | Pacienta ir | NO DATA   |
| 17 | URGENTA                      | AMELIORAT | EXTERNAT | B37.0 Stomatita prin Candida,D50.9 Anemia prin care      | M   | 47     | u     | NO DATA     | NO DATA   |
| 18 | TRIMITERE MEDIC SPECIALIST A | AMELIORAT | EXTERNAT | B96.2 Escherichia coli [E. coli] cauza unor boli clasate | F   | 91     | r     | Pacienta ir | NO DATA   |
| 19 | URGENTA                      | AMELIORAT | EXTERNAT | E78.0 Hipercolesterolemie esentiala,E78.8 Alte tulbur    | M   | 50     | u     | Pacient in  | NO DATA   |
| 20 | TRANSFER INTER-SPITALICESC   | AMELIORAT | EXTERNAT | B37.0 Stomatita prin Candida,E11.9 Diabet mellitus ti    | F   | 78     | r     | Pacienta ir | Dec 28 20 |

Figure 3. Snippet of the converted data

## 4 Results

The first major result obtained during this research was of administrative nature, namely establishing collaborations with the medical staff and obtaining access to the medical data, of course, anonymized, which we will further use in this research.

Clustering is applied in many branches of society in order to organize data, based on a set of patterns that abstract them. Thus, the data are organized in such a way that each entry more or less fits a category or a group.

In a large-scale view, there are two types of data partitioning algorithms: hierarchical and non-hierarchical clustering. Hierarchical clustering algorithms construct a hierarchy of partitions, represented as a dendrogram in which each partition is nested within the partition at the next level in the hierarchy. Partitional clustering algorithms generate a single partition, with a specified or estimated number of nonoverlapping clusters of the data, in an attempt to recover natural groups existing in the data. One of the important problems in partitional clustering is to find a partition of the given data, with a specified number of clusters, that minimizes the total within-cluster variation. To find the optimal number of clusters, we used the *silhouette* method. The simplest and most popular among iterative and hill-climbing clustering algorithms is the K-means algorithm (KMA) which attributes the information to the nearest mean cluster (Krishna et al., 1999).

Figure 4 shows the distribution made by k-means with three clusters on the patient files. To plot data, we used PCA (Principal Component Analysis) to reduce the dimensions, as data can

thus be seen as more friendly for us. After looking at the results, we can conclude that the data used are well established, and possible patterns are easily observable, although the number of classes set at 3 might have been too limited.

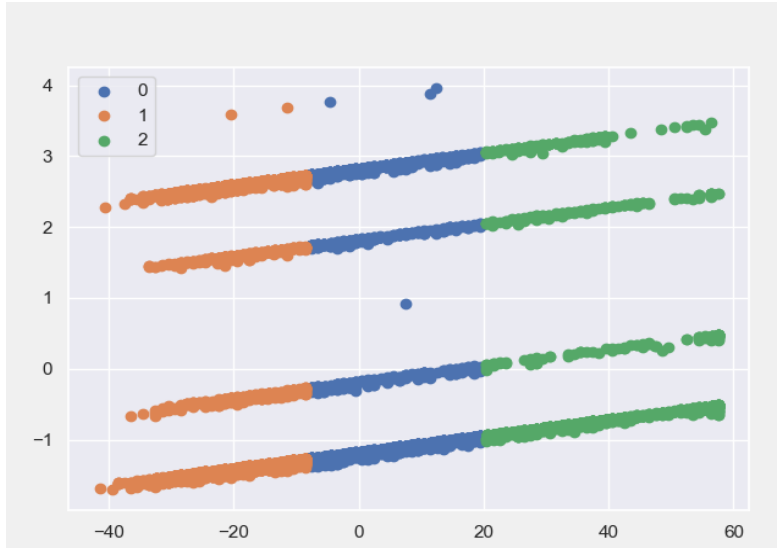


Figure 4. K-means results

DBSCAN is a popular density-based clustering algorithm that is found in many clustering tools and has been used quite often in applications. DBSCAN estimates the minimum density between a number of neighbors within an arbitrarily chosen radius. Thus, nodes that have a larger number of neighbors than the threshold are considered central points. As such, DBSCAN tries to find the areas that meet the minimum density and are separated from the low-density areas. DBSCAN does not estimate the density between points, but the density between all neighbors within a certain radius (Schubert et al., 2017).

One can notice in Figure 5 that DBSCAN found a different number of clusters, above the number found by the k-means algorithm using the *silhouette* score. In this case, nineteen clusters were formed, using an *eps* (i.e., the distance between neighbors) of 0.5 and the minimum number to define a cluster of 5 points. More tests should be performed where the minimum number of points that define a cluster and the *eps* are gradually changed.

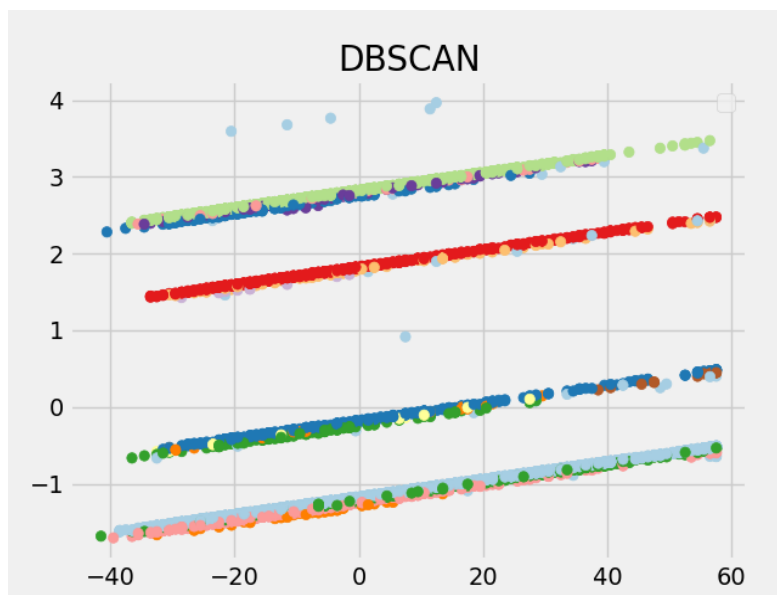


Figure 5. DBSCAN results

## 5 Conclusion

This paper has introduced a linked medical dataset containing patient data, medication, and analyses of patients during hospitalization. The data is anonymized and various cleaning steps have been performed. Various discoveries and possible applications can be made in advancing medical programs, treating patients, preventing infections, and possibly finding new associations based on medications administered and related tests.

Two clustering exercises have been applied to this corpus in order to test its usability in a computational context, one with K-means and one with DBSCAN. Further analysis on the early identification of patients prone to nosocomial infections is currently being developed.

In conclusion, this Romanian medical database has a huge potential in the development of health care applications. Our work will continue also in the direction of obtaining new and various data, in correlation with future advices from medical experts in these infections.

## References

- Bartlett J. G. (2006). Narrative review: the new epidemic of *Clostridium difficile*-associated enteric disease. *Annals of internal medicine*, 145(10), 758–764. <https://doi.org/10.7326/0003-4819-145-10-200611210-00008>
- Bartlett, J. G., Moon, N., Chang, T. W., Taylor, N., & Onderdonk, A. B. (1978). Role of *Clostridium difficile* in antibiotic-associated pseudomembranous colitis. *Gastroenterology*, 75(5), 778–782.
- Cherednichenko, O., Kanishcheva, O., Yakovleva, O., & Arkatov, D. (2020). Collection and Processing of a Medical Corpus in Ukrainian. In *COLINS* (pp. 272-282).
- Curry, S. R., Muto, C. A., Schlackman, J. L., Pasculle, A. W., Shutt, K. A., Marsh, J. W., & Harrison, L. H. (2013). Use of multilocus variable number of tandem repeats analysis genotyping to determine the role of asymptomatic carriers in *Clostridium difficile* transmission. *Clinical infectious diseases: an official publication of the Infectious Diseases Society of America*, 57(8), 1094–1102. <https://doi.org/10.1093/cid/cit475>
- Ijaz, M. F., Alfian, G., Syafrudin, M., & Rhee, J. (2018). Hybrid prediction model for type 2 diabetes and hypertension using DBSCAN-based outlier detection, synthetic minority over sampling technique (SMOTE), and random forest. *Applied Sciences*, 8(8), 1325.
- Krishna, K., & Murty, M. N. (1999). Genetic K-means algorithm. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 29(3), 433-439.
- Kukhtevich, I. I., Goryunova, V. V., Goryunova, T. I., & Zhilyaev, P. S. (2020, August). Medical decision support systems and semantic technologies in healthcare. In *Russian Conference on Digital Economy and Knowledge Management (RuDECK 2020)* (pp. 370-375). Atlantis Press.
- Langnickel, L., Krockauer, K., Uebachs, M., Schaaf, S., Madan, S., Klockgether, T., & Fluck, J. (2021). Information Extraction from German Clinical Care Documents in Context of Alzheimer’s Disease. *Applied Sciences*, 11(22), 10717.
- Li, B. Y., Oh, J., Young, V. B., Rao, K., & Wiens, J. (2019, May). Using machine learning and the electronic health record to predict complicated *Clostridium difficile* infection. In *Open forum infectious diseases* (Vol. 6, No. 5, p. ofz186). US: Oxford University Press.
- McFarland, L. V., Mulligan, M. E., Kwok, R. Y., & Stamm, W. E. (1989). Nosocomial acquisition of *Clostridium difficile* infection. *New England Journal of mMedicine*, 320(4), 204-210.
- Pollock, N. R., Banz, A., Chen, X., Williams, D., Xu, H., Cuddemi, C. A., Cui, A. X., Perrotta, M., Alhassan, E., Riou, B., Lantz, A., Miller, M. A., & Kelly, C. P. (2019). Comparison of *Clostridioides difficile* Stool Toxin Concentrations in Adults With Symptomatic Infection and Asymptomatic Carriage Using an Ultrasensitive Quantitative Immunoassay. *Clinical infectious diseases: an official publication of the Infectious Diseases Society of America*, 68(1), 78–86. <https://doi.org/10.1093/cid/ciy415>
- Schubert, E., Sander, J., Ester, M., Kriegel, H. P., & Xu, X. (2017). DBSCAN revisited, revisited: why and how you should (still) use DBSCAN. *ACM Transactions on Database Systems (TODS)*, 42(3), 1-21.
- Zweigenbaum, P., Jacquemart, P., Grabar, N., & Habert, B. (2001). Building a text corpus for representing the variety of medical language. In *MEDINFO 2001* (pp. 290-294). IOS Press.



# GIS-BASED PUBLISHING SYSTEM FOR THE ROMANIAN LINGUISTIC ATLAS BY REGIONS – SYNTHESIS

SILVIU-IOAN BEJINARIU<sup>1,2</sup>, RAMONA LUCA<sup>2</sup>, FLORIN IFTENE<sup>2</sup>,  
MANUELA NEVACI<sup>1</sup>, CARMEN IRINA FLOAREA<sup>1</sup>, AND  
NICOLAE SARAMANDU<sup>1</sup>

<sup>1</sup> “Iorgu Iordan – Al. Rosetti” Institute of Linguistics of the Romanian Academy,  
13 Septembrie Way, 13, Bucharest, Romania

<sup>2</sup> Institute of Computer Science, Romanian Academy Iași Branch,  
T. Codrescu Street, 2, Iasi, Romania

## Abstract

The Romanian Linguistic Atlases study language variation in terms of geographical distribution and temporal evolution. The publication of Romanian regional atlases began in 1967, but only some of them published after 2000 were produced using computerized methods, and only for them the linguistic information is stored in specialized databases which makes it available for further analysis. After 1990, the elaboration of a synthesis of regional atlases began, in which dialectal forms and phenomena are rendered on the map using symbols. In this paper, a novel Geographic Information System (GIS) based framework for the Romanian Linguistic Atlas by Regions – Synthesis publishing is proposed. For increased flexibility, the linguistic information is stored in an XML data collection. The interface gives the possibility to import linguistic information from regional atlases if available or to edit it from scratch, manages the graphic symbols library, and creates the legends and the GIS-based maps projects. The maps management and printing are performed using a free open-source GIS package. To keep compatibility, the system allows creating the maps projects using the old-style incomplete data collections used for the previously published volumes. The proposed system is already used to prepare the maps of the next volume of the Romanian Linguistic Atlas by Regions – Synthesis.

## 1 Introduction

Dialectology is a subfield of sociolinguistics that studies language with its territorial variants: dialects, subdialects, and speeches. The language variations can be: *diatopic* when variations depend on the geographical or territorial criterion; *diastratic* when the spoken language is influenced by social parameters such as instruction, social position, gender, and age; and *diaphasic* which is determined by the intention of the speaker (Saramandu and Nevaci, 2013). In linguistic geography, the linguistic material is shown on dialectal maps in order to analyze the variations (Saramandu and Nevaci, 2018). In the last decades, the research in the field was continued by developing new theoretic-methodological instruments (Olariu and Olariu, 2014). The use of information technology facilitates the creation of complex dialects databases as well as the linguistic atlases publishing in classical printed format and also their accessibility using the Web interactive interfaces. By including analyzes related to the *diasexual* and *diagenational* variations that detail more the *diastratic* variation, “the one-dimensional linguistic geography becomes multidimensional” (Olariu and Olariu, 2014).

In Romania, the Linguistic Atlases have a history of more than a century which began in 1898 when Gustav Weigand started to publish the *Linguistischer Atlas des dacorumänischen Sprachgebietes* (WLAD). After the investigations carried out in the interwar period, three volumes of the *Romanian Linguistic Atlas* (in Romanian: *Atlasul Lingvistic Român*, abrev. *ALR*) with analytical maps were published by Sever Pop and Emil Petrovici. Also, three volumes of the *Small Ro-*

*manian Linguistic Atlas* (in Romanian: *Micul Atlas Lingvistic Român*) which contains color synthetic maps were published until WWII. The publication continued after 1956 with seven volumes of the *Romanian Linguistic Atlas – new series* (in Romanian: *Atlasul Lingvistic Român. Serie nouă*) published until 1972 and four volumes of the *Small Romanian Linguistic Atlas - new series* (in Romanian: *Micul Atlas Lingvistic Român. Serie nouă*) until 1981.

A more detailed investigation was carried out by the dialectologists from Bucharest, Cluj-Napoca, Iași, and Timișoara between 1960 and 1980 for the publication of the *New Romanian Linguistic Atlas by Regions* (in Romanian: *Noul Atlas Lingvistic Român pe Regiuni*). At the initiative of Emil Petrovici, the investigation network was enriched to about 1000 survey points for the Daco-Romanian dialect, and the publication started in 1967. It continues for some of the 7 historical regions: Banat, Crișana, Maramureș, Transylvania, Moldova and Bukovina, Muntenia and Dobrogea, and Oltenia (Saramandu and Nevaci, 2018). In the same period, in the Republic of Moldova, “as part of the series of these regional atlases” (Pavel, 1993), the *Romanian Linguistic Atlas by Regions. Bessarabia, Northern Bukovina, and Transnistria* was published but only the last volume was edited using the Latin alphabet.

Saramandu (1990) posed “the problem of elaborating an atlas of regional linguistic atlases or a synthesis of the New Romanian Linguistic Atlas by Regions”. Three volumes with interpretative maps were published under the title *Romanian Linguistic Atlas by Regions. Synthesis* (abbrev: *ALRR.Synthesis*, in Romanian: *Atlasul Lingvistic Român pe Regiuni. Sinteză*) coordinated by Nicolae Saramandu (Fig. 1). It capitalizes the dialectal material from the seven regional linguistic atlases from Romania (Banat, Crișana, Maramureș, Transylvania, Moldova and Bukovina, Muntenia and Dobrogea, Oltenia), to which is added the moldovan linguistic atlas published in the Republic of Moldova. The linguistic material from *ALRR.Synthesis* is much richer than the one contained in ALR and provides a very valuable documentary basis for studying the dialectal structure of current Romanian (Saramandu and Nevaci, 2018). The elaboration of linguistic maps in the context of these interpretative atlases is an innovative scientific method because the analysis of lexical units is carried out from the onomasiological, semasiological, and motivational points of view (Constantin, 2021). The network consists of more than 1200 investigation points, including those from the Republic of Moldova and Ukraine.

Most of the regional atlases were prepared using traditional techniques, respectively the content was drawn by specialized cartographers. The use of IT solutions for the elaboration of linguistic atlases started after 2000, thus for the old atlases, the information has to be digitalized again because almost all data included in these atlases have to be included in the synthesis. However, the maps included in *ALRR.Synthesis* are interpretative, which means that the phonetic transcription is specified in the legends only. This particularity was exploited in the first three volumes and probably will be used in the next volume. The survey points were grouped according to the similarity of the answers obtained during the survey by analyzing the maps in the regional atlases, and the phonetic transcription is edited only one time in the legend.

In the third volume of *ALRR.Synthesis*, the maps are made using a computer application which has a major limitation. The forms used are numbered and only their index in the list of forms is stored in Excel files. Thus, the actual form and its phonetic transcription are not stored, making further processing or analysis difficult. In order to eliminate this shortcoming, in this paper it is proposed an information system in which the linguistic information is stored in a database giving the possibility to perform later other linguistic processing or analysis. Also, the quality of the automatically generated drawings will be significantly improved.

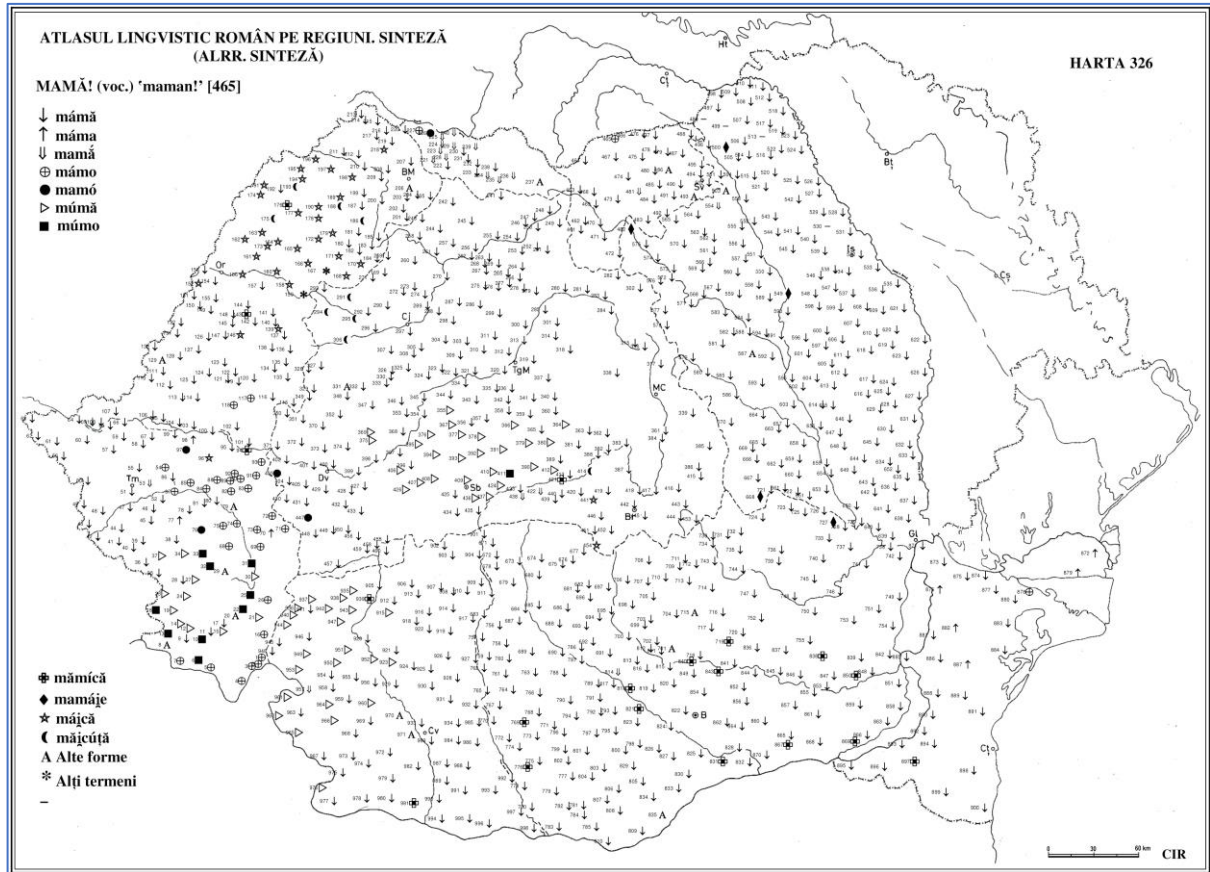


Figure 1. Map 326 from Romanian Linguistic Atlas by Regions. Synthesis, vol III

## 2 ALRR.Synthesis Information System

The proposed information system contains two main components: the *Legend Manager* and the *Map Generator* and it is presented in Fig.2. It also uses *QGIS*<sup>1</sup>, an open-source Geographic Information System package to generate and view the maps. The system is designed to manage the complete linguistic information required to publish the *ALRR.Synthesis* but, to keep compatibility with the old implementations, it is able to use the old data collections.

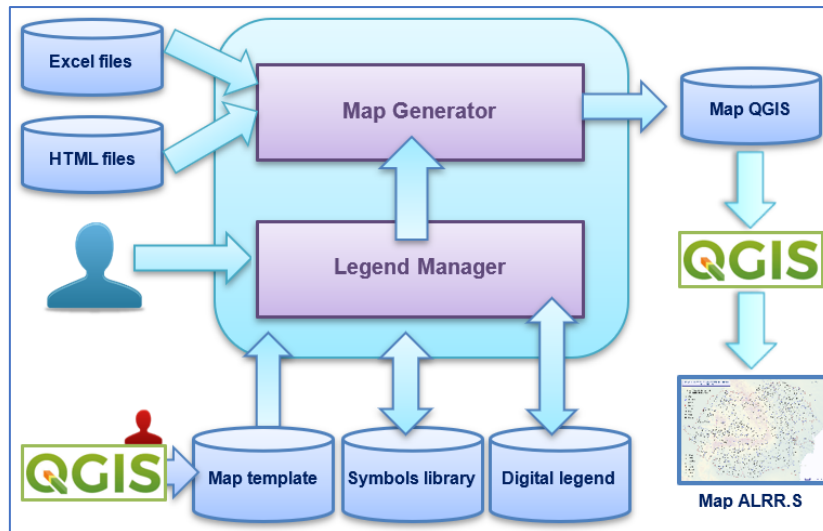


Figure 2. The structure of the proposed system for the realization of the ALRR.Synthesis

<sup>1</sup> QGIS. A Free and Open Source Geographic Information System, <https://qgis.org/en/site/>. Last accessed at 8.09.2022.



The Symbols library is a collection of images in SVG format, the only one accepted by QGIS to draw external symbols. It contains about 340 symbols created with the FontForge application from 15 fonts that contain glyphs for graphical symbols. Some of these symbols are similar, probably defined by different authors of maps, others are no longer used, but all of them have been kept because they were used in previous versions of the system. The symbols are referenced by their SVG file names of the form “<FontCode>\_<CharCode>.svg” where *FontCode* is a sequence of five characters that encode the source font name and *CharCode* is a sequence of three characters containing the code of the corresponding character in the font. To allow validation of the correctness of the symbol names, the existence of the files as well as to store some properties, the Symbols library is described in the “*alrss.alrsymb*” file in XML format (Fig. 5). Apart from font name and character code, for each symbol the following information is stored: character set, category (circles, squares, triangles, diamonds, stars, math, and so on), and scaling factor. The last property is required to allow the symbols having a similar size on the map.

```
<Symbol Code="GSIDS_143" Font="GSI Default Symbols" UnicodeChar="143" CharSet="2" Name="" Category="Miscellaneous" ScaleFactor="1">
</Symbol>
<Symbol Code="GSIDS_145" Font="GSI Default Symbols" UnicodeChar="145" CharSet="2" Name="" Category="Miscellaneous" ScaleFactor="2">
</Symbol>
<Symbol Code="GSIDS_146" Font="GSI Default Symbols" UnicodeChar="146" CharSet="2" Name="" Category="Miscellaneous" ScaleFactor="2">
</Symbol>
<Symbol Code="GSIGS_091" Font="GSI Geologic Symbols" UnicodeChar="91" CharSet="2" Name="" Category="Crosses" ScaleFactor="1.5">
</Symbol>
<Symbol Code="GSIGS_114" Font="GSI Geologic Symbols" UnicodeChar="114" CharSet="2" Name="" Category="Circles" ScaleFactor="1">
</Symbol>
<Symbol Code="GSIGS_120" Font="GSI Geologic Symbols" UnicodeChar="120" CharSet="1" Name="" Category="Squares" ScaleFactor="2">
</Symbol>
```

Figure 5. Description of symbols in the library

### 3 ALRR.Synthesis Map Template and Layout

To get the final map, the following steps have to be performed. First, the actual map is automatically created using the *MapGenerator* application. Then the map is placed on the layout which includes also the page borders, titles, map number, the author’s name, and the map scale.

#### 3.1 ALRR.Synthesis Map Template

The maps are created using a predefined template prepared using the QGIS package. The map is described in QGIS by a project file (“*.qgs*”). It contains the layer definitions including all their drawing styles. The map’s elements are organized in layers according to their common features but also to the type of graphic element. The layers are described using the shape file format, thus for each layer, there are three files: “*.shp*” (geometric description of the elements), “*.dbf*” (internal database which contains a record for each element in the layer), and “*.shx*” which is the index file. The relations with the linguistic information are described using the link fields in the internal database. All the layers in the template are described in geographical coordinates using the EPSG:31700, Dealul Piscului 1970<sup>3</sup>, the Romanian local coordinate system.

The *MapGenerator* application automatically updates the content of the template by customizing it for each linguistic map. To facilitate this, a coding mode that, in the file names, includes the question number and the title words, is used. Thus, <Map\_000\_MapName.qgs> is the name of the QGIS project containing the template. During map generation, the sequence <000> is replaced by the question number for which the word title is the answer, and the sequence <MapName> is replaced by the transliteration of the word title. Also, the template contains a layer of points <000\_MapName.shp> that corresponds to the survey points of the linguistic map. This layer is the only one customized in the generated map and therefore, its name will be modified in the same way as the name of the project.

Graphically, each map element is displayed using a symbol and a label. The symbol is not necessarily a graphic symbol, it can be a dot, line, or polygon. All the display properties (line color,

<sup>3</sup> EPSG Projection 31700, <https://spatialreference.org/ref/epsg/dealul-piscului-1970-stereo-70/>.

line weight, fill color, font, color, and text size) are stored in the QGIS project file in each layer's description. The display properties may vary from one element to another depending on some conditions related to the internal database content. The following layers are defined in the template:

- *000\_MapName* – the layer of points representing the survey points in *ALRR.Synthesis*. This layer is related to linguistic information and it is customized during map generation. It contains 1203 survey points out of which 998 are from Romania and the remaining 205 are from the Republic of Moldova and Ukraine in which the daco-romanian dialect is spoken. Both, the label and the symbol are displayed on the map to represent the survey point code and the graphic symbol associated with the answer.
- *Page\_Limits* – polygon layer; it is used to indicate the limits of the drawing but it is not actually displayed.
- *Page\_Border* – polygon layer, represents the map border. The associated symbol is of the line type and the label is not used.
- *Frontiers\_External* – line layer that is used to draw the external frontiers on the map. The associated symbol is line type and no labels are displayed.
- *Frontiers\_Regional* – line layer that is used to draw the boundaries of regions in Romania, as they are defined in the Regional Linguistic Atlases. The associated symbol is line type and no labels are displayed.
- *Cities* – points layer used to display the important cities and capitals on the map. In the internal database, there is a flag indicating the category of the city (major city/capital) and its name. The display on the map is conditional, in the sense that the size and the color of the point used as a symbol as well as the font size used for the label (city name) depend on the category indicated in the flag.
- *Rivers* – line layer used to display important watercourses; also, in this case, a flag is used to specify whether the watercourse is an ordinary or major river (the case of the Danube). Depending on the value of this flag, there is chosen the thickness of the line used as a symbol in the drawing; no labels are displayed.
- *Lakes* – polygon layer used to show on the map the water accumulations (lakes). Its elements are drawn using a polygon as a symbol; the labels are not displayed, although the names of the lakes are available in the internal database.
- *000\_MapName\_AreaRo* – polygon layer that can be used to draw hatched areas; it contains one polygon representing the vicinity for each survey point located in Romania; the polygons network is actually the Voronoi diagram of the survey points network that was intersected with the polygon in the *Frontiers\_External* layer. From a visual point of view, common phonetic features or identical answers can be better highlighted on maps using hatched areas, as in the synthetic maps in regional atlases.
- *000\_MapName\_AreaRoMd* – polygon layer, defined as the previous layer with the notice that it includes the polygons for all the survey points located in Romania, the Republic of Moldova, and Ukraine.
- *DEM\_ALR\_S* – raster layer that is used to display in the background the relief of the studied area. The image is created using the digital elevation model (DEM) drawn with a convenient color palette.

Most layers in the map template have been edited in QGIS using free geographic data sources available on the internet: Digital Elevation Model of Europe<sup>4</sup>, Administrative boundaries<sup>5</sup>, and

<sup>4</sup> Digital Elevation Model of Europe, © U.S. Geological Survey's Center for Earth Resources Observation and Science (EROS), <https://databasin.org/search/#query=30%20arc-second%20DEM%20of%20Europe>. Last accessed at 8.09.2022.

<sup>5</sup> Administrative boundaries, © EuroGeographics, <http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/countries#countries14>. Last accessed at 8.09.2022.

Hydrographic network<sup>6</sup> while the *000\_MapName* layer was manually digitalized using the templates from earlier versions of *ALRR.Synthesis*. Fig. 6 shows the *ALRR.Synthesis* map template.

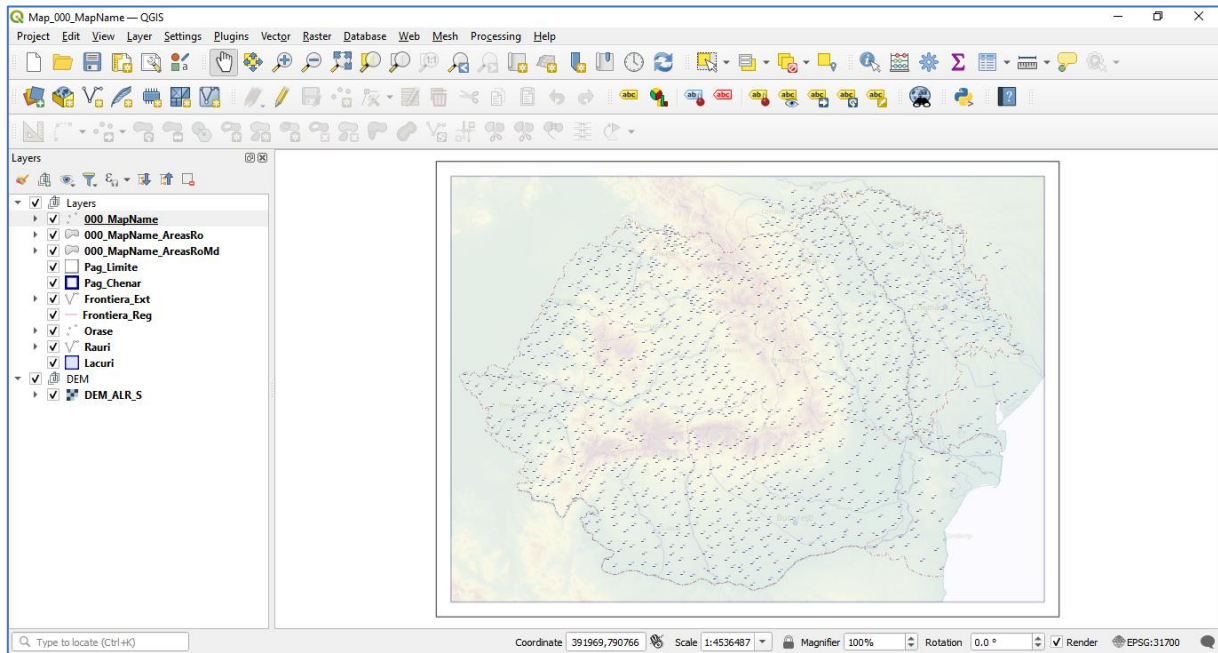


Figure 6. *ALRR.Synthesis* map template in QGIS interface

As it was already mentioned, only one layer *000\_MapName* is customized during the map generation. In fact, the changes are made in the internal *000\_MapName.dbf* table associated with the layer. It contains the following fields: *CountryCode*, *PointCode*, *Form*, *Symbol*, *SymbolColor*, *PointName*, *City*, *Country*, *Region* and *Country*. The values of some fields are not used (*City*, *Country*, *Region* and *Country*) but can be useful in further developments or statistical analysis of the language. The *CountryCode* value is '1' for the survey points in Romania and '2' for the points located in the Republic of Moldova or Ukraine and it allows one to select if the symbols for the points abroad are displayed or not. The *PointCode* is the primary key that uniquely identifies the survey point and it is also displayed on maps. The *Form* identifies the answer obtained in the survey and its value is updated during the map generation. The *Symbol* and *SymbolColor* are the code of the graphic symbol and its color on the map. These two values are also updated during the map generation.

The linguistic information is shown on the map using the renderer of the *000\_MapName* layer which is actually renamed during the map generation. For each element in the *000\_MapName* layer, the code of the survey point and the graphic symbol specified using an SVG (Scalable Vector Graphics) file are displayed. A rule-based renderer is defined, with different properties for each symbol as shown in Fig. 7. The first column (Label) contains the name of the rule with a checkbox that allows one to enable/disable it. For easy identification, the name of the rule is the same as the symbol code which is displayed if the rule is satisfied. The second column contains the condition to be satisfied, where SYMBOL is the content of the corresponding field in the internal database of the survey points. Each rule can be edited: the user can choose the symbol code, the size, color, and many other properties. It must be noticed that in this project, the size of the symbols, as well as the size of the other elements, is given in map units (meters). This makes the size ratio of the elements in the map remain constant regardless of the zoom at which the display is made.

<sup>6</sup> Hydrographic network (large rivers and lakes), © European Environment Agency (EEA), <https://www.eea.europa.eu/data-and-maps/data/wise-large-rivers-and-large-lakes>. Last accessed at 8.09.2022.

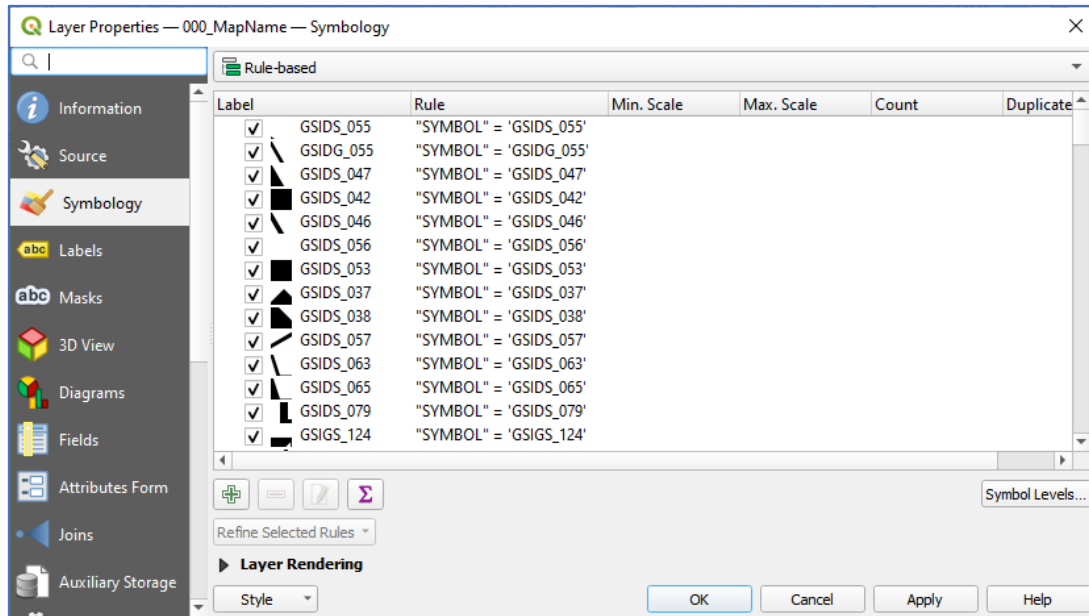


Figure 7. Definition of symbol renderer for the 000\_MapName layer

### 3.2 ALRR.Synthesis Layout

The map template presented in the previous section contains all the geographic-related elements that are usually displayed on maps. Apart from these, the printed map includes also other elements like title, legend, and scale which need to be included in the map layout. As in the case of the map template, the layout has been designed using the QGIS package interface, and it is customized by the *MapGenerator* application. All layout information is stored in the QGIS project file. The same encoding method (question number and title word) is used also for the layout's name. The map layout contains the following items:

- *Map\_ALRS* – the generated map itself,
- *Label\_Atlas\_Title* – text label containing the name of the atlas, in Romanian: “ATLASUL LINGVISTIC ROMÂN PE REGIUNI. SINTEZĂ (ALRR. SINTEZĂ)”. This label is the same for all pages of the atlas, so it does not change during the map generation;
- *Rectangle\_Atlas\_Title* – a bounding rectangle for the title of the atlas;
- *Label\_Map\_Title* – is a text label placed in the upper left corner just below the title of the atlas. This label contains the title word studied, the question number, and the question for which it is an answer;
- *Label\_Map\_Number* – text label placed in the upper right corner indicating the map's number,
- *Label\_Map\_Author* – text label placed in the lower right corner containing the initials of the author's name;
- *Map\_Scale* – a graphic element automatically generated by QGIS containing the map scale; placed in the lower right corner of the drawing;
- *Label\_Legend\_1*, *Label\_Legend\_2*, and *Label\_Legend\_3* – text labels reserved for the map legend. The three legends are placed in the upper left, lower left, and lower right corners. Usually, two such items are enough for the legend's content.

In Fig. 8 the layout components in the QGIS Layout Manager application window are shown which allows customizing a wide set of properties of these components: position, size, font, contour color, fill color, etc. The survey points do not have a uniform density on the map, therefore overlaps of the symbols and/or codes may appear (e.g., in the Maramureș area). These situations are solved by QGIS which automatically repositions the labels or symbols to minimize overlaps and the user also has the possibility to establish a priority order for the map's elements drawing.

To allow obtaining the best quality of the printed materials it is recommended to prepare different layouts for each paper format.

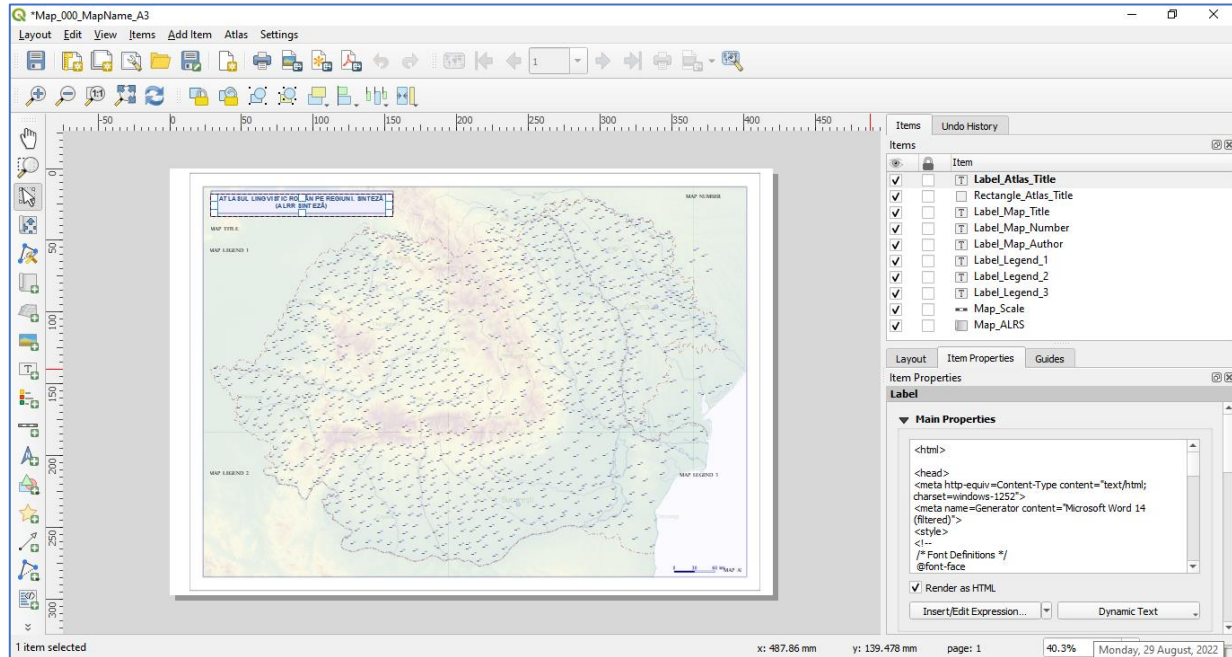


Figure 8. ALRR.Synthesis map layout in QGIS interface

#### 4 ALRR.Synthesis Maps Customization

The map customization refers to the creation of the QGIS project and map for each title word in the atlas. It was already mentioned that the current working version is based on the old Excel file usage. In this case, the user has to prepare an Excel file for each title word, with the structure shown in Fig. 9. Because these files were also used in previous versions, some data are no longer used and some have been newly added. Thus, only the values in column 'A' which contains the survey points codes, column 'D' which contains the form number for each point, and cells 'E1'-'X1' which contain the symbol codes and colors for each form, are used. The text color in the cells that contain the symbol codes will be used to draw the symbols. All the cells used have a colored background in Fig. 9. It must be noticed that the number of symbol codes in the first row depends on the number of forms used and this information is newly introduced.

|     | A   | B       | C       | D     | E        | F         | G         | H         | I         | J        | K         | L         | M        | N        | O        | P        | Q        | R         | S         | T | U | V | W | X |
|-----|-----|---------|---------|-------|----------|-----------|-----------|-----------|-----------|----------|-----------|-----------|----------|----------|----------|----------|----------|-----------|-----------|---|---|---|---|---|
| 1   | Sat | Nord    | Est     | Forma | WNG1_233 | GSIDS_041 | GSIDS_044 | GSIDS_057 | GSIDS_060 | WNG1_117 | GSIDS_091 | GSIDS_093 | WNG1_203 | WNG1_170 | WNG1_045 | WNG1_077 | WNG1_167 | TIMES_042 | TIMES_045 |   |   |   |   |   |
| 102 | 101 | 2111.38 | 3357.27 | 19    |          |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 103 | 102 | 2018.85 | 3478.44 | 8     |          |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 104 | 103 | 1690.59 | 3485.05 | 5     |          |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 105 | 104 | 1582.64 | 3507.08 | 1     | 1        |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 106 | 105 | 1459.27 | 3542.33 | 2     |          | 1         |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 107 | 106 | 1379.96 | 3564.36 | 1     | 1        |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 108 | 107 | 1097.96 | 3590.80 | 2     |          | 1         |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 109 | 108 | 985.60  | 3511.49 | 13    |          |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 110 | 109 | 919.51  | 3511.49 | 19    |          |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |
| 111 | 110 | 1368.94 | 3844.15 | 1     | 1        |           |           |           |           |          |           |           |          |          |          |          |          |           |           |   |   |   |   |   |

Figure 9. Structure of the Excel files for the word “muruiesc”

The map creation is done in several steps. First, the user has to give the required information in the *MapGenerator* application interface: map name, the path of the source Excel file, the path of the QGIS project that contains the map template, and the folder in which the new map is created. Also, the user can choose if the survey points located in the Republic of Moldova and Ukraine are included in the map and if points in which the answer is unknown are displayed or not. Then the generation of the map starts:

- The information from the Excel file is loaded in memory;
- The availability of the required symbols is checked. If at least one symbol is missing, the process is stopped;
- The folder of the new map is created and the files that contain the survey points layer (*000\_MapName.\**) are copied into it and renamed with the actual name of the map;
- The internal database (*.dbf* file) of the survey points layer is updated: The *Symbol* and *SymbolColor* fields are updated using the values from the Excel file. By default, the value of the *Symbol* field is 'TIMES\_045' which corresponds to the '-' symbol. Depending on the choices made in the application interface, for the points in which the answer is unknown and/or for the points located abroad, the value of the *Symbol* field can be set empty to hide them.
- The QGIS project file that contains the map template and layout (*MAP\_000\_MapName.qgs*) is copied and renamed. The sequence "000\_MapName" is replaced by "nnn\_WordTitle" where "nnn" is the question number and "WordTitle" is the transliteration of the word title.
- The newly created project file which is of type xml is updated. It is parsed and the same substitution is made for all the appearances of the "000\_MapName" sequence.

In the final step, the layout has to be updated. As it was already mentioned, the layout contains six elements (map title, map number, map author, and the three legends) which initially have generic content. Excepting the map number and initials of the map's author which contain simple text, the other four elements contain formatted text, including the symbols and phonetic transcription (in the case of the legends). In QGIS, the formatted text can be inserted using HTML files only. Thus, the six files have to be prepared before generating the map. Otherwise, they are replaced with generic HTML files. To update the layout, the *MapGenerator* application parses again the QGIS project file. It identifies the tags that correspond to the element to be updated and insert the content of the HTML files. If these are not available, the user has the possibility to prepare them later and either generate again the map or insert them manually using the QGIS layout manager's interface. Anyway, after the automatic map generation, the result must be reviewed because some repositioning or alignment of the elements may be necessary.

In Figs. 10 and 11, the results of the map generation are presented. In Fig. 10, the Legend\_1 element is selected and the HTML content is visible in the lower-right corner of the window. It can be modified in the QGIS Layout Manager Interface also. The maps generated can be exported as TIF images (Fig. 11) or PDF files with the possibility to choose the page dimensions and resolution.

## 5 Conclusions

In this paper, an information system for data management and publication of *ALRR.Synthesis* is proposed. It is a new contribution to the series of research tools for dialectology designed and implemented in the last 20 years.

For instance, the proposed system is used for the preparation of maps which will be included in the next volume of *ALRR.Synthesis* using the old style of data collection which requires the manual grouping of survey points in which the answer is identical. To automatize this activity a complete re-digitalization of the linguistic material already published in the regional linguistic atlases is required. This requires a huge amount of work because only a small part of them was prepared using computerized methods and the linguistic material is available in electronic format. However,

# GIS-BASED PUBLISHING SYSTEM FOR THE ROMANIAN LINGUISTIC ATLAS BY REGIONS – SYNTHESIS

er, considering that the content of these atlases represents a part of the Romanian cultural heritage, this action should be undertaken.

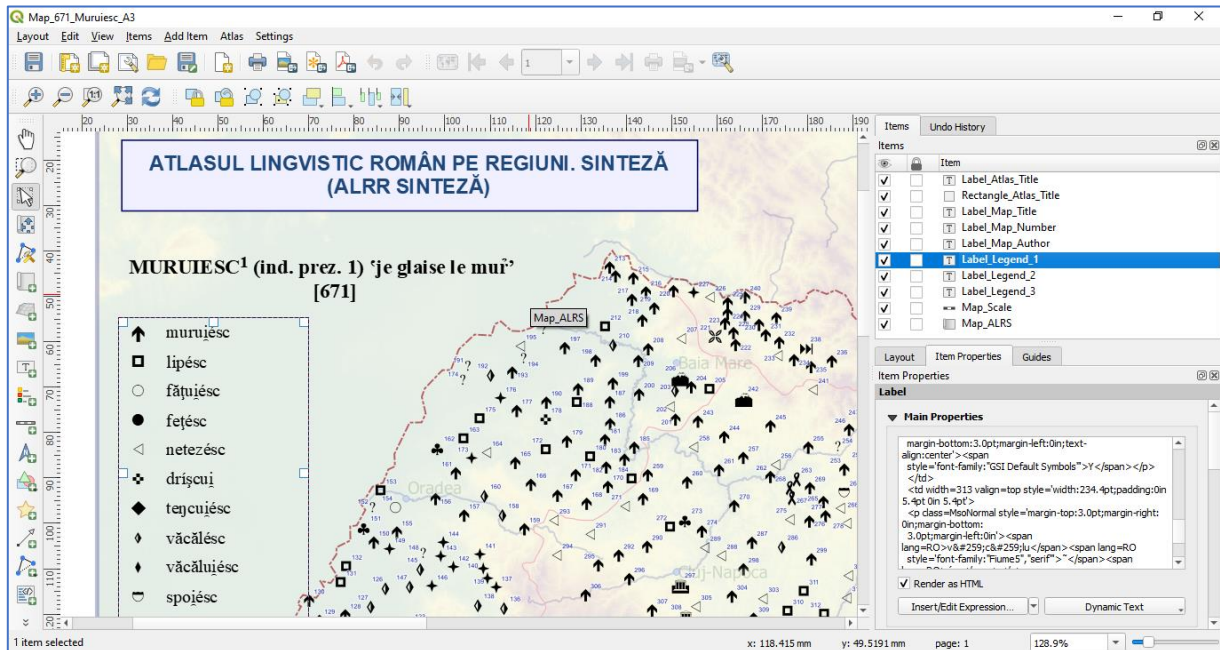


Figure 10. Generated map shown in the layout manager interface (digitalized by M. Morcov; will be included in the next volume of ALRR.Synthesis)

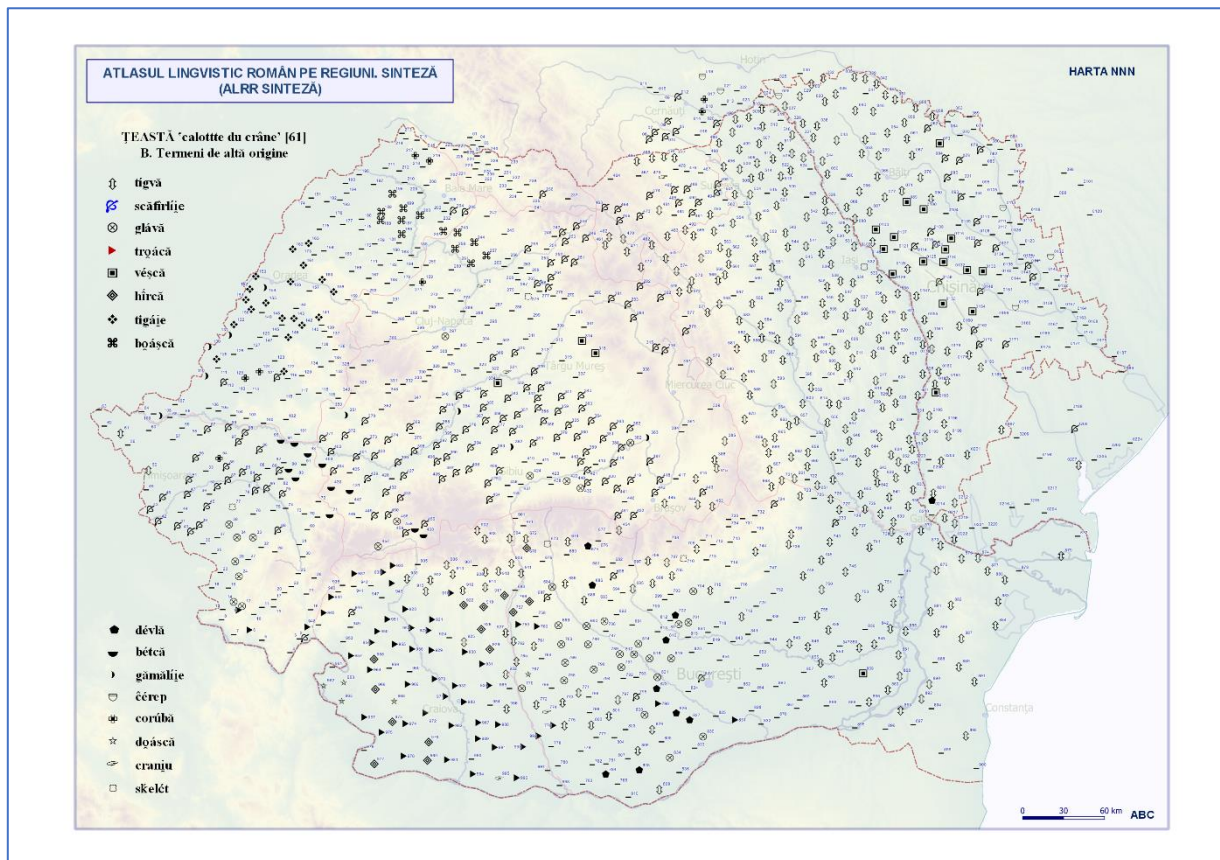


Figure 11. The final generated map (map from ALRR.Synthesis vol. I, re-digitalized)

## Acknowledgements

The work presented in this paper was partially supported by the research grant from the Romanian National Authority for Scientific Research and Innovation CNCS = UEFISCDI, project no PN-III-P4-ID-PCE-2020-0451.

## References

- Silviu-Ioan Bejinariu, Vasile Apopei, Manuela Nevaci, and Nicolae Saramandu. (2019). Information Management Framework for the European Linguistic Atlas. In *Atlas Linguarum Europae (ALE) Symposium, The 52nd Annual Meeting*, Vilnius, Lithuania, 8-12 September 2019.
- Silviu-Ioan Bejinariu, Vasile Apopei, Manuela Nevaci, and Nicolae Saramandu. (2018). New GIS-based Approaches for the Linguistic Atlases. In V. Păiș, D. Gifu, D. Trandabă, D. Cristea, D. Tufiş (eds.), *Proceedings of the 13th International Conference "Linguistic Resources and Tools for Processing Romanian Language"*, Iași, 22 November 2018, Iasi, Romania, pages 147-158.
- Silviu-Ioan Bejinariu, Vasile Apopei, and Florin Iftene. (2021). Romanian geolinguistics in the digital age (in Romanian: Geolingvistica românească în era digitală). In: Manuela Nevaci, Irina Floarea, Ioan-Mircea Farcaș (eds.), *Ex Oriente lux. In honorem Nicolae Saramandu*, Alessandria, Edizioni dell'Orso, «La colonna infinita», 14: 163-176.
- Florin-Teodor Olariu and Veronica Olariu. (2014). The Romanian linguistic cartography in the digitizing era: the electronic atlases. In *Dialectologia et Geolinguistic*, 22(1): 75-90.
- Francisco Răzvan Constantin. (2021). Motivational linguistic areas based on the Romanian Linguistic Atlas by Regions. Synthesis (in Romanian: Aree lingvistice motivaționale pe baza Atlasului lingvistic Român pe Regiuni. Sinteză). In: Manuela Nevaci, Irina Floarea, Ioan-Mircea Farcaș (eds.), *Ex Oriente lux. In honorem Nicolae Saramandu*, Alessandria, Edizioni dell'Orso, «La colonna infinita», 14: 361-376.
- Vasile Pavel. (1993). The Romanian Linguistic Atlas, by region. Bessarabia, Northern Bukovina, Transnistria. Reconsiderations, tasks, ways of mapping (in Romanian: Atlasul lingvistic român, pe regiuni. Basarabia, Bucovina de nord, Transnistria. Reconsiderări, sarcini, modalități de cartografiere). In *Al VI-lea Simpozion Național de Dialectologie*, Cluj-Napoca, 12-14 octombrie 1990, pages 159-167.
- Nicolae Saramandu. (1990). For an atlas of regional linguistic atlases (in Romanian: Pentru un atlas al atlaselor lingvistice regionale). In *Limba română*, XXXIX(1): 57-67.
- Nicolae Saramandu and Manuela Nevaci. (2013). *Syntheses of Romanian dialectology* (in Romanian: *Sinteze de dialectologie română*). Editura Universitară, București. 38-65.
- Nicolae Saramandu and Manuela Nevaci. (2018). Linguistic atlases (in Romanian: Atlase lingvistice). In Marius Sala, Nicolae Saramandu (coords), *Lingvistica românească*, Editura Academiei Române, pp. 451-458.

## Romanian Linguistic Atlases

- (ALE) Nicolae Saramandu, Manuela Nevaci and Ionuț Geană (coords). (2015). *Atlas Linguarum Europae – ALE*. Volume I. Neuvième fascicule. Cartes Linguistiques Européennes. București, Editura Universității din București. <https://lingv.ro/atlas-linguarum-europae/>. Last accessed at 8.09.2022.
- (ALRR.Sinteza.I) Nicolae Saramandu (coord), Verginica Barbu Mititelu, Irina Floarea, Mihaela Mariana Morcov, Carmen-Ioana Radu, Marilena Tiugan, Dana Mihaela Zamfir. (2007). *Atlasul lingvistic român pe regiuni. Sinteză*. vol I. Editura Academiei Române, București.
- (ALRR.Sinteza.III) Nicolae Saramandu (coord), Mihaela Morcov, Manuela Nevaci, Daniela Răuțu, Irina Floarea, Carmen-Ioana Radu, Ionuț Geană and Mara Iuliana Manta. (2018). *Atlasul lingvistic român pe regiuni. Sinteză*. vol. III, Editura Academiei Române, <http://www.lingv.ro/images/ALRR%20sinteza.pdf>. Last accessed at 8.09.2022.
- (WLAD) *Linguistischer Atlas des dacorumänischen Sprachgebietes / herausgegeben auf kosten der rumänischen Akademie in Bukarest* von Prof. Dr. Gustav Weigand - Leipzig: Johann Ambrosius Barth. (1909), available with Public Domain License at [https://www.europeana.eu/en/item/9200456/handle\\_123456789\\_54079](https://www.europeana.eu/en/item/9200456/handle_123456789_54079). Last accessed at 8.09.2022.

# ParemCor – CONTEMPORARY ROMANIAN LANGUAGE CORPUS OF PROVERBS AND ANTI-PROVERBS

ANCA-DIANA BIBIRI<sup>1</sup>, MARIA MORUZ<sup>1</sup>, AND MIHAI ALEX MORUZ<sup>2</sup>

<sup>1</sup>*Department of Social Sciences and Humanities, Institute of Interdisciplinary Research, “Alexandru Ioan Cuza” University of Iași, Romania*

<sup>2</sup>*Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași, Romania*

## Abstract

Coming from the multi-millennial wisdom of people, usually, a proverb is defined as a short sentence that contains, in a standard form, and with a metaphorical connotation, a universally held truth. Starting from this definition, particularly in later decades, special care has been given to intentional creative modifications of proverbs, with the purpose of fulfilling certain stylistic functions. Starting from this observation, the present paper will describe the ParemCor corpus, which has been derived from CoRoLa (Corpus of Contemporary Romanian Language), and which contains annotated proverbs, together with the creative changes performed on them. The proverbs and anti-proverbs have been automatically extracted and then manually validated and richly annotated. The results we have obtained show that proverbs are still used in language, particularly in journalistic and literary texts; we have also shown that proverbs frequently undergo transformations, for stylistic or rhetorical reasons, as means of attracting attention and persuading the reader.

**Keywords:** proverbs, anti-proverbs, CoRoLa, Romanian, ParemCor, annotated corpus, repeated discourse.

## 1 Introduction

Proverbs, masterpieces of thought, represent popular moral teaching born from human experience and having a prebuilt sense, fixed by convention for all speakers, and thus a part of the common linguistic background (Kleiber, 2000, p. 40).

W. Mieder defines the proverb as “a short, generally known sentence of the folk which contains wisdom, truth, morals, and traditional views in a metaphorical, fixed and memorable form and which is handed down from generation to generation.” (Mieder, 1993, p. 24).

According to Hrisztova-Gotthardt & Varga (2015, p. 1), “proverbs summarize everyday experiences and common observations in a concise and figurative way. They have been created and used for thousands of years and passed as expressions of wisdom and truth from generation to generation.”

Kleiber approaches the semantic perspective of proverbs and considers that his “semantic structure always conveys an implication which relies upon two situations involving humans” (2000, p. 39); while Steyer analyses proverbs from a structural point of view, and considers them to be “equivalent to a sentence” along with functional features: “have become part of everyday language as an expression of worldly wisdom” and “express generalized experiences or value judgements” (2015, pp. 209-210).

Dumistrăcel (2006) considers that proverbs belong to the gnomic aspect and are truths extracted on the basis of observation of the environment and life experience, and consequently are assertions on reality. He follows the opinion of Eugeniu Coșeriu, according to whom proverbs are, alongside idioms, common expressions, traditional comparisons, clichés, sayings, etc., belonging to repeated discourse, as opposed to free speech technique. By repeated discourse, we designate

everything which, in the utterances of the members of a community, is repeated more or less the same as in previous discourses (Coseriu, 2000, p. 258).

Since in language these sentences are usually widely known, it has become commonplace to refer to them in shorthand (that is, to say only part of the proverb) or to modify them for some kind of rhetorical or psychological effect. According to Groupe  $\mu$  – (Dubois et al., 1974, p. 37), “all rhetorical operations are based on a fundamental property of linear discourse – that it can be decomposed in smaller and smaller units”; because of this, our research focused on the units of the proverbs which undergo changes as they are transformed into anti-proverbs. Upon study, it has been discovered that most of these changes fit into the categories proposed by Quintilian (cf. Dumistrăcel, 2006, pp. 46-47; Quintilianus, 1974).

Dating back to ancestral times, like proverbs themselves, the term *Antispruchwort* (anti-proverb, perverb) was coined by Wolfgang Mieder in 1982 and represents “parodied, twisted, or fractured proverbs that reveal humorous or satirical speech play with traditional proverbial wisdom” (Mieder, 2004, p. 28); or is “an allusive distortion, parody, misapplication, or unexpected contextualization of a recognized proverb, usually for comic or satiric effect” (Doyle et al., 2012, p. XI) (e.g. Great minds **think** alike. → Great minds **drink** alike.; **Laughter** is the best medicine. → **Slaughter** is the best medicine).

In this chapter, the first section presents considerations about proverbs and anti-proverbs, followed by a second part describing the theoretical framework, the extraction process from the CoRoLa corpus, as well as the outcome of their analysis, followed by some conclusions.

## 2 On Proverbs and Anti-proverbs

The essence of the proverb (and also the anti-proverb) is very well captured by the famous Romanian writer Lucian Blaga who underlines the idea that in the center of attention is a man with all his qualities and shortfalls, with all his mannerisms and meaning, both wanted and unwanted:

Proverbs are the aphorism of the people. The song and the proverb have a difficult what to define: a naturalness that softens the heart and the most incorruptible intelligence, a grace of chance, something above good and evil when it comes to a song, something about the truth and untruth when it comes about the proverb...We have proverbs that are floggings with fire and proverbs that, before turning into words, were flowers. Some discreet, opening metaphysical horizons. Others smile from a tragic resignation to life. Some are the ugly habit of moralizing teachers. Others – life-saving humor of sadness. Depth, play, grotesque, you meet at every step... proverbs are both fragments of philosophical systems, fragments of psychology, pieces of a large pamphlet<sup>1</sup>. (Blaga, 2003, p. 311).

Literature of anti-proverbs is very rich and it is reflected in various valuable works dedicated to many languages, gaining ground in the science of paremiology. The first collection is represented by Mieder, German collection of anti-proverbs (Mieder, W. (1982a) *Antispruchwörter*. Band I, Wiesbaden: Verlag für Deutsche Sprache.); for English, the collection is represented by the book *Twisted Wisdom: Modern Anti-Proverbs*, by Wolfgang Mieder and Anna Tothne Litovkina in 1999. From now on – a flourishing domain was born, the transformation of anti-proverbs becoming more and more popular due to their spreading on the Internet and in mass media. Litovkina discusses anti-proverbs extracted from English and American written resources; she underlines the fact that anti-proverbs are, like proverbs themselves, both the most pervasive and the most elusive of expressions and their reinterpretation in innovative ways can create humor:

“We laugh at some anti-proverbs because they skew our expectations about traditional values, order, and rules. We are, however, sometimes struck by the absurdity of some situ-

<sup>1</sup> Original quote in Romanian, translated by authors

ations portrayed in proverb parodies, especially when they rely purely upon linguistic tricks employed for the sole purpose of making punning possible.” (Litovkina, 2015, p. 344).

Anna Konstantinova shows how in modern mass media the proverb, and also its modifications, represent an important way of choice for conveying different attitudes and views:

“Such creative transformations broaden the linguistic boundaries of a given culture, demonstrate novel ways of thinking, and, more importantly, introduce new directions for action. Proverbs can be adapted to any socio-cultural context, and, therefore, frequently modified, immediately come in view of broad audiences with the help of any of the existent mass” (Konstantinova, 2015, p. 281).

Romanian literature concerning the study of proverbs and anti-proverbs is flourishing. Many scholars have turned their attention to the study of these folk jewels. One well-known scholar who researched repeated discourse as opposed to the free speech technique is Stelian Dumistrăcel. He examines many examples of RDU (Repeated Discourse Utterances) according to the four Quintilian “figures of constructions” (*adiectio*, *detractio*, *immutatio*, and *transmutatio*) in two reference works: *The Romanian Journalistic Language in Terms of Functional Styles* (2007) and *Repeated Discourse of the Journalistic Text. The Temptation of Settling Phatic Communion through the Media* (2006).

Based on Speech Act Theory and Relevance Theory, Ioan Milică analyses some Romanian proverbs and anti-proverbs and considers that the transformation of a proverb into an anti-proverb is the manifestation of a communicative intention that is revealed in context and gives authority to the discursive construction it summarizes (Milićă, 2013, pp. 46-48). Cristinel Munteanu explores the RDU from different perspectives, giving examples of atypical journalistic headlines based on their sound similarity or based on the resemblance of graphic nature (Munteanu, 2013, p. 11).

Monica Bilaucă makes an inventory of Romanian neologisms in repeated discourse occurring in different lexicographic works and mass media, reflecting the variation of lexical mutation (of language in use). She chose the journalistic style because it presents “the most faithful the dynamics of language in function” (2015, p. 7) and reflects, as observed by Dumistrăcel, “the most obvious valences for high-performing public expression” (2006, pp. 128-129). In the second part of her book, the author reviews the variations of neologisms in public discourse, both for canonical phrases as well as for their decomposition and restructuring. The Index of Modified Expressions at the end of the book (pp. 197-202) is the starting point for our choice of phraseological units. Other proverbs were extracted empirically, by means of manual corpus analysis and discourse markers, explained in the next section.

Proverbs are considered universal truths, and, as such, transcend time and cultures relatively unchanged; however, structural variations have been sometimes observed. These changes can be unintentional, due to insufficient knowledge of the language, or intentional, such as anti-proverbs, which come about by changing well-known proverbs either for irony or to emphasize an idea.

The purpose of our research is to study these stylistically relevant intentional changes. The Corpus of Contemporary Romanian Language (CoRoLa) contains many such modifications, especially since a large portion of the corpus is made up of literary and journalistic texts. The changes which lead to anti-proverbs are found at the confluence of repeated discourse and free speech technique – one part is stable and belongs to repeated discourse (in this case, the stable part of the proverb) and the other part is changed by addition, removal, switching around or replacing elements; these types of modifications correspond to the rhetorical categories described in Quintilian’s *Quadripartita ratio*: *adiectio*, *immutatio*, *permutatio*, *detractio* (cf. Dumistrăcel, 2007). Journalistic discourse is a thriving field for repeated discourse and is considered by Dumistrăcel as an expression of the primary universals of language represented by creativity and

change. Thus, journalistic texts are governed by the four constructions figures mentioned above and further discussed in the following section.

### 3 Paremcor: Anti-Proverb Corpus Extracted from CoRoLa

#### 3.1 The CoRoLa Corpus

CoRoLa (Corpus of Contemporary Romanian Language) is a resource resulting from a priority project of the Romanian Academy (through its Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy Bucharest (RACAI) and the “Dan Gâlea” Institute for Computer Science Romanian Academy – Iași Branch (IIT)). CoRoLa represents the cultivated Romanian language designed to include both written and spoken forms; mirroring the cultivated Romanian language, large enough to include in significant contexts the greatest part of the Romanian vocabulary, with more than 500 million tokens and more than 300 hours of transcribed speech (Barbu-Mititelu et al., 2018). The textual collection is made up of publications covering the period from the 2nd World War to our days, while the spoken collection includes only recent recordings.

The texts in CoRoLa cover a large variety of domains and subdomains, distributed, in an unbalanced way, in several language styles (legal, administrative, scientific, journalistic, imaginative, memoirs, blog posts) and the included domains are represented by: arts and culture, society, science, nature, law, journalistic, and others (Bibiri et al., 2015; Gîfu et al., 2019).

The richness of literature and journalistic style in CoRoLa offers the opportunity to extract and create a corpus of the most used proverbs and anti-proverbs in our language for their colorful and ironic style meant to capture attention, stand out, and highlight the ideas.

#### 3.2 Anti-proverbs Extraction from the Corpus

Due to the great lexical and syntactic variation, the process of automatic identification of proverbs in texts is difficult and represents a challenge for NLP (Natural Language Processing) (Reis & Baptista, 2016). In this sense, Steyer also considers that the process of searching for proverbs in a language corpus is not a trivial matter (2015, p. 210).

As such, each proverb has a set of annotations, pertaining to the following information: proverb nucleus, annotated by lemma (in order to facilitate filtering, ordering, and searching of the corpus), source proverb(s) (sometimes, a modified proverb is derived from more than one source), the encountered modification(s) (based on the rhetorical structures proposed by Quintilian), the stable part of the proverb (the word or sequence which remains unchanged).

In order to extract occurrences of base or modified proverbs, various interrogations of CoRoLa must be performed. As the stable part usually refers to a part of speech or a lemma of a word, we had to create specific interrogations for each type of proverb or derivative, taking into account such things as word position in the construct, possible inter-posed words, fixed or limited morphological variability (i.e., specific verbal times or diathesis), etc. All these types of queries were possible as CoRoLa is annotated at the lemma and part of speech level for each word, and uses a complex query system based on the Lucene query engine. Queries were performed on the full corpus, with no special regard towards language styles or other similar considerations.

The starting point for the interrogation process was a set of proverbs on the basis of which we built a set of queries in the CoRoLa query language. We have extracted these base proverbs from literature, namely from (Bilauca, 2015), as these show a higher propensity for forming anti-proverbs. The queries were based on the stable part of the proverbs taken into consideration, and allowed for syntactic, morphologic, and lexical variability, similar to the procedure described in (Reis & Baptista, 2016); the key differences with regards to their approach come from our not using an automaton to generate variations on the proverbs, from the fact that we allow for optional interposed words (denoted in queries by the group “[?]”) and that we allow for morphological

descriptions in place of exact words. The queries were written in the Poliqarp query language (Janus & Przepiórkowski, 2007), used in the CoRoLa corpus interface (Kupietz, 2010), and manually validated to improve the quality of the retrieved results; two examples are given below (translations for the queries are given in section 4.3).

```
[drukola/orth="toamna"] []? []? []? []? []? [druko-  
la/orth="se"] [drukola/orth="numără"]  
[drukola/m="msd:pw.*"] []? scoală []? []? []? []? [druko-  
la/m="msd:vmip.*"]
```

The queries reflect the nature of the changes we have observed in the corpus with regards to proverb use: we take into account inserted or changed words, using the optional word operator ([]?), the core of the proverb is searched for by using exact word match ([drukola/orth="toamna"]), and, for common or expected changes, we search for specific morphological descriptions ([drukola/m="msd:vmip.\*"] – *msd* refers to the morphological tag, and the *vmip* code means *verb main indicative present tense*). The second example query given above can be interpreted as follows: the first word must be a relative pronoun, followed by an optional word, then the exact form ‘scoală’ (i.e., *rise* or *wake up*), followed by up to four other words and then, in the end, an action verb in the present tense. For a more in-depth description of the Poliqarp query language, see the CoRoLa homepage<sup>2</sup>.

We have empirically determined that, with regard to proverbs transforming into anti-proverbs, there exists a stable part for each base proverb, which remains unchanged. For example, in the case of the proverb “Cine se scoală de dimineață departe ajunge” (literally Whoever rises in the morning travels far, En. ‘The early bird gets the worm’), the stable part is made up of the sequence “se scoală de dimineață”. To these, we have added some morphological patterns, such as ‘relative pronoun + verb in present tense + verb in present tense’, taken into account the possibility of interposed words, and gave more consideration to sequences preceded by certain discourse markers (e.g. *vorba (a)ceea*, ‘as the saying goes’). Following manual annotation, we have identified more possible modifications of proverbs, and thus expanded, changed, or added queries.

It is important to note that fully automatic recognition of proverbs, both changed and in their original form, is currently not possible over the available annotation level for CoRoLa. It is often the case that the morphological and syntactic structure of free discourse is the same as that of repeated discourse (e.g., relative pronoun followed by a verb in the present tense). As such, the automatic recognition of repeated discourse could only be feasible following a rigorous semantic annotation of the base corpus.

### 3.3 Storage of the Corpus

The list of proverbs considered (Bilauca, 2015) contains approximately 100 proverbs, all of which have been processed into queries. The results of the queries have been saved in an XML format (see Fig. 1), and then manually validated by human experts. The expert validation was necessary as the results contained only a small proportion of repeated discourse instances, because of the aforementioned similarity to regular discourse. One example of a sequence that was automatically extracted but not an instance of repeated discourse is “*un ins care se scoală dimineața, citește Scânteia*” (literally a person who wakes up in the morning, reads *Scânteia* – a communist newspaper). The useful results were then further annotated with the following information (an example XML file in the corpus can be seen in Fig. 2):

- proverb (element) – the exact span of the repeated discourse instance was marked;
- stable (element) – the stable part of the proverb;

<sup>2</sup> <https://corola.racai.ro>

- adiectio, immutatio, transmutatio, and detractio (elements) – the modification to the original with regards to the processes described above;
- modification[1|2|3] (attribute of proverb) – Describes the modification(s), as an anti-proverb can undergo multiple stylistic changes;
- obs (attribute of proverb) – observations regarding the annotated text.

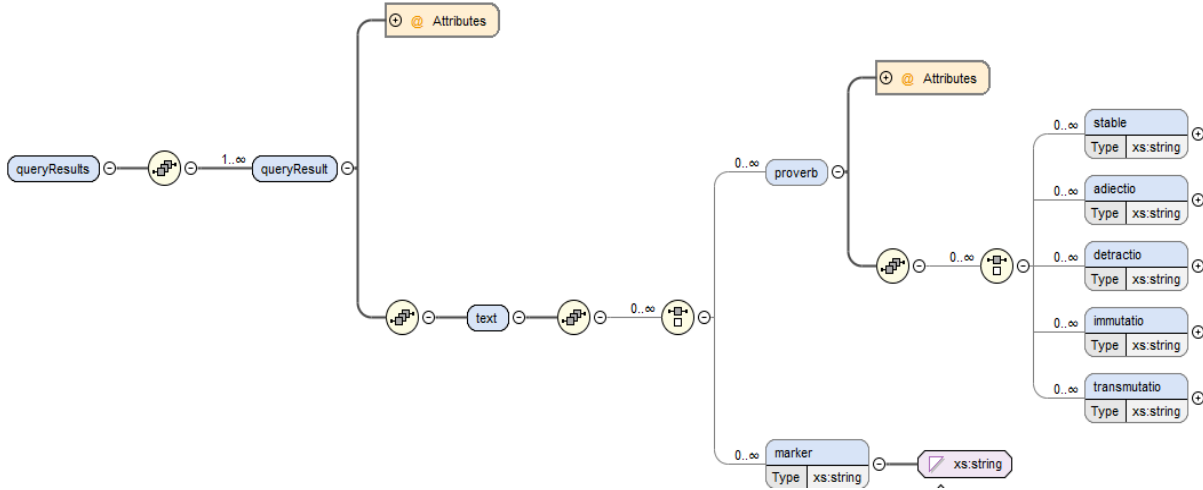


Figure 1. Structure of the XML file

```

<queryResult base="[Toamna se numără] bobocii" sigle="Băsescu, ocnașul de la Roșia Montana (2011) [Corola-journal/Journalistic/25151_a_26476]">
  <text>
    chiar Guvernul, care ar urma să pice de pe țocul minuscul al încrederii pe care o mai au românii în el. De văzut, dar s-ar putea ca Lăzăroiu să zică mersi că a scăpat primul, așa, ca la vaccin, pentru că
  </text>
  <proverb core="toamnă" modification1="immutatio">
    <stable>toamna se numără</stable>
    <immutatio>miniștrii</immutatio>
  </proverb>
  , iar pentru PDL urmează vremuri extrem de tulburi.
</text>
</queryResult>
<queryResult base="[Toamna se numără] bobocii" sigle="Înaltele Toamne by Barbu Cioculescu (2007) [Corola-journal/Journalistic/9092_a_10417]">
  <text>
    ocrotită țin minte descărcarea lemnelor de foc - singura traducere în versuri pe care am încercat-o vreodată a fost aceea din Baudelaire: Bientôt nous plongerons dans les froides tenebres..." cu sunetul butucilor căzând pe caldarâm. Pentru că mai exista o toamnă, toamna gospodărească, sezonul
    <marker>corespunzând zicalei</marker>
    "
  </text>
  <proverb core="toamnă" modification1="-" obs="interesantă introducerea proverbului în context">
    <stable>toamna se numără</stable>
    bobocii
  </proverb>
  ", toamna pregătirilor pentru apropiatul asediu al iernii. Cu acesta nu era de glumit într-o lume care se troienea cu lunile, la cheremul săniilor, al viscoleurilor, iubită de cei puternici, lunecători pe schiuri și suflători pe nări de aburi
  </text>
</queryResult>

```

Figure 2. XML encoding of the anti-proverb 'Toamna se numără miniștrii' ('Count your ministers in the autumn')

### 3.4 Examples of Anti-proverbs

The following examples, extracted from our corpus, refer to each of the four categories described by Quintilian – *quadripartita ratio*. The examples are given in Romanian, followed by a word-

for-word translation to English and then the proper English translation. The examples given in bold font are the canonical forms of proverbs in Romanian.

1. **adiectio (addition)** – the modification of the canonical form of a proverb by means of lexical accumulation, with a stylistic intent.

(1) **Minciuna are picioare scurte.**

A lie has short legs.

'A lie has a short life.'

(2) Minciuna *prezidențială* are picioare scurte.

A *presidential* lie has short legs.

(3) Minciuna are picioare scurte *și întotdeauna o ajunge din urmă adevărul.*

A lie has short legs *and the truth always catches up.*

(4) Minciuna are picioare scurte... *Dar sprintene!*

A lie has short legs... *But they are quick!*

(5) Minciuna are *totdeauna* picioare scurte *și multe ițe, în care te poți încurca.*

A lie *always* has short legs *and many threads that can tangle you up.*

(6) **Sângele apă nu se face.**

Blood does not turn into water.

'Blood is thicker than water.'

(7) Sângele *Ninei Cassian* apă nu se face.

The blood of *Nina Casian* does not turn into water.

(8) Sângele *textualist* apă nu se face.

*Textualist* blood does not turn into water.

2. **immutatio (substitution)** – the substitution of one or more lexical elements in the canonical form of a proverb with new words.

(9) *Populismul* are picioare scurte.

*Populism* has short legs. (The base form is given in (1))

(10) **Toamna se numără bobocii.**

Count your goslings in the autumn.

'Don't count your chickens before they are hatched.'

(11) Toamna se numără *miniștrii*.

Count your *ministers* in the autumn.

(12) **Unde dai și unde crapă.**

Where you strike and where it breaks. (Literally explained as to perform an action and get a completely unexpected result).

(13) Unde dai și *cine* crapă.

Where you strike and *who* breaks. (Perform an action and wait to see who reacted to it, usually in a negative way).

3. **transmutatio (transposition)** – the permutation of words in the canonical form of proverbs, or of sequences of words in the case of changes obtained by combining parts of two canonical proverbs.

(14) **Ai carte, ai parte.**

You have documents, you have rights. (archaic); You have knowledge, you have advantages. (modern)  
 'Knowledge is power.'  
 (15) *Ai parte, ai carte.*  
*Power is knowledge.*

4. **detractio (omission)** – the process of removing part of the canonical form of a proverb while keeping the bare minimum of relevant context (referred to by us as the core), and thus engaging the imagination of the reader (Dumistrăcel 2007).

- (16) **Cine are urechi de auzit** (*să audă* is missing)  
 Whoever has ears to hear (*should listen* is missing)  
 Whoever has ears (, let them hear)  
 (17) **Cine se scoală de dimineață** (*departe ajunge* is missing)  
 Whoever rises in the morning (*travels far* is missing)  
 'The early bird' (*gets the worm*).

In some cases, antiproverbs are obtained by means of two or more of the above-mentioned methods, as shown in examples (18) and (19).

(18) **Cine sapă groapa altuia departe ajunge** (combination of two proverbs, half of each is missing and is substituted with half of the other: *Cine sapă groapa altuia* cade singur în ea.- *Whoever digs another's hole* falls in it himself, and *Cine se scoală de dimineață departe ajunge* - *Whoever rises in the morning travels far*. This is also a case of Transmutatio; see below)

Whoever digs another's hole travels far.

(19) **Cine fură azi un ou la nevoie se cunoaște.** (*Cine fură azi un ou mâine va fura un bou.*- *Whoever steals an egg today will steal an ox tomorrow*, and *Prietenul la nevoie se cunoaște.* - *A friend in need is a friend indeed*.)

Whoever steals an egg today is a friend indeed.

### 3.5 Statistics

The corpus is built on approximately 100 base proverbs extracted from (Bilauca, 2015), and also on several markers known to precede proverbs in discourse. Some of these markers include constructions such as 'vorba (a)ceea' (as the saying goes), 'se spune' (it is said), 'vorba proverbului' (as the proverb goes).

From these, queries were created and run on CoRoLa and the results were analyzed. If the annotation yielded changes we had not expected, further queries were performed, and the results were added to the corpus.

Because of time constraints, we have only managed to manually validate and annotate 16 of the analyzed proverbs, and, for this first attempt, we have been unable to have more than one expert analyze each set of results so cross-validation was not possible. The 16 sets yielded approximately 900 candidate sequences, from which we have found 430 proverbs and modifications; the distribution of results is given in Fig. 3. Since the manual annotation was carried out by experts (two of the authors of the current paper), there was no need for training and no requirement for inter-annotator agreement; the annotation was carried out using the Oxygen XML editor<sup>3</sup> and the validation schema described in Fig. 1, thus guaranteeing consistency of annotation.

<sup>3</sup> <https://www.oxygenxml.com/>

Of the four rhetorical procedures, the first two, *adiectio* and *immutatio*, are the most common, while *transmutatio* and *detractio* suppose a more significant effort from the part of the writer and also from the part of the reader with regards to the intent of the discourse.

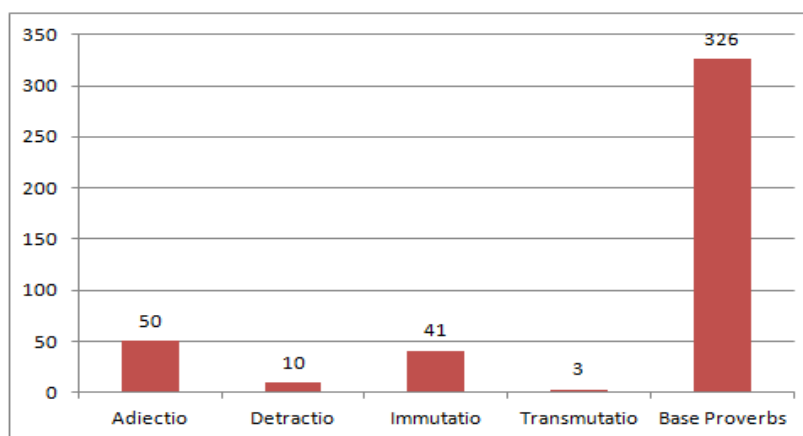


Figure 3. Composition of the Corpus

## 4 Conclusions

In this paper, we have described the ParemCor, a corpus for proverbs and anti-proverbs in Romanian, extracted from CoRoLa. The proverbs and anti-proverbs have been automatically extracted and then manually validated and richly annotated. The results we have obtained show that proverbs are still used in language, particularly in journalistic and literary texts; we have also shown that proverbs frequently undergo transformations, for stylistic or rhetorical reasons, as means of attracting attention and persuading the reader. Even though the absolute number of proverbs found is small (on average, about 20 proverbs and anti-proverbs for each of the 16 base versions we have investigated and annotated, in a corpus of hundreds of millions of words), “frequency is not all: the most interesting discourse phenomena need not be particularly frequent and yet interlocutors may well be aware of them and use them as heuristics when they co-construct, adapt to and interpret discourse in context.” (Lüdeling & Kytö, 2008, p. 1043).

The abundance of ancestral wisdom dictums, especially their specific feature of creation, testify the immense potential for describing, interpreting the world, and helping express different attitudes and values. For further research, we aim to increase the size of the corpus, by analyzing the rest of the results for the remaining base forms (we have more than 80 left to process and annotate); the corpus is not yet available online, but we aim to make it publicly accessible once we have finished the analysis.

## References

- Bański, P., Frick, E., Hanl, M., Kupietz, M., Schnober, C. & A. Witt. (2013). Robust corpus architecture: a new look at virtual collections and data access. In Hardie, A. & Love R. (Eds.), *Corpus Linguistics* (pp. 23–25). Abstract Book. Lancaster, UCREL.
- Barbu-Mititelu, V., Tufiş, D. & Irimia, E. (2018). The Reference Corpus of the Contemporary Romanian Language (CoRoLa). In *Proceedings of LREC 2018*, (pp. 1178-1185). Japan.
- Bibiri, A. D., Bolea, S. C., Scutelnicu, L. A., Moruz, M. A., Pistol, L. & Cristea, D. (2015). Metadata of a Huge Corpus of Contemporary Romanian Data and Organization of the Work. In: *Proceedings of the 7th Balkan Conference on Informatics Conference*, (pp. 351-358). Craiova, Romania, September 2-4, 2015.
- Blaga, L. (2003). *Zări şi fapte* [Sees and Facts], (cap. Studiul proverbului [Study of the proverbs] (pp. 310-312). Editura Humanitas, Bucureşti.
- Coseriu, E. (2000). *Lecţii de lingvistică generală* [Lessons of General Linguistics]. Editura Arc. Chişinău.

- Doyle, C. C., Mieder, W., Shapiro, F. (2012). *The Dictionary of Modern proverbs*. New Haven: Yale University Press.
- Dumistrăcel, S. (2007). *Limbajul publicistic românesc din perspectiva stilurilor funcționale* [The Romanian Journalistic Language in Terms of Functional Styles]. Institutul European, Iași.
- Dumistrăcel, S. (2006). *Discursul repetat în textul jurnalistic. Tentația instituirii comuniunii fatice prin mass-media* [Repeated Discourse of the Journalistic Text. The Temptation of Settling Phatic Communion through the Media]. Editura Universității „Alexandru Ioan Cuza” din Iași.
- Gifu, D., Moruz, M. A., Bolea, C., Bibiri, A.-D. & Mitrofan, M. (2019). The Methodology of Building CoRoLa. *Revue roumaine de linguistique*, LXIV, 3, 241-253.
- Grupul μ: Dubois, J., Edeline, F., Klinkenberg, J. M., Minguet, P., Pire, F., & Trignon, G. (1974). *Retorică generală* [General Rhetoric], Introducere de Silvian Iosifescu, Traducere și note de Antonia Constantinescu și Ileana Littera. Editura Univers. București.
- Hrisztova-Gotthardt, H. & Varga, M. A. (Eds.), (2015). *Introduction to Paremiology: A Comprehensive Guide to Proverb Studies*, De Gruyter.
- Janus, D., & Przepiórkowski, A. (2007). Poliqarp: an open source corpus indexer and search engine with syntactic extensions. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, (pp. 85–88). Association for Computational Linguistics.
- Konstantinova, A. (2015). Proverbs in Mass Media. In: Hrisztova-Gotthardt, H., & Varga M. A. (Eds.), *Introduction to Paremiology: A Comprehensive Guide to Proverb Studies* (pp. 276-293). De Gruyter.
- Kupietz M., Belica, C., Keibel, H. & Witt, A. (2010). The German Reference Corpus DeReKo: A primordial sample for linguistic research. In: Calzolari, N. et al. (Eds.), *Proceedings of LREC 2010*, (pp. 1848-1854).
- Litovkina, A. T. (2015). Anti-proverbs. In: Hrisztova-Gotthardt, H., & Varga, M. A. (Eds.), *Introduction to Paremiology: A Comprehensive Guide to Proverb Studies* (pp. 326-352). De Gruyter.
- Lüdeling, A. & Kytö, M. (2008 & 2009). *Corpus Linguistics. An International Handbook*, Walter de Gruyter GmbH. Berlin, vol. 1, 2008, vol. 2, 2009.
- Justyna, M. (2016). A Proverb a Day Keeps Boredom Away. *New Horizon in English Studies*, 1, 21-30. DOI: 10.17951/nh.2016.1.21
- Kleiber, G. (2000). *Sur le sens des proverbes*. [On the Meaning of Proverbs]. *Langages*, 34 (139), 39-58.
- Mieder, W. (1993). *International Proverb Scholarship: An Annotated Bibliography, with supplements*. Garland Publishing. New York.
- Mieder, W. (1982a). *German collection of anti-proverbs Antispruchwörter. Band I*, Wiesbaden: Verlag für Deutsche Sprache); for English, the collection is represented by the book *Twisted Wisdom: Modern Anti-Proverbs*, by Wolfgang Mieder and Anna Tothne Litovkina in 1999.
- Mieder, W. (2004). *Proverbs: A Handbook*. Greenwood Folklore Handbook. Greenwood Press.
- Milică, I. (2013). *Proverbe și antiproverbe* [Proverbs and anti-proverbs]. In Saramandru, N., Nevaci, M. & Radu, C.-I., *Convergențe lingvistice* [Linguistic Convergences], Editura Universității din București.
- Munteanu, C., (2013). *Discursul repetat și titlurile jurnalistice atipice* [Repeated Discourse and Atypical Journalistic Headlines]. *Limba Română* [Romanian Language], Chișinău, XXIII, 9-12, 11-19.
- Quintilianus, M. Fabius (1974). *Arta oratorică* [Oratory], vol. 1, Traducere, studiu introductiv, tabel cronologic, note, indici de Maria Hetco, Editura Minerva, Biblioteca pentru toți, București.
- Reis, S., & Baptista, J. (2016). *Let's play with proverbs? NLP tools and resources for iCALL applications around proverbs for PFL*. <https://doi.org/10.13140/RG.2.1.1244.7603>
- Reis, S., & Baptista, J. (2017). Estimating Lexical Availability of European Portuguese Proverbs. In Mitkov, R. (Ed.), *Computational Corpus-Based Phraseology. EUROPHRAS 2017*. Lecture Notes in Computer Science, 10596. Springer, Cham. [https://doi.org/10.1007/978-3-319-69805-2\\_17](https://doi.org/10.1007/978-3-319-69805-2_17).
- Steyer, K. (2015). Corpus Linguistics Approaches to Proverb Studies. In Hrisztova-Gotthardt, H., & Varga, M. A. (Eds.), *Introduction to Paremiology: A Comprehensive Guide to Proverb Studies* (pp. 206-228). De Gruyter.

## **CHAPTER 4**

### **DIGITAL HERITAGE**



# THE RECURRENCE OF THE 16<sup>TH</sup>-CENTURY ROMANIAN PSALTERS VOCABULARY IN THE 17<sup>TH</sup>-CENTURY LEXICONS BASED ON AUTOMATIC CORPORA COMPARISON

ANA-MARIA GÎNSAC<sup>1</sup>, MIHAI ALEX MORUZ<sup>1,2</sup>,  
AND MĂDĂLINA UNGUREANU<sup>1</sup>

<sup>1</sup> *Department of Social Sciences and Humanities, Institute of Interdisciplinary Research, “Alexandru Ioan Cuza” University of Iași, Bd. Carol I, nr. 11, Romania*

<sup>2</sup> *Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași*

## Abstract

The first translation of the Psalter into Romanian occurred at the beginning of the 16<sup>th</sup> century. Being copied, this unpreserved translation from Church-Slavonic was subjected to revisions following other Slavonic sources, resulting in related versions of which four manuscripts and three printed texts survived; four of them are Slavonic-Romanian. They are presently digitally edited within the project roPSALT (*Re-evaluating the Sixteenth Century Romanian Psalters. Aligned Corpus and Comparative Studies*). Since the Psalter is among the oldest Romanian texts, one may assume that it influenced the vocabulary from the first Slavonic-Romanian lexicons, compiled a century later. These lexicons contain biblical and hagiographical vocabulary and were adapted following Pamvo Berynda’s Slavonic-Ruthenian lexicon (Kyiv, 1627). They are edited within the project eROMLEX (*The First Romanian Bilingual Dictionaries from the 17<sup>th</sup> Century. Digitally Annotated and Aligned Corpus*). On the basis of data resulting from the automatic alignment of their vocabulary, we analyzed the degree of recurrence of the 16<sup>th</sup> century Romanian autosemantic words from the first 70 Psalms in the next century’s lexicons, focusing on words missing in the lexicons, their Slavonic equivalent and linguistic area(s). We argue that the lack of some Psalter’s specific terms in the Slavonic-Romanian lexicons is due to diachronic and dialectal differences, intrinsic features of both text type and translation equivalence.

## 1 Introduction

The roPSALT and eROMLEX are bilingual comparable translation corpora (for the terminology and corpora comparison, see (Krugger, 2002:87; Granger, 2010:3; Lapshinova-Koltunski, 2013; Barrière 2016:105)). They both contain old Romanian biblical vocabulary translated from Slavonic. Following its automatic alignment and linguistic analysis, this paper surveys aspects related to the recurrence of the 16<sup>th</sup> century (Slavonic-)Romanian Psalters’ vocabulary (from the first 70 Psalms) in the Slavonic-Romanian lexicons compiled a century later.

## 2 Data

### 2.1 Corpora Description

The roPSALT corpus<sup>1</sup> contains the Romanian version of the Psalter from the 16<sup>th</sup> century, aligned with Slavonic and Latin sources used for comparison. The first Romanian translation of the Psalter occurred at the beginning of the 16<sup>th</sup> century or the end of the previous one. The text was translated from Slavonic, as the numerous loans, calques, sentence structures, and translation errors show (Candrea, 1916; Mareș, 1982, etc.). This primary Romanian translation, unpreserved,

---

<sup>1</sup> Presently under construction within the project *Re-evaluating the Sixteenth Century Romanian Psalters. Aligned Corpus and Comparative Studies*.

was copied and revised following Church-Slavonic (Mareş, 1982)<sup>2</sup> and possibly Latin sources<sup>3</sup>, resulting in related versions of which four manuscripts and three printed versions have survived. Four of them are Slavonic-Romanian; the text is arranged alternately, i.e., after a short Slavonic sequence, the corresponding Romanian text follows, but the Slavonic text (of a late redaction) is not the source of the Romanian text (Mareş, 2005:276). Three manuscripts belong to the group of “rhotacizing” texts, as they show the conversion of intervocalic *n* into *r* (eventually through the phase *nr*)<sup>4</sup> in words of Latin origin. The oldest of them, the “*Hurmuzaki*” *Psalter*<sup>5</sup>, was dated before 1516 or even towards the end of the 15<sup>th</sup> century, and it may be so far the oldest Romanian text preserved (Mareş, 2000). The *Scheian Psalter*, dated between c. 1573 and 1578 (Mareş, 1984), has the highest percentage of words inherited from Latin compared to the other versions. The Slavonic-Romanian version from Voroneţ, dated between c. 1551 and 1558 (Gheţie and Mareş, 1985:299), was not considered here, as it preserves only the Psalms from 77 onwards, with consistent *lacunae*. The localization of these texts was widely debated. Most scholars consider that the two linguistic layers in the text (from Moldavia and Banat-Hunedoara) point to the Banat-Hunedoara area for the primary translation (Gheţie and Mareş, 1985:300). The text was then copied in the North of Moldavia, possibly through a version copied in the North-Transylvanian dialectal area (Gheţie, 1982:246). In the descendance of the “rhotacizing” *Psalter*s, there are three versions – printed by deacon Coresi and <Şerban> Coresi – from the second half of the 16<sup>th</sup> century, as well as a manuscript with Moldavian dialectal features from the same century, known as the “*Ciobanu*” *Psalter*; this text does not preserve the files containing the first 47 Psalms. The division of the Psalms into 20 *kathismata* follows the Orthodox monastic typicon; the Psalms are not divided into verses.

While the oldest Romanian *Psalter*s have been studied for more than a century, their sources, translation practices, and relations are not entirely clarified. Aiming to compare them with their sources and other similar mono- and bilingual corpora, the Romanian texts (phonetically transcribed into the Latin alphabet) and Slavonic and Latin versions were collected in Word and aligned at the verse level according to the version from the oldest Romanian printed Bible (= BB). The texts were annotated within an interface built with Oxygen XML Editor (see Fig. 1), having the following utilities: (a) indexing and aligning texts at verse level; (b) integration and visualization of the Cyrillic Unicode font (CyrillicaBulgarian10U); (c) simultaneous viewing, editing, and updating the texts; (d) importability of (new) texts from Word and exportability of annotated texts in other types of databases (e.g., TEI, pdf.).

As regards the eROMLEX corpus<sup>6</sup>, it contains the oldest Slavonic-Romanian lexicons, which are, at the same time, the oldest Romanian dictionaries (an exception is *Dictionarium Valachico-Latinum*, from the middle of the 17<sup>th</sup> century (Chivu, 2021)). As the first lexicon was written (or copied from another Romanian version) in 1649, by the monk Mardarie, at the Cozia Monastery, the other five lexicons date from the second half of the 17<sup>th</sup> century and are modified copies from a unique lexicon, which has not been preserved. Their compilation was based on the Slavonic-Ruthenian lexicon published in 1627 in Kyiv by Pamvo Berynda, which contains biblical and hagiographical vocabulary (around 7,000 entries), including among its sources the *Bible of Ostrog* (1581), polyglot bibles and various glossaries (Rozumnyj, 1968:1). In relation to their source, the Romanian lexicons show various changes at the inventory level (omission or addition of entries) and the definition level (simplifi-

<sup>2</sup> See also Camară (2022) for the newest research about the Church-Slavonic sources of the 16<sup>th</sup> century Romanian *Psalter*s.

<sup>3</sup> This hypothesis is based on Latin inherited words mainly found in the *Scheian Psalter* (see chapter 3.2 for examples) that might be explained through a Latin source used either for the primary translation of the *Psalter* into Romanian (Chiţimia, 1981) or during its revision (Munteanu, 2008), in the process of its transmission. Gheţie (1982) rejects Chiţimia’s argumentation, but assumes that a complete comparison of the Romanian texts with the Slavonic and Latin versions from the epoch is needed to address this thesis.

<sup>4</sup> See Gheţie (1975:135).

<sup>5</sup> See the *sigla* of each text in the References.

<sup>6</sup> The corpus was developed within the project *The first Romanian Bilingual Dictionaries from the 17th Century. Digitally Annotated and Aligned Corpus*. For a comprehensive description, see Gînsac et al. (2021).

cation or explanatory additions). Regarding the purpose of their compilation, the lexicons are related to the translation of liturgical texts (Camară (2021:131) links Staicu to the translation of the Gospel from 1697) or even the “Slavonic documents of property” (Gherman, 2021:2); another hypothesis points to their function as didactic tools for learning the Slavonic language (Gînsac and Ungureanu, 2018:873). The lexicons are presently digitally annotated and aligned at the level of the lexicographic entrance (Gînsac et al., 2022:225–226). Only Staicu and Mard. have been lexicographically exploited in DLR.

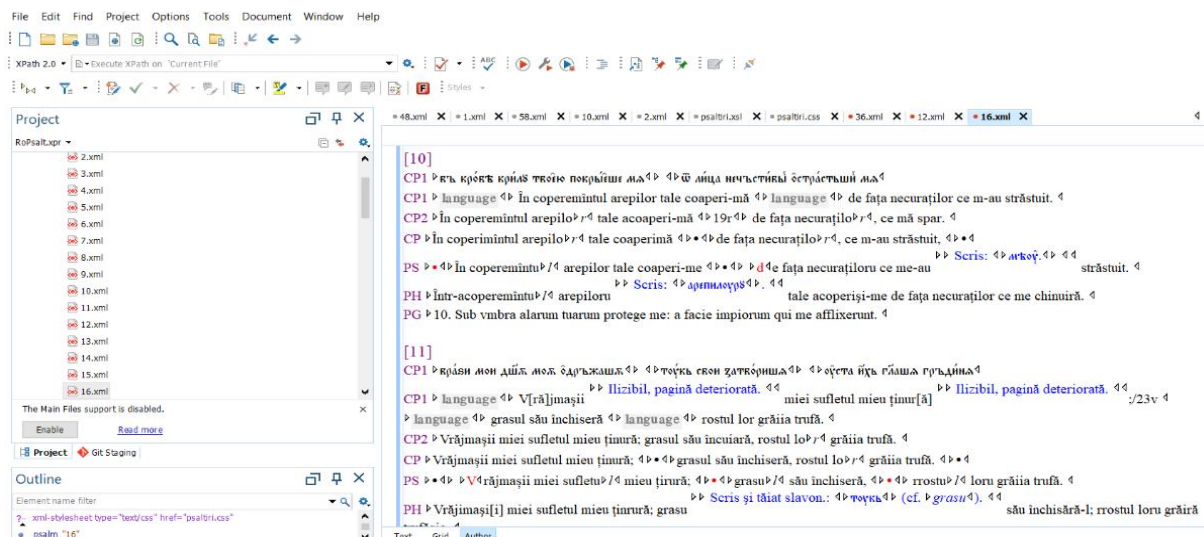


Figure 1. The roPSALT corpus: Oxygen XML editing interface

## 2.2 Database and Methodology

In order to extract the relevant statistical data, both the Psalter corpus and the bilingual lexicons had to be pre-processed. Initially, the Psalters, which are available in manuscript form, were manually transcribed into the Latin alphabet and stored in a rich text format. The rich text format was then transformed to an XML format, as shown in Fig. 1; the actual text of each Psalm (by which we mean the verses in themselves, without notations and observations) from each of the available versions was extracted and then morphologically analyzed. Similarly, the glosses from the lexicon entries were also morphologically analyzed in order to increase the precision of the matching. The morphological analysis is necessary because we only searched for autosemantic words (i.e., nouns, verbs, adjectives, and adverbs). Also, without access to the lemma form, matching becomes significantly more difficult and prone to error. Initially, we attempted to forego lemmatization for the lexicon glosses, seeing as those particular spans of text are not proper sentences but rather standalone words and short phrases. However, the decrease in matching accuracy was very large. In the case of the lexicons, automatic lemmatization was not followed by a manual correction, which resulted in some errors in analysis; unfortunately, it is not possible to determine the exact error rate without manual correction, which is not within the scope of our research. Some of these errors are due to the specific type of text in lexicon glosses, which are not usually made up of full sentences, and are thus less likely to be correctly analyzed. This type of morphological processing was not an objective for the eROMLEX project, but could prove an interesting further development.

For the analysis, we have used the entire available lexicon corpus, which we matched against the first 70 Psalms in 6 Psalters (PH, PS, CP, CP<sub>1</sub>, CP<sub>2</sub>, and PC). The seventh Psalter, PV, is missing the first 70 Psalms. For each candidate word in the Psalters, we have attempted to automatically find a match in the lexicons using a multi-tiered approach:

- first, a perfect lemma match is sought in the glosses of each lexicon, and once the first match is found for one lexicon, the process continues to the next;

- b) if a perfect lemma match is not found, the system then attempts to find a direct match to a surface form available in the lexicons, to account for potential errors in morphological analysis;
- c) if a match is still not found, the system then attempts to match on the basis of word similarity, computed as an aggregate of a Levenshtein distance and a k-shingle similarity measure (cosine similarity).

The system stops at the first match, since our analysis is more interested in whether the words in the Psalters were included in the vocabulary of the 17<sup>th</sup>-century lexicons, irrespective of which headword in each lexicon it might be.

### 3 Data Analysis

The roPSALT and eROMLEX bilingual corpora contain the same type of vocabulary and are both related to Slavonic sources. To determine what part of the Psalters' vocabulary was not preserved in the following centuries' Slavonic-Romanian lexicons, we have selected from the database the Romanian lexemes (nouns, adjectives, verbs, and adverbs) from the first 70 Psalms not or rarely occurring in the Slavonic-Romanian lexicons. Two criteria were followed, explaining their non-inclusion in the lexicons: (a) the common Slavonic source word (where the case) and its treatment from one corpus to another; (b) the attestation of the word following the 16<sup>th</sup> century; (c) the diachronic and dialectal differences. The data were structured in two categories corresponding to the oldest Psalters, PH and PS, which belong to different groups (PH to group A; PS and the Coresian versions – to group B)<sup>7</sup> and have distinct lexical particularities (Mareş, 1982:260); the words occurring in both PS and PH (e.g., *a năstăvi* 'to guide, to direct') were discussed only once.

#### 3.1 Statistical Data

As stated in Section 2.2, we have analyzed Psalms 1 through 70 in all of the Psalters in our corpus; the table below gives the statistics for the matching process.

Table 1. Occurrences of 16<sup>th</sup>-century Psalter words in the next-century lexicons

|                                 | <b>Ber.</b>  | <b>Mosc</b>  | <b>Staicu</b> | <b>Lex<br/>3473</b> | <b>Pet.</b>  | <b>Lex<br/>1348</b> | <b>Mard.</b> | <b>Not<br/>found</b> |
|---------------------------------|--------------|--------------|---------------|---------------------|--------------|---------------------|--------------|----------------------|
| CP <sub>1</sub><br>(924 words)  | 751<br>(81%) | 654<br>(71%) | 629<br>(68%)  | 680<br>(73%)        | 657<br>(71%) | 468<br>(51%)        | 583<br>(63%) | 140<br>(15%)         |
| CP <sub>2</sub><br>(1020 words) | 812<br>(79%) | 712<br>(70%) | 682<br>(67%)  | 740<br>(72%)        | 711<br>(70%) | 509<br>(50%)        | 634<br>(62%) | 169<br>(16%)         |
| CP<br>(899 words)               | 719<br>(80%) | 631<br>(70%) | 605<br>(67%)  | 655<br>(73%)        | 636<br>(71%) | 454<br>(50%)        | 561<br>(62%) | 150<br>(17%)         |
| PC<br>(586 words)               | 505<br>(86%) | 448<br>(76%) | 440<br>(75%)  | 466<br>(80%)        | 455<br>(77%) | 346<br>(59%)        | 410<br>(70%) | 66<br>(11%)          |
| PS<br>(816 words)               | 618<br>(76%) | 548<br>(67%) | 512<br>(63%)  | 568<br>(70%)        | 543<br>(66%) | 392<br>(48%)        | 487<br>(60%) | 175<br>(21%)         |
| PH<br>(851 words)               | 650<br>(76%) | 571<br>(67%) | 552<br>(65%)  | 604<br>(71%)        | 586<br>(69%) | 420<br>(49%)        | 506<br>(59%) | 170<br>(20%)         |

For each of the texts we have analyzed, we have given the total number of distinct autosemantic lemmas identified (in brackets, in the first column). The next column, Ber., gives the number of words in the corresponding Psalter that have a match in one of the lexicons, and the headword of that matching gloss is also found in Ber. For the rest of the columns, the numbers show the total amount of words in the corresponding text which are found in at least one gloss in that particu-

<sup>7</sup> For a stemmatic approach, see Mareş (1985:312–314) and Camară (2022:91).

lar lexicon. Finally, the last column shows the number of words in each text which have no match in any of the lexicons.

### 3.2 Linguistical Analysis

#### 1) “Hurmuzaki” Psalter’s vocabulary and its recurrence in the Slavonic-Romanian lexicons

The database shows that 170 autosemantic words used in PH do not appear in the eROMLEX corpus. However, some of them are due to errors generated by automatic lemmatization, mostly due to the form of the words. Like the other rhotacizing versions of the Psalter, PH contains phonetic (and thus graphic) features specific to the Banat-Hunedoara and North-Moldavian dialectal areas, inaccurately identified by the lemmatizer, which was trained on biblical texts from the Southern dialectal area (BB, OT4389) written in the 17<sup>th</sup> century. Most problems are caused by rhotacism and archaic forms from oblique cases. Moreover, PH is the oldest preserved Romanian text; some of its archaic features (graphic/phonetic and lexical) are also incorrectly processed by the lemmatizer. PH words not found in the Slavonic-Romanian lexicons from the 17<sup>th</sup> century fall into several categories:

(a) Words whose Slavonic equivalent (from the Slavonic Psalter) has not been included in Ber.

We should mention here that Berynda’s lexicon contains Slavonic words that were not understood by Ruthenian users, rather than an exhaustive vocabulary of the Slavonic language, despite the impressive number of entries, around 7,000. Therefore, such words, as the Slavonic equivalents of the Romanian words discussed below, are not listed as entries in the Slavonic-Romanian lexicons:

(1) *blagoslovenie* ‘blessing’ (Ps. 20:34,<sup>8</sup> cf. all versions), cf. Slv. БЛ҃ГОВЕНІЕ. Ber. does not include this Slavonic word, nor the Romanian lexicons add it, but they use words from the same lexical family (Staicu 48<sup>r</sup> uses *binecuvîntat* “blessed” as an equivalent for БЛАГОСЛОВЕНЬ). However, *blagoslovenie* was not altogether forgotten in the era, as it can be found in BB (cf. OT4389 *buna cuvîntare*).

(2) *ciudat* ‘marvellously’ (Ps. 44:6<sub>5</sub>, cf. group B *mirurat*), cf. Slv. ДИВНО ‘mirabiliter’ (Mikl., s.v.). Ber. does not include this adverb, which therefore does not appear in the Romanian lexicons, where, however, words of the same lexical family are translated into Romanian by derivatives of *minune*. Instead, *Mard.* 174 uses *ciudat* with the actual meaning ‘strange, unusual’ as equivalent for Slv. НЕНАЧАЕМЫЙ, while other lexicons opt for the word *neaşteptat* ‘unexpected’ (Mosc. 115<sup>v</sup>, Pet. 48<sup>v</sup> *cel neaşteptat, minunat*).

(3) *a ciudi* ‘to wonder’ (Ps 16:8<sub>6</sub>, cf. group B *a înminuna*), cf. Slv. ОУДИВИТИ (СА). The Slavonic word does not appear in Ber., which instead includes ОУДИВЛЯЮ (1<sup>st</sup> person singular present of ОУДИВЛЯТИ), explained in the Romanian lexicons as *a mira* and *a (se) minuna* (*Mard.* 327 *mirumă*, Staicu 155<sup>v</sup> *minunez pre alții eu*, Lex3473, 285<sup>r</sup>, Mosc. 208<sup>v</sup>, Pet. 82<sup>r</sup> *mir pre alții eu*). In the mid-17<sup>th</sup> century, the word was highly specialized for religious texts.

(4) *a îndrăgi* ‘to love’ (Ps. 25:8, cf. group B *a iubi*), cf. Slv. ВЪЗЛЮБИТИ. The verb does not appear in Ber., probably because it was so well known.

(5) *înfierbîntat* ‘burning, ardent, tried by fire’ (Ps. 17:33<sub>30</sub>, cf. group B *aprints*), cf. the textual form Slv. РАЖДЕЖЕНА. This adjective is not included in Ber., unlike the corresponding noun РАЖДЕЖЕНІЕ (Mikl. Lat. *incendium*) with the Romanian equivalent *aprinde* and *încindere* in Staicu 131<sup>r</sup>. In such cases, where PH opts for a word with a less common meaning (as opposed to other Psalters), its absence from the lexicons is due to the fact that the lexicographic works contain words with their main meaning, independent of context.

<sup>8</sup> The corresponding verse number from the *Septuagint* is displayed in the subscript.

(6) *a oteși* ‘to comfort, to soothe’ (Ps 22:5<sub>4</sub>, group B *a mîngîia*), cf. Slv. оутѣшити. The Slavonic-Romanian lexicons do not include this entry, as it is not indexed in their source. Nevertheless, the word is attested in the 17<sup>th</sup> century (in Cantemir, Milescu, OT4389, cf. DLR).

(7) *potop* ‘flood, inundation, deluge’ (Ps. 28:9<sub>10</sub>, in all versions), cf. Slv. потопъ. Ber. does not include this word, nor the Slavonic-Romanian lexicons.

(8) *a prilăsti* ‘to deceive, to practice deceit’ (Ps. 35:2<sub>3</sub>, group B *a hicleni*), cf. Slv. оульстити. Ber. does not include this Slavonic word among its entries; the same observation applies to the derivative noun *prilăstire*.

(9) *a căzni* ‘to torment, to punish’ (Ps. 37:1, PH *nece căzni*, group B *nu pedepsi*, cf. Slv. не покаже). Frequently attested in the 17<sup>th</sup> century, the word occurs only in Mard. 358, s.v. *вглашенный*, as part of an explanation referring to a specific moment in the ritual of the divine service: *cei ce nu sînt încă nebotezați sau sînt supt canon greu de păcate; ...unii ca aceia trebuie să iasă afară din besearecă, că rău sunt căzniți de îngeri...* ‘those who are not yet baptized or are under heavy canon for their sins; some like that must come out of the church, because they are badly tormented by the angels’; this addition is missing from the other lexicons.

(b) Regional or rarely used words

(10) *bănat* ‘grief, sorrow’ (Ps. 17:7, cf. group B *scîrbie*), with 17 occurrences in the first 70 Psalms, is specific to the Banat-Hunedoara-Transylvania dialectal area. In contrast, in the other versions, it is replaced by *scîrbie* (from Slv. скръбь), as in the Slavonic-Romanian lexicons from the 17<sup>th</sup> century.

(11) *învăscut* ‘dressed’ (Ps. 44:11<sub>10</sub>, cf. group B *îneștit*) and *a învește* ‘to dress’ (Ps. 34:15<sub>13</sub>, in all Psalters), although attested in the 17<sup>th</sup> century, was most likely archaic at the time when the lexicons were compiled. The equivalent of Slv. *одеваю* (1<sup>st</sup> person singular) in Staicu 167<sup>r</sup> is *înbrac* (cf. the other lexicons and OT4389).

(12) *a dodei* ‘to anger (someone)’ (Ps. 34:15<sub>13</sub>, *dodeiră*, cf. Slv. сѣжжахъ). Although a regional word (ILRLEV:189), specific to the areas of Moldavia, Transylvania, Banat, and occasionally Oltenia, it appears in all the 16<sup>th</sup>-century Psalters. A century later, only Mard. 284 renders сѣжжаю through *dodeiesc*, the other lexicons provide the equivalent *a supăra*.

(c) Loans from Slavonic

(13) *aspidă* ‘viper; poisonous snake’ (Ps. 57:4<sub>5</sub>, in all Psalters), cf. Slv. аспидъ. Although included in Ber., none of the Romanian lexicons retains it, probably because it was too specialized. According to DLR, it is still recorded in the 17<sup>th</sup> century, mainly in Moldavia, in the works of Varlaam and Dosoftei.

(14) *heruvim* (Ps. 17:12<sub>11</sub>, in all Psalters), cf. Slv. херувимъ. The Romanian lexicons do not provide the equivalent from Ber., which would have been superfluous, but translate the definition, as is generally the case with the entries in the second part of Ber.: Mard. 216 *plin den înțelepciune* ‘full of wisdom’, Staicu 162<sup>v</sup>, Lex3473, Mosc. *mulțime de înțelese sau vărsare de înțelepciune* ‘much sense or flood of wisdom, i.e., much wisdom’. Ber. bases his (folk) explanation entirely on the Patristic literature (Rozumnyj, 1968:420).

(15) *mîscă* ‘mule’ (Ps. 31:11<sub>9</sub>, PS *mujdeiu*, the other versions *mașcoiul*). As evident from these various renderings, translators had difficulties translating the Slv. *мъскъ* (Mikl. Lat. *mulus*). The Romanian lexicons use the equivalents *mîșcoi* and *catîr* (Mard. 157) and only *catîr* in later lexicons. *Mîscă* is a contextual loan, not recorded in either DLR or Tikin.

(16) *plean* ‘captivity’ (Ps. 52:8<sub>7</sub>, group B *prădarea*<sup>9</sup>), cf. Slv. плѣненіе (Mikl. Lat. *captivitas*). Romanian lexicons render this Slavonic word as *prădare*, *robie* (see Staicu 114<sup>v</sup>). The loan *plean* with this archaic meaning is also attested in Dosoftei’s Psalter from 1680 (DLR, s.v.).

<sup>9</sup> The form *prădară* ‘take into captivity’ from PS is probably due to a copyist error.

(17) *séver* ‘north’ (Ps. 47:2<sub>3</sub>, in all versions), cf. Slv. *сѣверъ*, translating gr. βορέας, has not been used as an equivalent in the Romanian lexicons. Instead, it is rendered as *miazănoapte*: Staicu 145<sup>v</sup>, Mosc. 189<sup>v</sup> *miazănoapte* “north”, Mard. 301 *miazănoapte, partea despre miazănoapte* “north, the north side”, Lex3473, 261<sup>v</sup>, Pet. 75<sup>v</sup> *miază de noapte* “lit. middle of the night”. In the 17th century, its use was restricted to religious translations from Slavonic (DLR, s.v.).

2) *Scheian Psalter*’s vocabulary and its recurrence in the Slavonic-Romanian lexicons  
Among the first Romanian Psalters, PS has the highest percentage of words inherited from Latin. Based on the information provided by the common database, we will discuss a few words that do not or rarely appear in the lexicons, looking for their equivalence from Slavonic in both corpora.

(a) Loans and loan translations from Slavonic not attested or rarely attested in the lexicons

(1) *a năstăvi* ‘to guide, to direct’ (Ps. 42:3, cf. PH, CP<sub>1</sub> *a dereage*, CP<sub>2</sub> *a îndrepta*; Ps. 60:3, PH *îndereptatu-me-ai*, group B *năstăvitu-me-ai*, Slv. *настави мѣ*). Besides the loanword, the 16<sup>th</sup> Psalters contain several Romanian equivalents for this Slavonic word. According to DLR, this verb rarely appears in the next two centuries. Although it is not included in Ber., the Romanian lexicons additionally include vb. *наставляю* and the forms *настави* and *наставит*, which are explained by *a îndrepta* “to guide” (Staicu 140<sup>v</sup>, Mosc. 112<sup>r</sup>); OT4389 and Dosoftei use the term *a povăști*.

(2) *pești* ‘flesh; body’ (Ps. 55:4, cf. PH, CP<sub>1</sub>, PC *pești*, CP<sub>2</sub> *trup*, Slv. *плѣтъ*). Although *pești* was in use in the 17<sup>th</sup> century (see DLR, s.v.), the Slavonic-Romanian lexicons opt for the equivalent *trup* “body” (Mard. 196, Staicu 114<sup>v</sup>, Lex3473, 185<sup>v</sup>), which translates the word *тѣло* ‘corpus humanum’ (Mikl., s.v.) from Berynda’s definition.

(3) *a postîmpi* ‘to approach’ (Ps. 33:5<sub>6</sub>, cf. group B *postîmpiți*, PH *apropiați-vă*, Slv. *постѣпнѣ*). The loanword appears in the 17<sup>th</sup>-century lexicons as an equivalent for Slv. *ѿстѣпи* – *postîmpește* (see Staicu 164<sup>r</sup>), then it is no longer attested.

(4) *schimen* ‘lionet’ (Ps. 16:13<sub>12</sub>, cf. group B *schimenul*, PH *scîncitoriul*, Slv. *скимѣн*) is attested only in the Psalms. Slv. *скѣмѣн* gets a Romanian equivalent in the next century: *puiu de leu tânăr* “young lionet” (Staicu 147<sup>v</sup>) and *leu tânăr* “young lion” (Mard. 304).

(5) pl. *strasti* ‘passion, torment’ (Ps. 31:4, cf. CP<sub>1</sub> *straste*, CP<sub>2</sub> *chin*, PH *munci*, Slv. *страсть*) is used with this meaning until the 17<sup>th</sup> century. The definition from the Slavonic-Romanian lexicons translates the Ruthenian equivalent *пристрастїе* from Ber.: *chin* “tourment” (Staicu 141<sup>v</sup>), *chinuire*, *muncire* “tourmenting” (Mard. 282). The loanword continued to be attested in Romanian at the regional level with the meaning ‘evening service, vespers’ (see DLR, s.v. *straste*).

(6) *a se ucloni* ‘to avoid, to deviate’ (Ps. 52:4, cf. PH *rămară-se*, CP, CP<sub>1</sub>, PC *feriră-se*, CP<sub>2</sub> *lepădară-se*, Slv. *оуклонити сѧ*) is attested in Romanian only in this context. Staicu 156<sup>r</sup> explains vb. *оуклоняю сѧ* through a Romanian equivalent *mă feresc* “I avoid, I stay away”.

(7) *usnă* ‘mouth, lip’ (Ps. 11:2<sub>3</sub>, all versions, Slv. *оустѣна*) is attested until the 17<sup>th</sup> century. Nevertheless, Staicu 158<sup>r</sup> does not use it but renders the Slavonic word as *buzele* “the lips”, to which another hand later adds the Latin equivalent *labia* (cf. Mard. 334 *gura* “the mouth”).

(8) *vătuț* ‘solemn promise’ (Ps. 55:9, cf. PH *făgăduita*, group B *făgăduitul*, OxSlv *ѿбѣтованїе*). The word is attested only in PS (Densusianu, 1997 [1901]:632 points here to *завѣтъ* ‘testamentum’, attested instead in Ps. 24:11). The 17<sup>th</sup>-century lexicons explain the word *ѿбѣтованїе* (DSL, s.v. ‘promesse’) as *făgăduire* (Staicu 166<sup>v</sup>, Mosc. 235<sup>v</sup>).

(b) Words inherited from Latin, not recorded in the 17<sup>th</sup>-century lexicons

(9) *lînged* ‘weakened by illness, sick’ (Ps. 68:34<sub>30</sub>, cf. all the other versions), from Lat. *languidus*, translates Slv. *болѣ* ‘sick, suffering’ and it is also attested in the next century’s Psalter of Dosoftei. The Slavonic-Romanian lexicons index instead Slv. *болѣзнь* with the equivalent *boală*

“disease” (Lex3473, 10<sup>f</sup>). The option *mai bunr* “better” from PH may be due to another source or a translation error.

(10) *strat* ‘bed, couch’ (Ps. 6:6, where it alternates with *așternut* “couch”, cf. PH, CP<sub>1,2</sub> *așternut*), from Lat. *stratum*, is recorded with this meaning in the following centuries at the popular level and in literary works (see Cantemir, Asachi, Brătescu-Voinești, in DLR). The Slavonic-Romanian lexicons do not contain the Slavonic word *ложѣ*, nor the Romanian word *strat*.

(11) *șerb* ‘servant’ (Ps. 68:42<sub>37</sub>, and in the other versions), from Lat. *servus*, is attested with this meaning until the 17<sup>th</sup> century. In the Psalters, it translates Slv. *раба*, rendered in the Slavonic-Romanian lexicons by *rob*, *slugă* “slave, servant” (see Lex3473, 225<sup>r</sup>, Mosc. 158<sup>r</sup>).

(12) *trufă* ‘pride’ (Ps. 16:11<sub>10</sub>, cf. group B *trufă*, PH *trufășie*), from Lat. *trufa*, translates Slv. *гордыня* meaning ‘pride’ and is rarely attested in the next century. The Slavonic-Romanian lexicons instead index the word *гордоусть* with the equivalent *trufie* “pride” (see, for instance, Lex1384, 14<sup>f</sup>), formed by derivation as *trufășie*.

(13) *undă* ‘wave’ (Ps. 41:10<sub>8</sub>, cf. CP<sub>1</sub> *undă*, PH, CP<sub>2</sub> *val*), from Lat. *unda*, renders Slv. *вълна*. This word appears in the definition of *вълненіе* – *valul de mare*, *вълна* “the sea wave” from Lex3473, 30<sup>v</sup>. Ber. defines the word as “wave, where there is a tumult of water, wave on water”.

#### 4 Results and Future Work

Based on data obtained by automatic comparison, this paper surveyed the Romanian vocabulary of two bilingual Slavonic-Romanian corpora from the 16<sup>th</sup> and 17<sup>th</sup> centuries that match in genre and size. The comparison shows that the lack of specific words from the 16<sup>th</sup> century (Slavonic-)Romanian Psalters in the next century’s Slavonic-Romanian lexicons is due to diachronic and dialectal differences, intrinsic features of each text type (liturgical book vs dictionary), as well as translation approach. The 17<sup>th</sup>-century lexicons do not preserve: (a) Slavonic loanwords, although most of them are attested in the 17<sup>th</sup> century and some even later; this is due to the explanatory nature of the definitions from the lexicons, which translate the Ruthenian equivalent from their source; on the other hand, the Psalter contains specialized words borrowed contextually, which usually disappear in the following centuries; (b) words inherited from Latin, usually replaced by other equivalents; (c) words with regional or rare use.

Aligned bilingual corpora can be used in studying the vocabulary and its change in translation and diachrony. We conclude that the lemmatization and morphological processing of both corpora (the Slavonic sources and their Romanian translations) could improve their comparability, offering a more systematized database for further studies on variation in translation equivalence and vocabulary change in diachrony.

#### Acknowledgements

This work was supported by a grant from the Ministry of Research, Innovation and Digitization, CNCS/CCCDI – UEFISCDI, project number PN-III-P4-ID-PCE-2020-2939, within PNCDI III.

#### References

- Academia Română. (2010). *Dicționarul limbii române* (anastatic edition), volumes 1–19. Editura Academiei Române, București.
- Alet Kruger. (2002). Corpus-based translation research: its development and implications for general, literary and Bible translation. *Acta Theologica. Supplementum*, 2:70–106.
- Alexandru Andriescu and Paul Miron (eds.). (2003). *Monumenta linguae Dacoromanorum. Biblia 1688. Pars XI. Liber Psalmorum*. Universitatea “Alexandru Ioan Cuza” din Iași, Iași (= BB)
- Alexandru Mareș. (1982). Filația psaltirilor românești din secolul al XVI-lea. In Ion Gheție (coord.), *Cele mai vechi texte românești. Contribuții filologice și lingvistice*, pages 207–261, Editura Universității, București.

- Alexandru Mareş. (2000). Considerații pe marginea datării Psaltirii Hurmuzaki. *Limba română*, 49(4-6):675–683.
- Alexandru Mareş. (2005). *Scritură și cultură românească veche. Studii și articole*. Editura Academiei Române, București.
- Alin Mihai Gherman. (2021). Lexicography and the history of culture (The Case of Teodor Corbea's Romanian-Latin Dictionary). *Diacronia* 14(A189):1–11, <https://doi.org/10.17684/i14A189en>.
- Ana-Maria Gînsac and Mădălina Ungureanu. (2018). La lexicographie slavonne-roumaine au XVII<sup>e</sup> siècle. *Zeitschrift für romanische Philologie*, 134(3):845–876.
- Ana-Maria Gînsac, Alex Moruz, and Mădălina Ungureanu. (2021). Slavonic-Romanian lexicons of the 17<sup>th</sup> century and their comparative digital edition (the eROMLEX project). *Diacronia*, 14(A192):1–11, <https://doi.org/10.17684/i14A192en>.
- Ana-Maria Gînsac, Alex Moruz, and Mădălina Ungureanu. (2022). The first Romanian dictionaries (17<sup>th</sup> century). Digital aligned corpus. In Annette Klosa-Kückelhaus et al. (eds.), *Dictionaries and Society. Proceedings of the XX EURALEX International Congress*, pages 222–228, IDS-Verlag, Mannheim.
- Caroline Barrière. (2016). *Natural Language Understanding in a Semantic Web Context*. Springer International Publishing, Cham, Switzerland.
- Coresi. (1570). *Psaltire românească*, Braşov (= CP).
- Coresi. (1577). *Psaltire slavo-română*, Braşov (= CP<sub>1</sub>).
- Ekaterina Lapshinova-Koltunski. (2013). VARTRA: A comparable corpus for analysis of translation variation. In *Proceedings of the 6<sup>th</sup> Workshop on Building and Using Comparable Corpora*, pages 77–86, Sofia, Bulgaria.
- Eugen Munteanu. (2008). *Lexicologie biblică românească*. Editura Humanitas, Bucureşti.
- Franz Miklosich. (1865). *Lexicon Palaeoslovenico-Graeco-Latinum*. Braumueller, Vindobonae. <http://www.monumentaserbica.branatomic.com/mikl2/> (last access: 28-09-2022).
- Gheorghe Chivu. (2021). Dictionarium Valachico-Latinum, primul lexicon bilingv original din scrisul românesc. *Diacronia*, 14(A188):1–5.
- <“Hurmuzaki” Psalter>. 16<sup>th</sup> century. Ms. no 3077, Romanian Academy Library, Bucharest (= PH).
- Hariton Tiktin. (2000-2005). *Rumänisch-deutsches Wörterbuch*, volumes I-III, überarbeitete und ergänzte Auflage von Paul Miron und Elsa Lüder. Clusium, Wiesbaden/ Cluj Napoca (= Tiktin).
- Ion Aurel Candrea. (1916). *Psaltirea Scheiană comparată cu celelalte psaltiri din sec. XVI și XVII traduse din slavonește*. Atelierele Grafice Socec&Co., Bucureşti.
- Ion C. Chițimia. (1981). Urme probabile ale unei vechi traduceri din latină în Psaltirea Scheiană. *Revista de istorie și teorie literară*, 30(2):151–156.
- Ion Gheție. (1975). *Baza dialectală a românei literare*. Editura Academiei Române, Bucureşti.
- Ion Gheție. (1982). Banatul și textele rotacizante. *Limba română*, 31(3):238–246.
- Ion Gheție and Al. Mareş. (1985). *Originile scrisului în limba română*. Editura Științifică și Enciclopedică, Bucureşti.
- Iosif Camară. (2021). Evanghelia din 1697 și legătura ei cu școala de slavonie din Țara Românească. *Dacoromania*, 2:119–131.
- Iosif Camară. (2022). New information on the Slavonic Sources of the oldest Romanian Psalters. *Palaeobulgarica*, 46(1):81–94.
- Jaroslav Rozumnyj. (1968). *Proper Names in Pamvo Berynda's “Leksikon slavenorosskij i imen tlkovanie” of 1627* (PhD thesis). University of Ottawa, Ottawa.
- Jean-Paul Deschler and Anastasia Weulersse. 2003. *Manuel du slavon liturgique*, volume 2, *Dictionnaire slavon-français*. Institut d'études slaves, Paris (= DSL).
- <Old Testament>. 17<sup>th</sup> century. Ms. no 4389, Romanian Academy Library, Bucharest (= OT4389).
- Ovid Densusianu. (1997 [1901]). *Histoire de la langue roumaine*. Grai și suflet – Cultura Națională, Bucureşti.
- <Oxford Serbian Psalter>. 14<sup>th</sup> century. Ms. e Mus. 184, The Bodleian Library, Oxford (= OxSlv).

- Pamvo Berynda. (1627). *Leksikon slavenorosskij i imen tlkovanie*, Kiev. <http://litopys.org.ua/berlex/be.htm> (last access: 28-09-2022).
- Romanian Academy. (1997). *Istoria limbii române literare. Epoca veche (1532-1780)*. Editura Academiei Române, București (= ILRLEV).
- Romanian Academy. (2010). *Dicționarul limbii române*. Editura Academiei Române, București (= DLR).
- <Scheian Psalter>. 16<sup>th</sup> century. Ms. no 449, Romanian Academy Library, Bucharest (= PS).
- <Slavonic-Romanian Lexicon>. (1649). Ms. no 450, Romanian Academy Library, Bucharest (= Mard.).
- <Slavonic-Romanian Lexicon>. 17<sup>th</sup> century. Ф. 188, оп. 1, ч. 2, Russian State Archives of Ancient Documents, Moscow (= Mosc.).
- <Slavonic-Romanian Lexicon>. 17<sup>th</sup> century. Ms. no 3473, ff. 1–369<sup>v</sup>, Romanian Academy Library, Bucharest (= Lex3473).
- <Slavonic-Romanian Lexicon>. 17<sup>th</sup> century. Ms. no 312, ff. 41–216<sup>v</sup> Romanian Academy Library, Bucharest (= Staicu).
- <Slavonic-Romanian Lexicon >. 17<sup>th</sup> century. Ms. no Q.XVI.5—Славяно-молдавский словарь 100 листов, составлен в 1695 году, National Library of Russia, Saint Petersburg (= Pet.).
- <Slavonic-Romanian Lexicon>. 17<sup>th</sup> century. Ms. no 1348, ff. 1–84<sup>v</sup>, Romanian Academy Library, Bucharest (= Lex1348).
- Stela Toma (ed.). 1979. *Coresi, Psaltirea slavo-română (1577) în comparație cu psaltirile coresiene din 1570 și din 1589*. Editura Academiei Române, București.
- Sylviane Granger. (2010). Comparable and translation corpora in cross-linguistic research: Design, analysis and applications. *Journal of Shanghai Jiaotong University*, 2:14–21.
- <Șerban Coresi>. c. (1589). *Psaltire slavo-română*, <Brașov> (= CP<sub>2</sub>).

# APPLYING YOLOv5 LEARNING IN DETECTING OLD CYRILLIC-ROMANIAN CHARACTERS

DAN CRISTEA<sup>1,2</sup>, PETRU REBEJA<sup>2</sup>, AND CRISTIAN PĂDURARIU<sup>1,2</sup>

<sup>1</sup>*Institute for Computer Science, Iași branch of the Romanian Academy, 2 Codrescu St.*

<sup>2</sup>*Faculty of Computer Science, Alexandru Ioan Cuza University of Iași, 16 Berthelot St.*

*{danu.cristea, petru.rebeja, cristian.padurariu94}@gmail.com*

## Abstract

Between the 16th century and the middle of the 19<sup>th</sup> century, a unique Cyrillic alphabet was used on the territories of historical Romania, with slight variations in the shapes of graphemes or their phonetic values. As a result, an ample bibliography of Cyrillic-Romanian texts was accumulated in various libraries, whereas very few of these books have been transliterated by a small number of specialized linguists. The huge diversity of shapes of characters, in documents that cover three and a half centuries, makes the classical OCR methods inapplicable. This paper presents new developments in a project intended to develop an application that deciphers old Romanian printings and uncials from the Cyrillic into the Latin script. Our interest here is limited to detecting Cyrillic characters in the scan of a page. Since the creation of annotated resources for a deep learning approach to this task is very time-consuming and the density of characters is highly non-uniform in documents, we developed a strategy for data augmentation that exploits a collection of documents entirely transcribed by experts. Initial identification of rows and characters not yet labeled on the page helps focussing the attention on narrower zones and in the augmentation of data. A whole technological pipeline is designed and implemented to fit a deep learning approach, from the creation of annotated resources, enhancement of original page images, export of data, segmentation of pages, training, and validation against a ground truth of characters manually annotated with surrounding boxes and Latin character values. The central interest in the paper is to build and test an algorithm for page segmentation in smaller windows and to prove that exposing the learner and the detector to these narrower areas much better results are produced than working directly on unsegmented pages.

**Keywords:** Cyrillic characters detection, deep learning, object detection, old Cyrillic-Romanian documents, page segmentation.

## 1. Introduction

This paper presents models and developments intended to realize an application that deciphers old Romanian printings and uncials (excluding manuscripts with ligatures) from the Cyrillic into the Latin script. Our attention here is limited to the detection (localization), of Cyrillic characters on the image of the page. In the context of the general task of deciphering old Cyrillic-Romanian documents in the Latin script, the interest for this operation is twofold: to help a process of data augmentation used in training a deep learning character recognition process and as an intermediary step towards a true character recognition.

Between the 16th century and the middle of the 19<sup>th</sup> century, a unique Cyrillic alphabet was used on the territories of historical Romania, with slight variations in the shapes of graphemes or their phonetic values. Only very few documents from the ample collection of Cyrillic-Romanian texts that have been accumulated in various libraries received the attention of a small number of

specialized linguists who transliterated them. As such, our research<sup>1</sup> is motivated by the necessity to build an application capable of automatically deciphering these texts from Cyrillic into the Latin script, on one side, for the benefit of researchers in different fields and, on the other, to facilitate the interest of editors to offer them to the large public.

Due to the vast diversity of shapes of the characters in documents printed in diverse typographies in the three Romanian provinces or copied by hand in monasteries, over three and a half centuries, the non-uniform writing, the dirty or worn-out pages make impractical the classical OCR methods.

The paper is structured as follows: section 2 very briefly describes similar approaches and some techniques that help the process of character detection, section 3 presents the algorithm of segmentation that precedes the YOLOv5 training and detection processes, section 4 focuses on the experiments, and section 5 – on the conclusions.

## 2. Related Work

Recently, deep learning methods proved to give good results in many object-identification and object-recognition tasks; an attentive state-of-the-art picture is depicted in (Pădurariu and Cristea, 2019). As in all supervised learning tasks, the acquisition of ground truth data is a very expensive and time-consuming activity, necessitating dedicated and high-quality human efforts. To boost this process, Cristea *et al.* (2021a, 2021b) describe a short-circuiting strategy for data augmentation that exploits a collection of old Romanian texts manually transliterated by linguists (critical editions).

Since the density of characters in a document is highly unequal, a similar discrepancy reverberates in the number of instances manually accumulated in the ground truth. Following this strategy, cycles of annotation-training-evaluation-alignment are iteratively performed on a collection of pairs `<page_image, transcribed_text>`, with the goal of gaining more training data than directly annotated by humans. In short, the short circuiting strategy aligns a number of unlabelled (void) character bounding boxes from the image of the page that are placed between a pair of safely recognized ones with the corresponding string of characters in the transcribed text. The set of pairs `<character_shape, label>` thus acquired is added to the ground truth. As such, this process requires a character recognition network sufficiently trained for a certain number of character classes and a character detection network, which is the target of the research reported here.

In a rather similar task, the detection of rows of text, Găman *et al.* (2022) report significant improvements of a YOLOv4 process, when it is augmented with a self-paced training and evaluation strategy (Kumar *et al.*, 2010).

## 3. Method

It is rather obvious that the processes of detecting rows of text and detecting characters are quite similar. What clearly differentiates them are the shapes of the objects to be detected, the ratio between widths and heights of the surrounding frames being much higher in the case of rows than in the case of characters. However, there is also another detail that makes the two endeavors rather different, requiring specific methods of segmentation to be applied for the latest. Indeed, the context in which rows are detected is clearly that of a whole page, while characters need a much narrower context. We will prove in this paper that this detail has significant importance. In this section, we present a segmentation algorithm and experiments that prove that a window

---

<sup>1</sup> The work described here is part of the DeLORo project (Cristea *et al.*, 2020), which went on between 2020 and 2022, as a collaboration between the natural language processing groups in the Institute for Computer Science of the Iași branch of the Romanian Academy and of the Faculty of Mathematics and Computer Science of the University of Bucharest.

segmentation (WS) approach gives better results in the task of detecting characters when compared against a page segmentation approach (PS).

The primary source of our data is the ROCC corpus (Cristea *et al.*, 2021b), which includes: metadata attached to the whole original document and to each page, scans of pages, explicit segmentations of the graphical objects in a page (represented as bounding boxes), labels attached to bounding boxes, and, only for a subset of the pages, the text transcribed by linguistic experts. One of the annotation tasks in collecting training data was to frame character shapes and, partially, to give them labels. However, not all characters in every page are framed<sup>2</sup>; still, a large number of pages contain sufficient material to be used for training in our task.

To form the training and evaluation sets for our task, we considered three possibilities of looking at a page image: (i) as it is (page segmentation – PS), (ii) segmented into rows (row segmentation – RS), and (iii) segmented into square windows (window segmentation – WS). Compared to the approach that takes the page as it is (PS), window segmentation (WS) has the advantage of narrowing the context and of speeding up the training process because the training samples are smaller. Compared to the approach that sees the page first segmented in rows (RS), WS has the advantage of simplifying the detection process to only one step, by eliminating the need to first detect rows. However, we keep as an alternative option also the RS approach, with the intention to concentrate on this possible solution sometimes in the future.

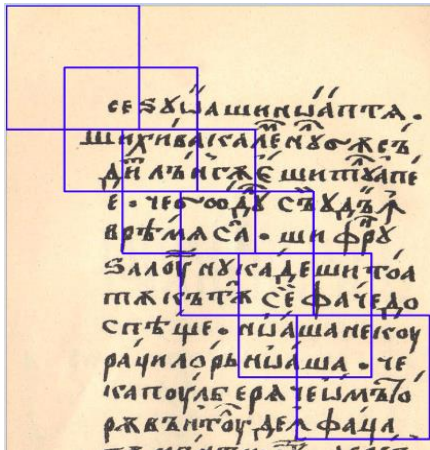


Figure 1. The upper part of a page in the process of window-segmentation

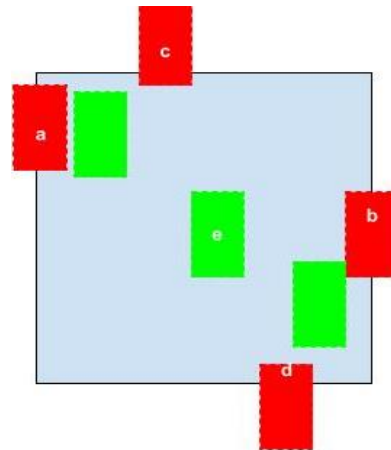


Figure 2. Blurred (in red) and accepted (in green) character boxes in extracting the training samples inside a window

Figure 1 displays a corner of a page containing only a few of the windows in the process of WS. As shown in the figure, a window of size  $n \times n$  pixels slides over the page in overlapping positions (see Fig. 3 for an explanatory sketch), collecting all inner character frames of the ground truth. Only the character boxes that are completely included inside the window are selected in the training sample (in green in Fig. 2). Those that intersect with one of the margins are blurred and do not contribute to the training of the model. When performing segmentation of images in windows, we deliberately remove the annotated characters that are fragmented (cut off) by the window borders. Our initial attempt at removing the cut-off characters was to apply a blur on the annotated area; however, we decided against this approach because it would produce a sharp contrast with the background of the page, which could be at the origin of other types of

<sup>2</sup> Pages where not all characters are framed are a result of changing the annotation strategy, with the goal of balancing the number of annotated instances of each character. Conforming to this strategy change, at the beginning of the project's annotation process, whole pages have been annotated, thus obtaining a highly unbalanced set of characters; to balance out the set, from a certain point on, the annotators were instructed to pay higher attention to specific instances, generally of lower frequency characters, which also lead to missing annotations on certain pages.

errors in training. To avoid a high contrast with the image background, we apply a thresholding filter that transforms the original, colored, image into a black-and-white one. In the future, we consider also applying adaptive thresholding or other binarization techniques (such as linear filtering) in order to remove part of the accidental pixels that could remain out of a dirty background.

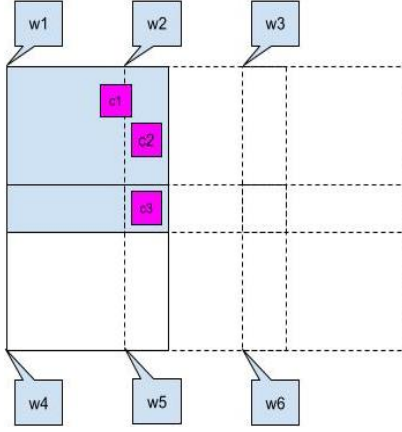


Figure 3. Positions of 6 windows in the original page  $P$  during detection;  $c1, c2, c3$  are characters in  $P$ ; in  $P'$ ,  $c1$  will appear once unbroken (collected in  $w1$ ) and once fragmented (in  $w2$ );  $c2$  will appear twice (collected in  $w1$  and  $w2$ );  $c3$  will appear 4 times (collected in  $w1, w2, w4$  and  $w5$ )

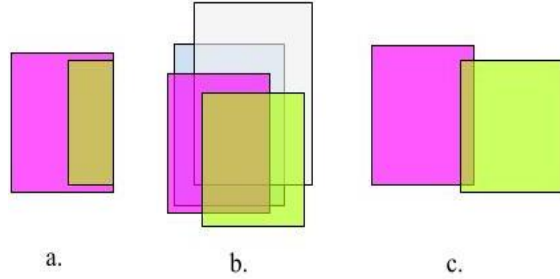


Figure 4. Possible relative positions of instances of character frames in  $P'$

- The same character from  $P$  is fully detected in one window and as a fragment in another window (e.g.,  $c1$  collected in windows  $w1, w2$ );
- One character from  $P$  produces more frame instances in  $P'$  (e.g.,  $c3$  collected in windows  $w1, w2, w4$  and  $w5$ );
- The frames in  $P'$  of two characters from  $P$  slightly intersect (both collected in the same window)

Window segmentation imposes a different treatment for training and detection. In detection, losing characters at boundaries cannot be accepted. It becomes now obvious that windows should intersect over a region that is larger than the maximum size of a character, on both dimensions. Figure 3 sketches such an arrangement. It is like the window sweeps the page horizontally and then vertically, picking data from certain fixed positions and sending them to the trained network. For each such position, the network returns the detected boxes, but, in order to synchronize these boxes with the corresponding characters on the page, filtering heuristics should be applied, for letting only one occurrence of a character when the partial superposition of the window produces multiple instances of the same character or only scraps of letters near the window's boundaries. The algorithm that describes this process is given below as Algorithm 1. In Fig. 3 and 4 and in Algorithm 1,  $P'$  is the reconstructed image of an original page  $P$ .

---

**Algorithm 1:** Filtering character frames in detection

---

**Notations:**

Let  $P'$  be the reconstructed image of an original page  $P$ .

Let  $\mathcal{W}$  be the set of all squared windows  $w$  that swept horizontally and vertically the page  $P$ ;

$f_a$  and  $f_b$  are character frame instances in  $P'$ ;

$x$  is a threshold (between 0 and 1);

$\text{area}(f)$  is a function that returns the area of the frame  $f$ ;

$\text{minAreaFra}(f_a, f_b)$  is a function that returns that frame with the minimal area between  $f_a$  and  $f_b$ ;

$\text{IoU}(f_a, f_b)$  is a function that returns the intersection over the union value of the frames  $f_a$  and  $f_b$ .

---

Phase 1: collecting instances

---

1. for all  $w$  in  $W$ , copy in  $P'$  all instances of detected character frames from  $P$  collected in  $w$ .

Copying is done by converting frame coordinates from  $w$  into  $P'$ , as such:

$$\begin{aligned} f_{h'}^{p'} &= f_h^w + w_h \\ f_{v'}^{p'} &= f_v^w + w_v \end{aligned}$$

where:  $f_{h'}^{p'}$  and  $f_{v'}^{p'}$  are the (horizontal and vertical) coordinates of the upper-left corner of the frame  $f$  in  $P'$ ;  $f_h^w$  and  $f_v^w$  are the coordinates of the upper-left corner of the frame  $f$  in  $w$ ;  $w_h$  ; and  $w_v$  are the coordinates of the upper-left corner of the window  $w$  in  $P'$ .

---

Phase 2: filtering

---

```

2. do {
3.     flag ← FALSE;
4.     if (there are  $f_a$  and  $f_b$ , such that  $\text{area}(f_a) \subseteq \text{area}(f_b)$ )
5.         then {delete  $f_a$ ; // see Figure 4.a
6.             flag ← true;
7.         else if (there are  $f_a$  and  $f_b$ , such that  $\text{IoU}(f_a, f_b) \geq x$ )
8.             then {delete  $\text{minAreaFra}(f_a, f_b)$  ; //see Figure 4.b
9.                 flag ← true;}
10. until (!flag);
    
```

---

Steps 3 to 9 are performed until there are no more two frames in  $P'$  that are either included one in the other (as in Fig. 4.a) or their intersection as compared to their union is above a threshold (as in Fig. 4.b). Cases like the one depicted in Fig. 4.c will not produce deletions. This will ensure that only one instance of a character box is collected by the detector. Moreover, when deciding between what to filter out and what to keep, we give preference to larger boxes than to smaller, with an eye towards the next process of character labeling, which, most probably, will work over the shapes framed in the detected boxes, this way maximizing the chance of not amputating character shapes. The detected boxes are, of course, paired by values indicating confidence factors. In the set of experiments described below, we leave only detected character boxes whose confidence factors are above 0.7.

## 4. Experiments

**Training and evaluation data.** The data were selected from 13 Cyrillic-Romanian books (prints and uncials) covering the centuries XVI<sup>th</sup> to XVIII<sup>th</sup>. Since, as mentioned before, the density of character annotation is not uniform in the acquired ground truth, we sorted all pages in this collection extracted from ROCC in the descending order of an *average-character-density* value. This was estimated by taking samples of pages from each annotated document and counting the lines on a page and the characters per line; then an *average-characters-per-page* number was computed by multiplying these counts. With the assumption that one document has a relatively uniform pagination, this number was considered a constant for every page of the same document. Then, the actual *annotated-character-boxes* were counted directly from the database for each page. With these, the *average-character-density* was obtained as the ratio between the *annotated-character-boxes* and the *average-characters-per-page*.

For WS, the training set consisted of a number of 1000 windows of size 500 by 500 pixels, including 140,324 annotated boxes of characters; the validation set consisted of a number of 500 windows of size 500 by 500 pixels including 48,359 annotated boxes of characters.

For PS, we split the pages containing more than one annotation into two sets of 80% and 20% for training and evaluation, respectively.

**Evaluation.** As an evaluation measure, we report the average precision at an Intersection over Union (IoU) threshold of 0.5, accepted in the majority of published studies.

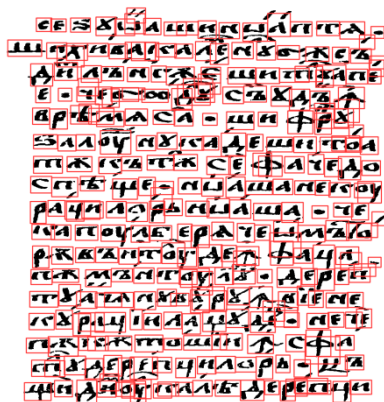
The baseline considered was the detection of characters with a YOLOv5 approach on the original unsegmented pages and a conventional training procedure.

**Hyperparameters.** We fine-tuned the official small model of YOLOv5 with the Cross Stage Partial Networks backbone (Wang *et al.*, 2020). Since our training was performed on CPU, we let the model decide the batch size number by specifying `batch-size -1` in the parameters of the training script. The size of the original unsegmented images was 1600x1600 pixels. The square shape of the original page image was obtained by slightly shrinking it in the vertical direction. The default SGD optimizer was used. The model was trained for 300 epochs. The IoU threshold for the NMS (non-maximum suppression) procedure is set to  $p = 0.5$ .

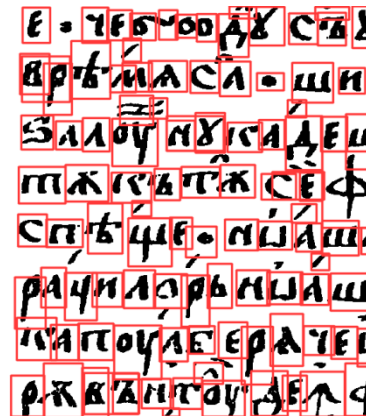
**Results and discussions.** Table 1 shows the results of the application of the WS model compared against the PS baseline. In both cases, the results were obtained with the YOLOv5 small model after 300 epochs. Pairs of samples from annotated pages and annotated windows are displayed in Fig. 5.

Table 1. Comparing WS against PS

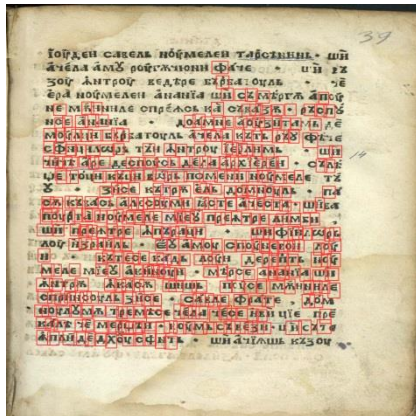
| Strategy                                         | mAP@0.5 |
|--------------------------------------------------|---------|
| PS: full images at 1600x1600 pixels              | 0.82    |
| WS: images segmented into 500x500 pixels windows | 0.99    |



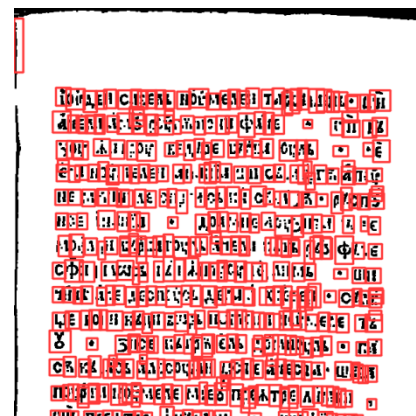
a.



b.



c.



d.

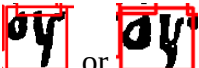
Figure 5. Detection: windows in the WS (b, d) compared against their corresponding entire pages in the PS (a, c)

By analyzing the output, a number of conclusions can be drawn. It is clear that by restricting the size of the image and, with that, the density of detectable objects in the image, the results are improved. Attentively comparing zones in the two images of Fig. 5 (a and b), it can be seen that agglomerations of false objects detected in the baseline PS disappear in the WS. Compare, for



instance: in the PS baseline, against in the WS solution. Also, the subsequent pair of images in Fig. 5 (c and d) display much clearer that the model trained on the whole page (c) misses a lot of characters, but they are detected by the model trained on windows (d).

However, errors still occur, and they are of more types, three of them only being discussed below:

- type A error: a character appears split into two distinct boxes: ;
- type B error: two characters are included in only one box:  or .
- type C error: a pair of characters appears distinctly detected, but also included in one common box: .

We suggest more syntactic heuristics that could configure possible solutions to these cases, the actual implementations remaining for future work. A bunch of such heuristics could configure the third phase of Algorithm 1: post-processing or error reduction. The type A error is characterized by a rather large intersection of two boxes. Taking the extreme values of the coordinates of the two boxes in all four directions (upper-most, lower-most, right-most, and left-most) would produce a box certainly covering both fragments. The type B error can be reduced in the recognition phase by learning combinations of characters frequently occurring together (in Figure 5b, there are 3 occurrences of the kind exemplified) as two consecutive labels. Type C error is a combination of type A and type B. If there are 3 or more intersecting boxes, such that one of them has a much larger intersection with any of the others, while the intersections of any two of these last ones are much smaller, then there is a high probability that each of the smaller boxes surrounds individual characters and the larger box should be eliminated.

## 5. Conclusions

We presented in this paper an algorithm and a methodology for characters detection in old Romanian texts that use the Cyrillic script. This will complement as a preliminary step both an alignment process between images of pages and their manual transcriptions, in view of gaining more training data with zero annotation costs (Cristea *et al.*, 2021a, 2021b), as well as the actual process of character recognition. The novelty of our approach is in preparing the input image segmented in windows, both for training and for detection, following a classical YOLOv5 architecture of learner/detector. This proves to add to the results 17% more well-detected boxes with respect to the same procedure that processes non-segmented pages. The evaluation measure is the average precision (AP) at an Intersection over Union (IoU) threshold of 0.5. A visual inspection of the type of errors induces a set of post-detection syntactic heuristics capable to reduce even further the errors.

Apart from applying these heuristics, the good results reported in this paper, which come from a previous focussing on just a zone of the original page, suggest also trying another type of

preliminary segmentation: row segmentation. Then, both window and row segmentation can be combined with a self-paced learning approach (Găman *et al.*, 2022) and retain the best solution.

## Acknowledgements

This work has been carried out with the financial support of the Romanian Executive Unit for Financing Higher Education, Research, Development and Innovation (UEFISCDI), in the project *Artificial Intelligence Models (Deep Learning) Applied in the Analysis of Old Romanian Language (DeLoRo – Deep Learning for Old Romanian)*, project code PN-III-P2-2.1-PED-2019-3952.

We are grateful to our non-author colleagues in the project: Cecilia Speranța Bolea, Paula Alexandra Crucianu, Daniela Gîfu, Gabriela Haja, Mihaela Onofrei, Andrei Scutelnicu, Elena Tamba, Roxana Vieru, involved in forming the ROCC collection and the database access technology behind, and Mihaela Găman, Radu Ionescu, and Marius Popescu – in experimenting detection and recognition DL models.

## References

- Cristea, D., Pădurariu, C., Rebeja, P., and Onofrei, M. (2020). From Scan to Text. Methodology, Solutions and Perspectives of Deciphering Old Cyrillic Romanian Documents into the Latin Script. In: *Knowledge, Language, Models, Volume in Honour of Prof. Galia Angelova on Her 65th Birthday* (pp. 38-56). Milena Slavcheva (Ed.), Shoumen, BULGARIA: INCOMA Ltd. ISBN 978-954-452-062-5.
- Cristea, D., Pădurariu, C., Rebeja, P., Scutelnicu, L. A., Onofrei M., Crucianu, P. A., Bolea, C.S., Gîfu, D., Haja, D., Tamba, I., and Vieru, R. (2021a). Short-circuiting manual annotation in the process of building a corpus of old Cyrillic Romanian, Presented at the Symposium *Intelligent Systems and Applications*, in the „Iași Academic Days”, the XXXVI<sup>th</sup> edition, 22 October, Iași, Romania.
- Cristea, D., Pădurariu, C., Rebeja, P., Scutelnicu, L.A., and Onofrei, M. (2021b). Data Structure and acquisition in DeLoRo. A Technology for Deciphering Old-Cyrillic-Romanian document. In *Proceedings of the 16th International Conference “Linguistic Resources and Tools for Natural Language Processing”*, 13-14 Dec., “Alexandru Ioan Cuza” University Printing House, Iași, pp. 59-74. ISSN 1843-911X.
- Găman, M., Ghadamiyan, L., Ionescu, R.T., and Popescu, M. (2022). Self-paced learning to improve text row detection in historical documents with missing labels. In *Proceedings of TiE (ECCV Workshop)*.
- Kumar, M.P., Packer, B., and Koller, D. (2010). Self-Paced Learning for Latent Variable Models. In: *Proceedings of NIPS*, vol. 23, pp. 1189–1197.
- Pădurariu C. and Cristea, D. (2019). *Solution for scanned documents segmentation and letter recognition*. In *Proceedings of the 14th edition of the International Conference on Linguistic Resources and Tools for Natural Language Processing – ConsILR-2019*, “Alexandru Ioan Cuza” University Printing House, Iași, ISSN: 1843-911X.
- Wang, C.-Y., Mark, L.H.-Y., I-Hau, Y., Yueh-Hua, W., Ping-Yang C., and Jun-Wei, H. (2019). CSPNet: A New Backbone that can Enhance Learning Capability of CNN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, <https://arxiv.org/abs/1911.11929>.

# EVALUATION OF PHILOLOGICAL RESOURCES IN THE ONLINE ENVIRONMENT

EUGENIA MINCU

*Institute of Romanian Philology “B. P.- Hasdeu”, State University of Moldova*

## Abstract

The present paper is based on the achievements of the national project “Scientific valorization of the national linguistic heritage in the context of European integration”, State Program (2020-2023), implemented in partnership with the Institute of Mathematics and Computer Science “Vladimir Andrunachievici”.

The research focus of the project concerns the Romanian language, including the revaluation of the static element of the Romanian language (valorization of old Romanian texts; safeguarding of the dialectological archive; digitization and transliteration into Latin alphabet of dialect texts, etc.) and the elucidation of the dynamic element of the Romanian language (dictionaries, elements of language cultivation, etc.).

In this context, *LogosPlus* (*logosplus.ifr.md*), a portal of philological resources, was created in collaboration with the Institute of Mathematics and Computer Science “Vladimir Andrunachievici”, which aims to be a promoter of national values (language and culture) and is intended for the general public.

*LogosPlus* promotes the norm and aesthetics of the Romanian language through etymologizing and language cultivation tablets (video and Word) distributed in several subsections.

The Online Dictionaries section includes the platform for the management of the electronic dictionaries developed (*dex.ifr.md*): Dictionary of Synonyms of the Romanian Language and Usual Explanatory Dictionary of the Romanian Language. In this respect, the relational database *dex* on-line was designed, and a system for converting the dictionary from MS Word format to JSON format was developed for subsequent uploading directly to the platform, etc.

In order to safeguard the dialectological archive, the program for transliterating dialect texts published in Cyrillic characters into the Latin alphabet was developed and tested, where dialectologists contributed by specifying and illustrating with examples the rules for transliterating Cyrillic characters into the Latin characters; developing samples of dialect texts transliterated into the Latin alphabet; checking and preparing dialect texts for automatic transliteration, etc.

**Keywords:** *LogosPlus*, philological resources, online environment.

## 1 Introduction

The universal world, the totality of physical and spiritual “things”, is perceived (subjectively) as reality, which man strives to know or recognize. The information society addresses gnoseological dimensions, centered on the concepts *res*, *rei*, *f* “thing”, *homo*, *înis*, *m* (*individuum*, *i*, *n*), *cogno-*

*sco, ere* “to know”, *cognitionis* “knowledge” in the process of understanding and perceiving “things”, including specialized things, which form the domain terminology.

In conditions of information globalization, in all fields of activity, new perspectives are offered to approach the flow of information, aiming at systematization and accessibility of information, increasing its socio-cultural and economic impact.

## 2 Philological Resources and Informatics

The philological sciences involve linguistic-terminological and literary studies. The national project “Scientific valorization of the national linguistic heritage in the context of European integration”, State Program (2020-2023), implemented in partnership with the Institute of Mathematics and Computer Science “Vladimir Andrunachevici” represents an opportunity to valorize philological resources in the online environment.

The research focus of the project is on the Romanian language, including the revaluation of the static element of the Romanian language (valorization of old Romanian texts; safeguarding the dialectological archive; digitization and transliteration into Latin alphabet of dialect texts, etc.) and the elucidation of the dynamic element of the Romanian language (dictionaries, language cultivation elements, etc.).

In this context, a portal of philological resources *LogosPlus* ([logosplus.ifr.md](http://logosplus.ifr.md)) was created in collaboration with the Institute of Mathematics and Informatics “Vladimir Andrunachevici”, which aims to be a promoter of national values (language and culture) and is intended for the general public.

The design and development of this *website* was an *a priori* condition in the process of implementing the above project, as new technologies become indispensable in the cultivation of language and literacy skills.

## 3 LogosPlus: Portal of Philological Resources

*LogosPlus*, accompanied by the motto *More than words*, is the philological resources portal of the Bogdan Petriceicu-Hasdeu Institute of Romanian Philology and is meant to be a promoter of national values (language and culture); to reflect the results obtained; to promote the image of the Institute; to elucidate the scientific activity within projects, etc. and is intended for the general public.

Thus, the sections *Proiectele noastre*, *Rezultate științifice*, *Echipa* (*Our Projects*, *Scientific Results*, *Team*) show data on the projects implemented within the Institute and the researchers involved in this activity. The Institute's image is promoted through Video Philological Portraits, which include 20 well-known personalities employed by the Institute (Fig. 1).



Figure 1. *LogosPlus*. Portal of philological resources. Bogdan Petriceicu-Hasdeu Institute of Romanian Philology

#### 4 On-line Environment and the Cultivation of the Romanian Language

From this perspective, the norm and aesthetics of the Romanian language are promoted through the Romanian Word etymology and cultivation tablets, distributed in the sections: *Fonetică, ortografie, punctuație, Morfologie, Lexic și semantică, Sintaxă* (Phonetics, Orthography, Punctuation, Morphology, Lexicon and Semantics, Syntax), and through the etymology and language cultivation video tablets (2-3 minutes long).

The *Articole recente* (Recent Articles) and *Apariții editoriale* (Editorial Appearances) sections aim to disseminate the scientific results obtained in the projects and to popularize them; these results are open access.

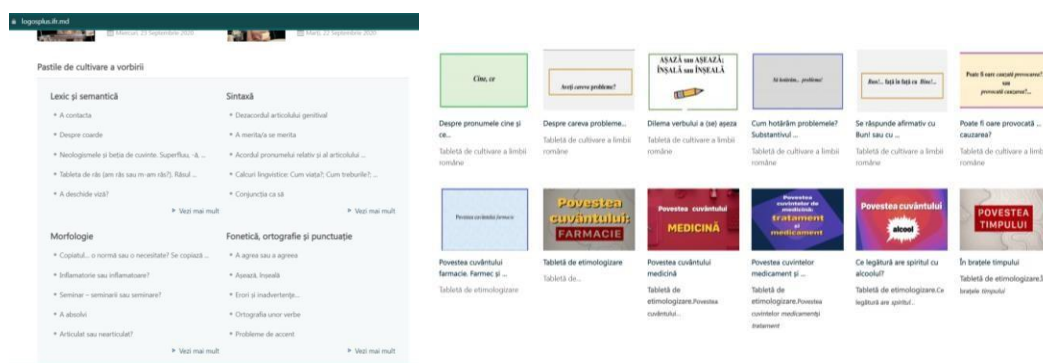


Figure 2. Tablets for etymologizing and cultivating the Romanian language

#### 5 Digitization and Backup Services for the Dialect Archive and Online Environment

Inter-institutional collaborations (Institute of Mathematics and Computer Science “Vladimir Andrunachievici”) focused on the sub-themes: *Research and evaluation of the lexicon of the Romanian language in a general Romanian context* – objective: creation and development of a virtual library of dictionaries of the Romanian language, and *Evaluation of old manuscript texts. Safeguarding the onomastic and dialectological treasury* – objective: development and testing of the program for transliteration into Latin alphabet of dialect texts, published in Cyrillic characters; development of the preliminary digital version of dialect texts in Latin characters (volume I); primary revision of dialect texts, transliterated into Latin alphabet. Establishment of the rules of literalized transcription of dialect texts; beginning of the process of transcribing dialect texts into literalized form.

Within the first sub-theme, the *dex.ifr.md* platform was created and developed (in collaboration with the Institute of Mathematics and Computer Science “Vladimir Andrunachievici”); the relational database *dex-online* with tables: *dicționare, cuvinte, utilizatori* (dictionaries, words, users) was designed; the following functionalities were developed: adding and editing words with attributes: *cuvânt-titlu, descriere lexicografică, sinonime, antonime, paronime, alte informații despre cuvânt* (word-lemma, lexicographic description, synonyms, antonyms, paronyms, other information on the word); adding and editing dictionaries with attributes: *titlul dicționarului, descrierea, instituția autorilor, etc.* (dictionary title, description (authors and content), authors' institution, etc.). Other opportunities: registration and authentication of lexicographers; word search by word-lemma; tabular display of words; random word-of-the-day selection and display on the user page, etc. (*dex.ifr.md*); a system for converting the dictionary from MS Word format to JSON format was developed for subsequent uploading directly to the platform: *Dicționarul de sinonime al limbii române* (Dictionary of Synonyms of the Romanian Language, over 10,000 entries) and *Dicționarul explicativ uzual al limbii române* (Usual Explanatory Dictionary of the Romanian Language, cca 25,000 entries) were uploaded.



Figure 3. Platform *dex.ifr.md*. *Dicționar de sinonime* (Dictionary of synonyms)

The functionality to upload a letter to the DEX online platform has been developed. This functionality has been integrated into the platform and can be accessed at: <https://dex.ifr.md/litera>. The following dictionaries are published on the platform *dex.ifr.md*: *Dicționar explicativ uzual* (Usual explanatory dictionary); *Dicționar de cuvinte și sensuri noi* (Dictionary of new words and meanings).

Therefore, section *Dicționare online* (Online Dictionaries, LogosPlus website) includes the platform for managing the electronic dictionaries (*dex.ifr.md*): *Dicționarul de sinonime al limbii române* (Dictionary of Synonyms of the Romanian Language, over 10 000 entries), *Dicționarul explicativ uzual al limbii române actuale* (Usual Explanatory Dictionary of the Romanian Language, over 30 000 entries), *Dicționar de cuvinte și sensuri noi* (Dictionary of New Words and Meanings), signed by a team of authors. In the future, *Dicționarul de antonime* (Dictionary of Antonyms) by Gheorghe Druță and *Dicționar poliglot de aforisme și expresii latine* (Polyglot Dictionary of Latin Aphorisms and Expressions) will be published.

Section *Biblioteca online* (Online Library) contains scholarly works available to the general public. The online library is intended to include the Institute's works in electronic format, which will be available to the general public.

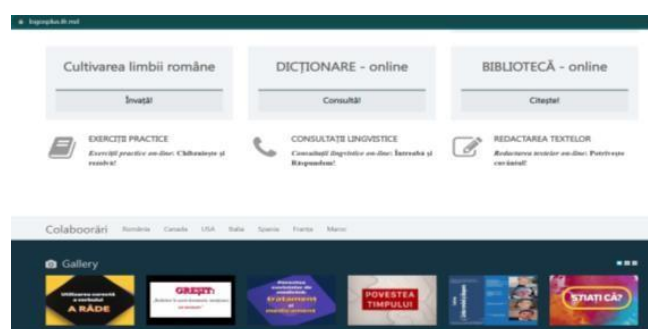


Figure 4. Section *Dicționare online* (Online dictionaries, LogosPlus website)

In the near future the following services will be available: *Consultații Lingvistice online: Întrebă și Răspundem!* (Online Language Consultations: Ask and Answer!), where various questions can be asked to the researchers involved in the project in order to obtain concrete answers; section *Exerciții practice online: Chibzuiește și Rezolvă!* (Online Practical Exercises: Think and Solve!) will offer a wide range of practical exercises for the general public (pupils, students, teachers, etc.), through which they will check and consolidate their theoretical (language) knowledge.

At the level of the second sub-theme, a program for transliteration into the Latin alphabet of dialectal texts, published in Cyrillic characters, was developed and tested, with dialectologists

contributing by a) specifying and illustrating with examples the rules of transliteration of Cyrillic characters into Latin characters; b) elaborating samples of dialectal texts transliterated into the Latin alphabet; c) checking and preparing dialectal texts for automatic proof transliteration; d) checking and correcting dialectal texts after automatic proof transliteration; e) reviewing the rules of transliteration of Cyrillic characters into Latin characters after testing the automatic transliteration program. The preliminary digital version of the dialectal texts with Latin characters (Volume I) has been prepared; the primary revision of *Textele dialectale* (*Dialectal texts*) transliterated into the Latin alphabet (Volume I) has been carried out. The rules of literalized transcription of dialect texts were established and the literalization of dialect texts (manuscript version) was carried out.

Then, the digitization work over *Texte dialectale* (*Dialectal Texts*, Volume 1) involved the following actions:

- a) scan (Fig. 5);
- b) preparation for electronic conversion/pre-processing using *ScanTailor* ("cleaning" of anything that might prevent electronic conversion: underlining, links between words, etc.) (Fig. 6);

СОЛОТВИНО (2)

Обичеурь ла нунтэ

Май ынтий мэрг батрыний ла фаты а пецы... Ши даке ынтра,  
зык... тречиц ши ходиниц ла ной. Дак не-ам пус а ходини,  
апи ынтребэм: ам аузыт, к-авец о фаты де мэритат. Пециторий  
ынтрабы эстрја. Чи-й дай д"е эстрја ла фаты? Кари чј-о  
авут о дат: кари о дат бой, вач<sup>1</sup>, пэмынт - аму ку сынт д"е че-  
ли. Аму он дэ: перн"е, бутур<sup>1</sup>, бан<sup>1</sup> шы кари чи ау, ачеја дау.  
Дупа чи с-о афлат бин"е ку пецитул, ку эстрја, апой с-о грэ-  
ит ш-о мэрс ла попа а да мына. Дупэ чј-о да мына ла попа, о  
ин"ит акасы, о молэтуит<sup>2</sup>, о фэкут молочјаг<sup>3</sup>. Апой а доузы  
мэрг ла кэнцэлэрийи<sup>4</sup> а бага зылили<sup>5</sup>. Дупэчи о багат зылили,

Figure 5. Stage 1. Scanned text

Обичеурь ла нунтэ

Май ынтий мэрг батрыний ла фаты а пецы... Ши даке ынтра,  
зык... тречиц ши ходиниц ла ной. Дак не-ам пус а ходини,  
апи ынтребэм: ам аузыт, к-авец о фаты де мэритат. Пециторий  
ынтрабы эстрја. Чи-й дай д"е эстрја ла фаты? Кари чј-о  
авут о дат: кари о дат бой, вач<sup>1</sup>, пэмынт - аму ку сынт д"е че-  
ли. Аму он дэ: перн"е, бутур<sup>1</sup>, бан<sup>1</sup> шы кари чи ау, ачеја дау.  
Дупа чи с-о афлат бин"е ку пецитул, ку эстрја, апой с-о грэ-  
ит ш-о мэрс ла попа а да мына. Дупэ чј-о да мына ла попа, о  
ин"ит акасы, о молэтуит<sup>2</sup>, о фэкут молочјаг<sup>3</sup>. Апой а доузы  
мэрг ла кэнцэлэрийи<sup>4</sup> а бага зылили<sup>5</sup>. Дупэчи о багат зылили,

Figure 6. Stage 2. Processed text

c) conversion to electronic format (optical character recognition, OCR) – with a special program (developed by scientific researchers at the Institute of Mathematics and Computer Science “Vladimir Andrunachievici”) (Fig. 7);

d) checking and correcting the text after conversion to electronic format (entering/correcting signs not recognized by the computer) (Fig. 8);

СОЛОТВИНО (2)

Обичеурь ла нунтэ

Май ынтий мэрг батрыний ла фаты а пецы... Ши даке ынтра,  
зык... тречиц ши ходиниц ла ной. Дак не-ам пус а ходини.  
апи ынтребэм: ам аузыт, к-авец о фаты де мэритат. Пециторий  
ынтрабы эстрја. Чи-й дай д"е эстрја ла фаты? Кари чј-о  
авут о дат: кари о дат бой, вач<sup>1</sup>, пэмынт - аму ку сынт д"е че-  
ли. Аму сы дэ: перн"е, бутур<sup>1</sup>, бан<sup>1</sup> шы кари чи ау, ачеја дау.  
Дупа чи с-о афлат бин"е ку пецитул, ку эстрја, апой с-о грэ-  
ит ш-о мэрс ла попа а да мына. Дупэ чј-о да мына ла попа, о  
ин"ит акасы, о молэтуит<sup>2</sup>, о фэкут молочјаг<sup>3</sup>. Апой а доузы  
мэрг ла кэнцэлэрийи<sup>4</sup> а бага зылили<sup>5</sup>. Дупэчи о багат зылили,

Figure 7. Stage 3. Converted text

СОЛОТВИНО (2)

Обичеурь ла нунтэ

Май ынтий мэрг батрыний ла фаты а пецы... Ши даке ынтра,  
зык... тречиц ши ходиниц ла ной. Дак не-ам пус а ходини,  
апи ынтребэм: ам аузыт, к-авец о фаты д"е мэритат. Пециторий  
ынтрабы эстрја. Чи-й дай д"е эстрја ла фаты? Кари чј-о  
авут о дат: кари о дат бой, вач<sup>1</sup>, пэмынт - аму ну сынт д"е че-  
ли. Аму сы дэ: перн"е, бутур<sup>1</sup>, бан<sup>1</sup> шы кари чи ау, ачеја дау.  
Дупа чи с-о афлат бин"е ку пецитул, ку эстрја, апой с-о грэ-  
ит ш-о мэрс ла попа а да мына. Дупэ чј-о да мына ла попа, о  
ин"ит акасы, о молэтуит<sup>2</sup>, о фэкут молочјаг<sup>3</sup>. Апой а доузы  
мэрг ла кэнцэлэрийи<sup>4</sup> а бага зылили<sup>5</sup>. Дупэчи о багат зылили,

Figure 8. Stage 4. Verified text

e) development and testing of the program for transliteration into the Latin alphabet of dialectal texts published in Cyrillic characters;

f) preparing the texts for automatic transliteration (putting accents on all accented words (the accents were either not recognized at the conversion stage or appeared in a different form than necessary: ле'мнили instead of лѐмнили) and correcting mistakes in the original) (Fig. 9);

g) automatic transliteration of texts using specially created software (developed by scientific researchers at the Institute of Mathematics and Computer Science “Vladimir Andrunachevici”) (Fig.10);

h) checking and correcting after automatic transliteration (correcting mistakes, restoring underlines and word links) (Fig.11).

## СОЛОТВИНО (2)

solotvino (2)

Обичеюрь ла нунтэobiiceiuri la nuntă

Май ынтыгый мэрг батрыний ла фаты а пецы... Ши дакэ ынтрэ, зык: ...трэчиц ши ходиниц ла ной. Дак не-ам пус а ходини, апи ынтрэбэм: ам аузыт к-авец о фаты д"е мэритат. Пециторий ынтрэбэы зэстрја. Чи-й дай д"е зэстри ла фаты? Кари ч-о авут о дат: кари о дат бой, вач<sup>1</sup>, пэмынт - амú ну сынт д"е чé-ли. Амú сы дэ: пэрн"е, бұтур<sup>1</sup>, бан<sup>1</sup> шы кари чи ау, ачэја даў. Дұпа чи с-о афлат бин"е ку пецитул, ку зэстрја, апои с-о грэ-йт ш-о мэрс ла póпа а да мýна. Дұпэ ч-о да мýна ла póпа, о ин"йт акасы, о молэтуйт<sup>2</sup>, о фэкұт молочјаг<sup>3</sup>. Апој а дөү-зы мэрг ла кэнцэлэрийи<sup>4</sup> а багá зылили<sup>5</sup>. **Дупэ чи** о багáт зылили,

Figure 9. Stage 5. Text prepared for transliteration.

mai întîi mărg batrîniî la fátă a peţă... şi dăcă întră/ zîc: ...tréçiţ şi hodinîţ la noi// dac-ne-am pus a hodinî/ apî întrebăm: am auzîţ c-avet o fátă d'e măritát// peţitóriî întrăbă zăstrja// „ci-î dai d'e zăstri la fátă? cári c-o avút o dat: cári o dat boj/ vác/ pămînt - amú nu sînt d'e cê-li// amú sâ dă: pérne/ bútur<sup>1</sup>/ bán! şi cári cî au/ acéja dău// dúpa cî s-o aflát bîne cu peţitul/ cu zăstrja/ apoi s-o gră-ít ş-o mǎrs la pópa a da mîna// dúpă c-o da mîna la pópa/ o inhít acásă/ o molătuít<sup>2</sup>/ o făcút moločág<sup>3</sup>// apói a dөu-ză mărg la cǎnţălărîi<sup>4</sup> a bagá zîlilil<sup>5</sup>// după cî o bagát zîlilil/

Figure 10. Stage 6. Transliteration of the text

## Solotvino (2)

Obiceiuri la nuntă

mai întîi mărg batrîniî la fátă a peţă...// şi dăcă întră/ zîc: „...tréçiţ şi hodinîţ la noi”// dac-ne-am pus a hodinî/ apî întrebăm: „am auzîţ c-avet o fátă d'e măritát”// peţitóriî întrăbă zăstrja// „ci-î dai d'e zăstri la fátă?”// cári c-o avút o dat: cári o dat bói/ vác/ pămînt – amú nu sînt d'e cê-li// amú sâ dă: pérne/ bútur<sup>1</sup>/ bán! şi cári cî au/ acéja dău// dúpa cî s-o aflát bîne cu peţitul/ cu zăstrja/ apoi s-o gră-ít ş-o mǎrs la pópa a da mîna// dúpă c-o da mîna la pópa/ o inhít acásă/ o molătuít<sup>2</sup>/ o făcút moločág<sup>3</sup>// apói a dөu-zî mărg la cǎnţălărîi<sup>4</sup> a bagá zîlilil<sup>5</sup>// după cî o bagát zîlilil/

Figure 11. Stage 7. Verified text

The planned finished product is the editing of *Textelor dialectale* (Dialect Texts, Volume 1) and publication of the *LogosPlus* website in the section *Biblioteca online* (Online Library).

## 6 Conclusions

The present article is based on the achievements of the national project “*Scientific evaluation of the national linguistic heritage in the context of European integration*”, State Program of 2020-2023, implemented in partnership with the Institute of Mathematics and Computer Science “Vladimir Andrunachievici”.

Collaboration with the Institute of Mathematics and Computer Science “Vladimir Andrunachievici”, the implementation of information technologies offered new perspectives for the organization of philological resources, and also for the promotion of the image of the Institute of Romanian Philology “Bogdan Petriceicu-Hasdeu”. At the same time, new perspectives of accessibility of the general public to the linguistic and literary sources of the Romanian language are opened. And, in general, the new digital technologies make the access, storage, and transmission of information on philological resources easier and more accessible. We also mention the procedure for safeguarding the Institute's Dialectological Archive by transliterating texts from the Cyrillic alphabet into the Latin alphabet. Digitized philological resources represent socio-cultural values and are key information resources, implementing long-term objectives: development, cultivation, and safeguarding of linguistic and literary sciences in the Republic of Moldova.



# THE FIRST SLAVONIC-ROMANIAN LEXICONS (17<sup>TH</sup> CENTURY): A DIGITAL COMPARATIVE EDITION AND HOW TO USE IT

MIHAI ALEX MORUZ<sup>1,2</sup> AND MĂDĂLINA UNGUREANU<sup>1</sup>

<sup>1</sup>*Department of Social Sciences and Humanities, Institute of Interdisciplinary  
Research, "Alexandru Ioan Cuza" University, Iași, Romania,  
andreea.ungureanu@uaic.ro*

<sup>2</sup>*Faculty of Computer Science, "Alexandru Ioan Cuza University", Iași, Romania*

## Abstract

This paper presents the results of a retrodigitization project of the oldest Romanian bilingual dictionaries (six Slavonic-Romanian dictionaries of the 17<sup>th</sup> century), the purpose of which was the creation of an electronic comparative edition. The paper begins with an exposition of the motives for this choice of encoding of linguistic content; we then follow with a brief description of the processing steps necessary for the creation of the corpus, focusing on the features of the query interface and the usefulness of the database. Since the main utility of the corpus will regard comparative studies of the various lexicons in question, in many possible combinations, a robust and complex interrogation method has been created, and this tool will also be discussed, in detail, as a collection of potential use cases for future researchers. This large lexicographical corpus for old Romanian is also a useful tool for Romanian lexicography; an integrated search engine will allow for searching over the gloss text, thus aiding in possible identifications of first attestations and senses not found in current dictionaries.

## 1 The Comparative Digital Edition of the First Slavonic-Romanian Lexicons (eRomLex). Premises

Although the beginnings of Romanian printed lexicography cannot be traced earlier than the end of the 18th century (Seche, 1966: 16), the first activities in this field can actually be traced as early as the first stage of writing in Romanian, according to the natural evolution glosses → glossaries → bilingual lexicons that characterizes the development of lexicography in general (Adamska-Sałaciak, 2014: 1) and is based on the confrontation between the language of culture and the vernacular language that tends to become, in turn, the vehicle of culture. The first Romanian lexicons are bilingual (Romanian with Latin, Slavonic, and, less frequently, Hungarian, Italian or Greek, see Seche, 1966: 7-13), mostly of the L2-L1 type (Slavonic-Romanian, Latin-Romanian, etc.). In the first century of Romanian lexicography, most dictionaries were Slavonic-Romanian; paradoxically, these have also been the least studied. The lack of interest in these works (with the exception of the Romanian-Latin dictionary issued in cca 1650) is most likely due to a combination of factors, including the fact that they were preserved in manuscript form (having therefore a rather restrained circulation and lacking impact on the overall cultural activity), that they were considered mere imitations of the source after which they were compiled (see, for instance, Gînsac and Ungureanu, 2018). There are six L1-L2 (Slavonic-Romanian) lexicons, the first dating from 1649 and the others from the period between 1660 and 1690<sup>1</sup>, which were compiled following the model of a Slavonic-Ruthenian lexicon published in Kyiv in 1627 (see, in this respect, Gînsac et al., 2022: 223). The fact that these works are related has been discussed since

---

<sup>1</sup> Lex.Staicu: 1660-1670 (Mihăilă, 1972: 313)/ 1667-1669 (Strungaru, 1966); Lex.1348: 1683 (Ștrempel, 1978: 300)/ 1678 (Mihăilă, 1972: 313); Lex.3473: 1672-1673; Lex.Pet.: ante 1693 (Mihăilă, 1972: 315); Lex.Mosc.: end of the 17<sup>th</sup> century-beginning of the 18<sup>th</sup> century (Ciobanu, 1914). For the sigla, see (Gînsac et al., 2022).

they started to be researched, but their filiations have remained mere subjects of debates, since each of these lexicons displays innovative elements in relation to all the others (omissions or additions), so that comparative editing has been regarded as the most efficient way to approach them. Undoubtedly, an edition in classical format would have been extremely difficult to compile in this case, for several reasons: the order of the entries differs from one lexicon to another; the inventory of entries does not overlap either with that of the model or with that contained in each separate lexicon; there are numerous entries added in each lexicon, which do not necessarily coincide with the entries added in the other lexicons. Considering all these aspects, an edition allowing their parallel display at the entry-level was considered the best option for their study, despite the fact that the linear image of the manuscripts is obscured in this way<sup>2</sup>. Therefore, the purpose of editing the lexicons, besides the fact that it facilitates access to them, which is in itself of great importance (the lexicons represent a significant part of the Romanian written cultural heritage, as they constitute the most consistent part of the Romanian lexicography of the 17<sup>th</sup> century), was to facilitate access to their linguistic content and, by the various possibilities of displaying the information, to their comparative study. In this regard, the display interface was an essential part of achieving this goal.

## 2 Compiling the Electronic Corpus

As the corpus development stage has been extensively presented in (Moruz and Ungureanu, 2021) and (Gînsac et al., 2022), we shall provide only a brief presentation below.

### 2.1 Filing the Entries

The lexicons were filed manually. Because of the specific layout at the page level (the linguistic material is displayed in columns, in some cases, there are 4 columns on a page), and equally because of the graphic features (cursive writing, with many overwritten letters and abbreviations, diacritics; in the case of Lex.Staicu, marginal and interlinear notes), the results we obtained testing the Transkribus automatic handwritten text recognition software (<https://readcoop.eu/transkribus/>) were not satisfactory; manual correction would have been a lot more time consuming and therefore less efficient.

Filing was carried out in a standard format designed especially for the purpose of the project; each entry was filed separately, the categories extracted being: headword (in Slavonic, in Cyrillic font); position of the word within the manuscript (lexicon name, page, column, entry number in the respective column - important for the possibility of page-level display, including in parallel, and for comparative observation of the order of entries); Romanian definition (transcribed in the Latin alphabet); (philological) remarks.

Our aim was to align the Romanian lexicons at the entry level, both with regard to each other and to the corresponding entry in the source lexicon, Lex.Ber. Therefore, at the filing stage, in order to facilitate the alignment, we added to the information mentioned above the Lex.Ber. entry, to which the respective entry in the Romanian lexicon corresponds (where applicable): the headword and the localization elements (column and entry number). The Ruthenian definitions from Lex.Ber. have been translated into Romanian in order to serve as an auxiliary element for the translation analysis of the Romanian definitions, yet this information will not be displayed.

### 2.2 Indexing

Slavonic headwords, both from Lex.Ber. and Romanian lexicons have been collected in compliance with their respective graphic peculiarities. In the case of Lex.Ber, the diacritical marks were

---

<sup>2</sup> For the pros and cons of using electronic tools in textual criticism (with application to the Bible), see (Tov 2021). As far as electronic editing of premodern Romanian lexicographical works is concerned, there is already the precedent of the Buda Dictionary, 1825 (available at: <https://doi.org/10.26424/lexiconuldelabuda>; last access: 09-10-2022).

collected as such; for the Romanian lexicons, following the observation that diacritics were mostly used for the sake of tradition, and hardly carried any phonetical value, we have recorded only the payerok, which helps to disambiguate certain forms, and the kendima, used sometimes in Lex.Staicu with the value /i/ in the final position. Without any doubt, it would have been ideal for us to be able to record the other diacritical marks as well, but this would have been time-consuming and relatively irrelevant for the reasons provided above, so this operation was postponed for a later stage, in order to provide a faithful rendition of the Slavonic words, as they were used by the Romanian scholars who compiled the lexicons. We have, however, rendered as such the alternations in the case of the letters with the values /i/ (и, ї, ѣ, v, ѡ), /u/ (ѹ, оѹ) and /o/ (o, ѡ), even though the choice of one particular character bears no functional value in Romanian texts, as they do in the Slavonic writing. Subsequently, in order to facilitate alignment, a simplified and unifying form of Slavonic word spelling (without diacritics, with the graphic variants reduced to one, replacing the superscripts with their standard variants) was constructed, based on a standard system of equivalences. Given the graphical differences between the Slavonic headwords in the Romanian lexicons, this standardized form allowed fairly successful automatic alignment of entries in the Romanian lexicons for which there is no equivalent in Lex.Ber., based on the degree of graphic equivalence.

### 3 Display and Search Interface

In creating the search interface, we considered, first of all, the type of texts and the purpose of the edition. Thus, on the one hand, the display possibilities should meet the requirements of Romanian philology (including the philological observations such as scribal errors, omissions, overwritten characters, particular signs, different hands, etc.); on the other hand, we needed an interface that would allow searching according to multiple query criteria. The problem encountered here, though, is that a single search interface will either be too large to be useful (seeing as the number of features of interest is very large), or will not offer all of the possible manners of searching that could be required by the users. Because of this, we have decided to create several search interfaces, according to the use cases which we have deemed to be most common.

#### 3.1 Scenario 1 – General Use

One of the most common scenarios for the use of the electronic version of the lexicons is the search for a particular word in Slavonic, generally in accordance with the following steps:

- List all the entries in the dictionaries, aligned at the headword level, and showing the Slavonic version for each manuscript, where available (see Fig. 1). Search over the normalized form of the headword (the normalized form of the headword is described in Section 2.2).
- Filter the results by typing parts of words in any of the columns. Multiple simultaneous filters are possible (i.e. filtering on multiple lexicons at once).
- Sort the results, in ascending or descending order, by any column.
- Access the link towards an entry in the list for details of the entry.

This particular search interface provides access, for searching, filtering, and ordering, to all of the headwords in the six Romanian – Slavonic lexicons and in Lex.Ber. in an eight-column view, providing the simplified version of the Slavonic word (without diacritics, superscripts, etc.) and the aligned headwords in the lexicons (where available).

| Word                  | Lex.Ber               | Lex.Mosc  | Lex.3473  | Lex.Pet | Lex.Staicu | Lex.1348 | Lex.Mard             |
|-----------------------|-----------------------|-----------|-----------|---------|------------|----------|----------------------|
|                       |                       |           |           |         |            |          |                      |
| води*                 |                       | вѣды*     |           |         |            |          |                      |
| водѣ                  |                       |           |           |         | вѣды       |          |                      |
| води                  |                       |           |           | вѣды    |            |          |                      |
| воевода               | вѣвѣда                | вѣвѣда    | вѣвѣда    | вѣвѣда  | вѣвѣда     | вѣвѣда   |                      |
| вождѣ                 | вѣждѣ                 | вѣждѣ*    | вѣждѣ     | вѣждѣ   | вѣждѣ      | вѣждѣ    | вѣждѣ                |
| возатай               | вѣзатай               |           |           |         |            | вѣзатай  |                      |
| возатай,<br>вѣсадникѣ | вѣзатай,<br>вѣсадникѣ | вѣсѣдникѣ | вѣсѣдникѣ | вѣзатай | вѣсѣдникѣ  |          | вѣзатай<br>вѣсѣдникѣ |

Figure 1. A view of the list of entries

| Word                | Lex.Ber          | Lex.Mosc | Lex.3473 | Lex.Pet | Lex.Staicu | Lex.1348 | Lex.Mard |
|---------------------|------------------|----------|----------|---------|------------|----------|----------|
| в%                  | в%               | в%       | в%       | в%      | в%         | в%       | в%       |
| вѣссонѣ,<br>перфира | вѣссонѣ, перфира | вѣссонѣ* | вѣссонѣ  | вѣссонѣ | вѣссѣ      | вѣссѣ    | вѣссонѣ  |
| ващій               | вѣщій            | вѣщій    | вѣщій    | вѣщій   | вѣщій      | вѣщій    | вѣщій    |
| вазаю               | вѣзѣю            | вѣзѣю    | вѣзѣю    | вѣзѣю   | вѣзѣю      | вѣзѣю    | вѣзѣю    |
| вѣщание             | вѣщѣнїе          | вѣщѣнїе  | вѣщѣнїе  | вѣщѣнїе | вѣщѣнїе    | вѣщѣнїе  | вѣщѣнїе  |
| вѣтрило             | вѣтрило          | вѣтрило  | вѣтрило  | вѣтрило | вѣтрило    | вѣтрило  | вѣтрило  |
| вѣтий               | вѣтїи            | вѣтїи    | вѣтїи    | вѣтїи   | вѣтїи      | вѣтїи    | вѣтїи    |
| вѣси                | вѣсен            | вѣсен    | вѣсен    | вѣсен   | вѣсен      | вѣсен    | вѣсен    |

Figure 2. All the words starting with the letter в, which appear in all lexicons, sorted in descending order by simplified headword

As can be seen from the figures above, the main search interface allows for searches of words, or parts of words, on any of the lexicons. It is, however, recommended that the search be carried out on the “Word” field, as the particular headwords in the lexicons may be written using diacritics, superscripts, or other special symbols, which would yield incorrect results. Also, joint searches can be carried out, as shown in Fig. 2, which add further restrictions on the result list. The example search given in Fig. 2 is designed to return all the words starting with the letter в, which appear in all lexicons.

If the user wants to see a detailed entry from the list, they can follow the link provided in the first column, which will lead to a page similar to that in Fig. 3, where all of the specific details are available.

In order to more precisely filter and search the lexicon, the user can make use of SQL wildcard characters, as shown in Fig. 2. The currently supported special characters for regular expressions are “%” (any string of length 0 or more in the search pattern), “\_” (any single character), “[ ]” (any single character in the specified range) and “[^]” (any single character not in the specified

range). For example, the query “a\_” would return all the words starting with the letter “a” which have a length of at least 2 characters.

| Source Volume | Head Word        | Page | Column | Number | Gloss                                   | Observations                               |
|---------------|------------------|------|--------|--------|-----------------------------------------|--------------------------------------------|
| Lex.Ber.      | Бѣссонъ, перфіра | 382  |        | 5      |                                         |                                            |
| Lex.Pet.      | Бѣссонъ          | 15v  | I      | 11     | caftanu                                 |                                            |
| Lex.Staicu    | Бѣссѡ            | 61r  | II     | 2      | caftan                                  |                                            |
| Lex.Mosc.     | Бѣссонъ*         | 39v  |        | 15     | caftan                                  | *Scris în continuarea intrării precedente. |
| Lex.Mard.     | Бѣссонъ          | 59   |        | 5      | cufturne, pînză suptîre                 |                                            |
| Lex.3473      | Бѣссонъ          | 39v  |        | 8      | caftan, sau cofturie, sau pînză suptîre |                                            |
| Lex.1348      | Бѣссонъ          | 12r  | I      | 4      | caftan.                                 |                                            |

Figure 3. A detailed view of the entry *Бѣссонъ, перфіра*

Since the possible number of search criteria is very large, as shown in Fig. 3 above, an interface allowing for all of them at once would be much too difficult to see (because of the numerous columns), let alone successfully use. We intend to also provide an on-screen Cyrillic keyboard for ease of use.

The search for a particular Slavonic word may be useful for translation comparative studies, for determining meanings of specific Romanian words; the type of search described above is also useful for studying the relations between the lexicons or between them and their Ruthenian model.

### 3.2 Scenario 2 – Searching for Romanian Words

The second scenario taken into account regards the search for Romanian words in the lexicons, either as translation equivalents to a given Slavonic word or as individual occurrences. This might be necessary in the course of identifying the usage of particular words in the Romanian language of the 17<sup>th</sup> century, for the purposes of determining attestations or variants, for extracting senses for Romanian words on the basis of the equivalence with some particular Slavonic word, etc.

The second use case interface allows for multicriterial searching and filtering over the contents of the glosses of all the Slavonic-Romanian lexicons, and for searching over the normalized headword form. The regular expression-based search, as described above, can also be used in this interface, allowing for flexible and complex queries over multiple fields and for the sorting of the results.

Fig. 4, given below, describes an example of a gloss query within the interface for the word “bun” in the Lex.Mosc. lexicon, which returns all of the entries that contain the search sequence in the glosses of Lex.Mosc. We must note that, in the form that this query is given, the sequence needs not be the exact word that has been searched for, but rather any sequence of characters which contains the word (i.e., it might be preceded or followed by other characters). For the pur-

poses of this example, we have ordered the resulting glosses in descending order, by the Unicode code value of the symbols.

This viewing mode is particularly useful for studying the influences that the lexicons have had on each other, as the comparison of the glosses becomes much more straightforward. For instance, from the results given in Fig. 4, one can notice that the entry **БЛОПОЛУЧЕНИЕ** shows a relation between Lex.3473 and Lex.1348, also between Lex.Mard., Lex.Mosc., and Lex.Pet., while Lex.Staicu provides a different translation. Also, the entry **БЛОПРИЯТНОЕ ВРЕМЯ** shows a certain degree of relatedness between Lex.Staicu, Lex.Mosc., Lex.3473, Lex.Pet., and Lex.1348, while it is omitted in Lex.Mard., confirming our hypothesis that the first five dictionaries are modified copies of another one, which was lost.

| Word                  | Lex.Mosc                   | Lex.3473                    | Lex.Pet                       | Lex.Staicu                           | Lex.1348                     | Lex.Mard                              |
|-----------------------|----------------------------|-----------------------------|-------------------------------|--------------------------------------|------------------------------|---------------------------------------|
|                       | bun                        |                             |                               |                                      |                              |                                       |
| БЛОПОЛУЧЕНИЕ          | întimplare bună noroc      | întimplare /5v bună, noroc  | întimplare bună. norocu       | întimplare bună. noroc               | întimplare bună noroc        | bună întimplare noroc                 |
| БЛОГСТЫНА             | îndurare bună              | îndurare bună               | îndurare: bună                | bunăitatea îndurarea                 | îndurare bună                | mila. blîndeătele, milă de la inimă   |
| БЛОДЕИСТВИЕ           | zilele bune                |                             | zile bune                     | traiu bun                            |                              | zile bune                             |
| ВЕДРО                 | vreme bună, senin          | vreame bună, senin          | vreame bu[nă]                 | vreme bun[ă], senin                  | vreame bună senin.           | sănin                                 |
| БЛОПРИЯТНОЕ ВРЕМЯ     | vreame de bună priimire    | vreame bună, priimire       | vreamea de bună priimire      | vreamea cea de bine /48r/II priimită | vreame de bună priimire      |                                       |
| БЛОГАЗЫЧЬЕ: БЛОГРЪЧЬЕ | vorbire bună graiu deşchis | vorbire bună, graiu deşchis | vorovire bună. graiu deşchisu | vorbire bună graiu deşchis           | vorobire bună graiul deşchis | bună vorbire, limbă uşoară spre graiu |

Figure 4. The gloss search interface

### 3.3 Scenario 3 – Searching over One of the Lexicons

Philological editing involves reproducing, as faithfully as possible, the image of the page in the edited text with its characteristics. Thus, the interface must allow the display of entries for each unique lexicon in the order in which they are listed in the respective lexicon. For this particular use case, it becomes necessary to create interrogation interfaces specific to each lexicon in part, so that a user will be able to perform queries such as that given in Fig. 5.

Fig. 5 presents the search interface specific to Lex.Mosc, but a similar interface is provided for all of the other lexicons taken into account (including Lex.Ber). It gives the user access to all the information available from the particular lexicon in question: Lex.Ber equivalent, if any; head-word; the page on which the entry is found in the original manuscript; in the case of manuscripts that contain two columns on each page, the column number is also available; the position of the word on the page in the manuscript, given as a number; the gloss for the particular entry; and philological observations. As is the case for the previous types of search interfaces, all the columns are searchable, filterable, and sortable.

For the example above, we have chosen to highlight the use case described at the beginning of this subsection; we have searched for all the words on page 35v of Lex.Mosc, and then we have ordered them according to their position on the original manuscript page. As can be seen, page 35v contains, among others, some entries which are not imported from Lex.Ber, but are newly added by the author; these entries are particularly interesting when studying the relationships be-

tween lexicons, as new words added in multiple manuscripts might point to either a common development or a common necessity in language.

| Lex.Ber                            | Lex.Mosc       | Page | Column | Position | Gloss                  | Obs                               |
|------------------------------------|----------------|------|--------|----------|------------------------|-----------------------------------|
|                                    |                | 35v  |        |          |                        |                                   |
| Въсѣвъръшѣнный                     | вѣсѣвъръшѣнный | 35v  |        | 1        | ce/ întru tot săvîrșit |                                   |
| Всѣвъръшѣннѣ,<br>всконѣннѣ, всѣкоу | вѣсѣвъръшѣннѣ  | 35v  |        | 2        | de tot desăvîrșit      |                                   |
| Въсѣвъръшѣннѣ                      | вѣсѣвъръшѣннѣ  | 35v  |        | 3        | în toț chipul          |                                   |
| Въсѣвъръшѣннѣ                      | вѣсѣвъръшѣннѣ  | 35v  |        | 4        | toate amele            |                                   |
| Въсѣвъръшѣннѣ, ѡй,<br>всѣвъръшѣннѣ | вѣсѣвъръшѣннѣ  | 35v  |        | 5        | întru toț întreg       |                                   |
| Въсѣвъръшѣннѣ                      | вѣсѣвъръшѣннѣ  | 35v  |        | 6        | să zîmbi a rîde        |                                   |
|                                    | вѣсѣвъръшѣннѣ  | 35v  |        | 7        | mă zîmbesc a rîde      |                                   |
|                                    | вѣсѣвъръшѣннѣ* | 35v  |        | 8        | să zîmbeaște a rîde    | *Scriș în<br>continuarea intrării |

Figure 5. Search interface for Lex.Mosc

### 3.4 Scenario 4 – New Search Types and Exporting of Results

Currently, our platform provides 9 separate predefined query interfaces: the general search engine for Slavonic words (see 3.1), the general search engine for Romanian words (see 3.2), and the specific search engines for each of the 6 Slavonic – Romanian bilingual lexicons and Lex.Ber. There exists, however, the possibility that some types of interrogation, which have not been predicted, might be necessary; fortunately, the system we have built for creating search interfaces allows for the quick creation of new interrogation solutions. In effect, any new type of search can be quickly created and integrated into the platform.

In terms of using the query results, we have also provided the possibility of exporting the search results in a CSV or Excel format.

## 4 Conclusions and Future Work

The comparative electronic edition of the first Romanian lexicons, possibly due to the common model, had as its starting point a series of hypotheses and questions regarding, on the one hand, the relationships between the lexicons and, on the other hand, those between the Romanian lexicons and the Ruthenian model. Thus, we have designed a tool that allows the processing of entries, both parallel and single lexicon display being possible. A minus of the edition is the lack of facsimiles, which is due to the short time we had for processing such a large amount of material and also to the problem of copyright. In the future, we are considering displaying one of the two dictionaries for which we have publishing rights.

Another advantage of electronic editing is constituted by the possibility of exploiting the material. Thus, the eRomLex database, by the way it is designed and by the search engines built, is a tool that can be used in lexicography and in works on the history of the Romanian literary language.

Finally, electronic editing also has the advantage of being improvable, augmented by the inclusion of other dictionaries (e.g., Macarie's lexicon, from the next century), and linked to other databases (dictionaries and electronic editions). To this end, we intend to provide the possibility of exporting the corpus of aligned lexicons in a TEI XML format, so that alignment to other resources will be more straightforward.

## Acknowledgements

This work was supported by a grant from the Ministry of Research, Innovation, and Digitization, CNCS/CCDI – UEFISCDI, project number PN-III-P1-1.1-TE -2019-0517, within PNCDI III.

## References

- A. Adamska-Sałaciak. (2014). Bilingual Lexicography: Translation Dictionaries. In P. Hanks and G. de Schryver (eds.). *International Handbook of Modern Lexis and Lexicography*, Berlin/Heidelberg, pages 1-11.
- Ş. Ciobanu. (1914). Славяно-румынскій словарь библиотеки Московскаго Общества Истории и Древностей No 240. *Русскій филологическій вестникъ*, 71(1):75-88.
- A.-M. Gînsac and M. Ungureanu. (2018). La lexicographie slavonne-roumaine au XVIIe siècle. *Zeitschrift für romanische Philologie* 134(3):845–876.
- A.-M. Gînsac, M. A. Moruz and M. Ungureanu. (2022). The first Romanian dictionaries (17th century). Digital aligned corpus. In *Dictionaries and Society. Proceedings of the XX EURALEX International Congress*. IDS-Verlag, Mannheim, pages 222-229.
- G. Mihăilă. (1972). Contribuții la studiul lexicografiei slavo-române din secolul al XVII-lea. In *Contribuții la istoria culturii și literaturii române vechi*, Editura Minerva, București, pages 307-324.
- M. A. Moruz, and Mădălina Ungureanu. (2021). Old Romanian Lexicons and Their Representation in DLR. In *Proceedings of the 16th International Conference "Linguistic Resources and Tools for Natural Language Processing"*, online, 13-14 December 2021, pages 93-102.
- M. Seche. (1966). *Schiță de istorie a lexicografiei române*, vol. 1: De la origini pînă la 1880. Editura Științifică, București.
- D. Strungaru. (1966). Începuturile lexicografiei române. *Romanoslavica*, 13:141-158.
- G. Ștrempel. (1978-1992). *Catalogul manuscriselor românești*, vol. 1 (1978), vol. 2 (1983), vol. 3 (1987), vol. IV (1992). Editura Științifică și Enciclopedică, București.
- E. Tov. (2021). Textual criticism of the Hebrew Bible in the Electronic Age: Advantages and Disadvantages of the Use of Electronic Tools. *Biblicum Jassyense*, 8:5-14.

## **CHAPTER 5**

### **SEMANTICS AND PRAGMATICS**



# A PLATFORM FOR PROCESSING HETEROGENEOUS DOCUMENTS

TUDOR BUMBU<sup>1</sup>, LYUDMILA BURTSEVA<sup>1</sup>, SVETLANA COJOCARU<sup>1</sup>,  
ALEXANDRU COLESNICOV<sup>1</sup>, LUDMILA MALAHOV<sup>1</sup>

<sup>1</sup>*“Vladimir Andrunachievici” Institute of Mathematics and Computer Science  
5 Academiei str., MD-2028, Chisinau, Republic of Moldova*

## Abstract

The paper describes the HeDy platform for processing **heterogeneous documents**, both contemporary and historical. A special aspect concerns the processing of Romanian texts, printed in Cyrillic characters in the 17th-20th century. One of the technological elements of text recognition is the analysis of the page structure and selection of page fragments with homogeneous content (text, images, formulas, musical scores, etc.). At large-scale processing of documents of the same origin (e.g., collected issues of a magazine for several years), the repetition of page makeups is observed, that is, page fragments are located approximately in the same place. Regarding the re-edition of old books, only a small number of makeups (even or odd pages, beginning or end of chapters, table of contents, etc.) are used within the same book. Page makeups are also repeated in series of books. To support page analysis during the re-edition of a book, a subsystem based on a neural network with deep learning is proposed, trained to recognize page makeups, followed by page fragmentation according to its makeup. This greatly facilitates further extraction of subfragments with homogeneous content, since the possible content of each element of the makeup is more or less fixed. Re-edition technology is embodied in the form of the developing platform for recognition of printed documents with heterogeneous content. The proposed subsystem is naturally included in this platform.

**Keywords:** computer science, digitization of cultural heritage, heterogeneous document recognition, framework architecture, integration of external applications.

## 1 Introduction

The solving workflow of heterogeneous content digitization problem consists of several steps, the main ones being: pre-processing, classification/layout analysis; assembling, and saving to digital form.

The pre-processing step is considered in detail in (Colesnicov et al., 2020), but during the last years, Deep Learning based image pre-processing has been included as an ordinary tool in many OCR systems, both commercial and free.

In the classification/layout analysis step, the heterogeneous document images are divided into fragments according to content type. Heterogeneous content digitization for general cases, when any type of content is allowed, is a time- and resources-consuming process. For this reason, the classification step was separated from the workflow. Another reason for such separation is that heterogeneous document image classification problem arises mostly once in the process of document archive digitization. After the digital archive is created, new documents are added through an interface, where the type of heterogeneous content is set. The one-time nature of heterogeneous document image classification problem makes it very suitable for online platform processing. Today the set of such platforms includes such giant commercial systems as Cannon IRIS (IRIS url, 2022) and Kofax Layout Classifier (Kofax url, 2022).

Besides the commercial systems, many particular projects, especially research ones, develop

their own platforms and systems, mostly based on Deep Learning (Subramani et al., 2020). Some particular projects concern general classification cases, but their majority consider some particular type of heterogeneous content. The works, concerning general case classification, also mostly use Deep Learning (Wu et al., 2021), (Mandivarapu et al., 2021). The interesting research (Kallempudi et al., 2022) proposes the processing of heterogeneous document images distorted during scanning. Another nonordinary approach is proposed by (Bakkali et al., 2020), showing the hybrid cross-modal methodology that learns simultaneously from a text corpus and image structural information. An approach not using Deep Learning is presented by (Rastogi et al., 2020). The authors propose a novel technique for document classification based on Formal Concept Analysis. The demonstrative example, that performs classification rather than layout analysis, is the processing of legal heterogeneous document images (Josi et al., 2022). Legal documents have strict patterns of their layouts. To process scanned images archived into digital ones, each image has to be classified as a specific legal document, and then the content is extracted taking the corresponding pattern into account. Another well-developed example is the table. Today the majority of table detection research uses pre-trained models (Ma et al., 2022), but many researchers also develop their own models and reusable deep learning elements like transformers (Lin et al., 2022). During the last years, two types of research appeared, which deal with heterogeneous content types of interest to our work. The first of them is the framework for historical document processing and analysis (Lenc et al., 2020). Like our HeDy, the HDPA framework provides the whole pipeline of document image processing. (Afzal et al., 2022) report on detecting mathematical formulas in scanned document images. Selecting HeDy chemical structures recognition backend tool, we tested DECIMER (Rajan et al., 2021), a Deep Learning-based system that has desktop and online versions.

Finally, we should mention that processing text-type layout elements also demonstrates interesting applications. Finding specific text areas on identity cards is in great demand (Visalli et al., 2021).

After reviewing the tools and platforms for digitizing historical documents, we have come to the conclusion that there were developed, or are under development, a lot of methods, tools, resources, and even platforms that offer the possibility to pre-process, recognize, post-process, translate historical documents in a fast and efficient way. Moreover, new recognition methods based on training OCR patterns on images with marked text lines (Clausner et al., 2020) increase the training speed of OCR engines, and therefore the speed of mass digitization. After recognition, OCR post-processing tools ensure quality text that can be used in further language processing, placement on the web for preservation, extension of metadata in digital libraries, etc. A valuable contribution is the development of digitization platforms that integrate most of the tasks for the successful processing of historical documents. A significant effort for the processing of old Romanian documents was made within the DeLORo project (Cristea et al., 2021) where the authors have recently come up with some promising recognition solutions, alternative ones to those proposed by us in this project that come once again to demonstrate the need to preserve and ensure access to the texts of the Cyrillic Romanian treasury, and, besides this, to strengthen and promote research and development of tools and resources for the digitization of the Romanian historical heritage.

## 2 General Structure of the Platform

The HeDy platform is developed based on advanced computing methods and techniques obtained from our experience in natural language processing, machine learning, and web development.

HeDy is a web/desktop application written under Django in Python and Javascript, which allows uploading images or PDF documents, automatically extracting content, its recognition, transliterating it into the modern Romanian Latin-based alphabet at necessity, editing, saving and exporting the results as texts and scripts. The purpose of the HeDy platform is to ensure access to

a wide range of digitization tools, including digitizing the Romanian historical heritage (Cojocaru et al., 2021).

These digitization tools include image preprocessing tools such as ScanTailor (ScanTailor url, 2022), ABBYY FineReader Image Processing (FineReader url, 2022), and OpenCV (OpenCV url, 2022); Cyrillic to Latin transliteration online tool AAConv (AAconv url, 2022); virtual keyboards for the Cyrillic Romanian, transitional alphabet, Soviet Cyrillic, and the modern Romanian. The tools are accompanied by the data sets like OCR models and word dictionaries for the 17th-20th centuries.

The essence of integrating these tools into a single platform is to provide the user with a unique graphical interface from which one can control the entire digitization process of a document. HeDy contains a set of modules to facilitate three main tasks: image preprocessing, OCR, and transliteration. The software part of the platform is available on Github (HeDy url, 2022). The web application is written in Django, which allows us to develop and integrate individual Python modules. Modules can run separately, and Django acts as a hub that connects the user interface with the outputs of the desired modules.

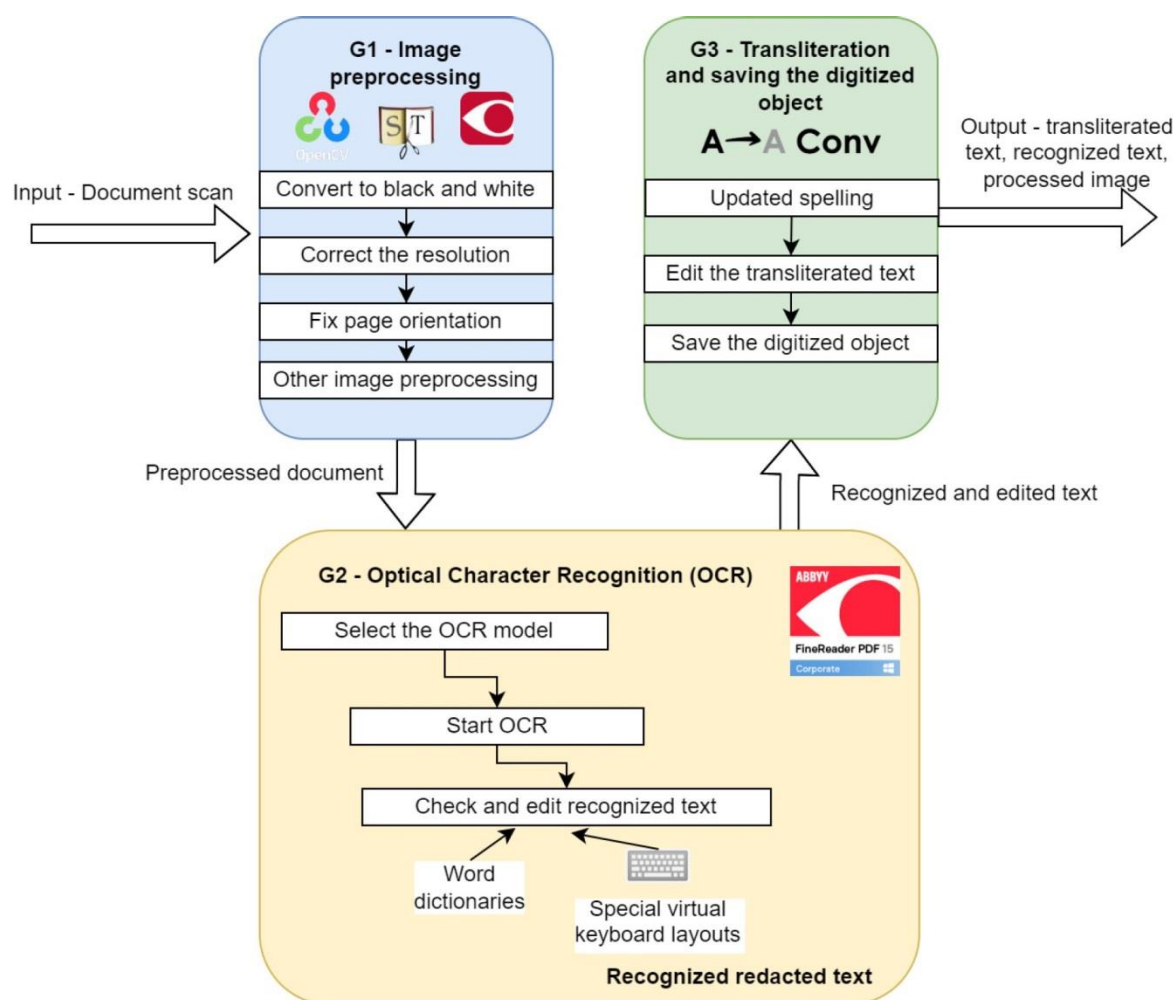


Figure 1. Tasks of document processing

HeDy digitization platform performs three main tasks (Fig. 1):

- *Image preprocessing.* The platform currently offers the user two basic preprocessing modules, namely, image binarization and image resolution correction, but also additional preprocessing modules added through integration with third-party applications such as

ScanTailor and FineReader. We also can divide heterogeneous content into homogeneous fragments and mark them with their content types.

- *Optical recognition of contents and editing of the recognized text or scripts.* For the Romanian historical documents, the proposed OCR system is based on pre-trained models with documents from the 17th-20th centuries. It is also possible to add new models by the administrator. OCR templates are in FineReader XML format and include the ground truth model, the required alphabet, and the word dictionary.
- *Post-recognition processing.* In particular, it includes the transliteration of the recognized text, as well as the editing of the text obtained during the transliteration, and the saving of the results.

### 3 Types of Heterogeneous Document Image Fragments

The set of homogeneous fragments of a heterogeneous document image, which was selected for the first version of HeDy, was described in detail in our works (Colesnicov et al., 2021), (Cojocaru et al., 2021), (Cojocaru et al., 2008). Here the choice summary is presented, except for the ordinary text.

**1. Historical text.** The framework includes recognition and digital saving of old texts for further processing or analysis. The related task has been in the development of our group for many years. So, the possible solutions, issues, and bottlenecks are well known.

**2. Mathematical text.** This choice is also inspired by our previous works and is related to the processing of the previous element. The problem of recognition of mathematical text elements arose when we implemented the re-edition of mathematical books printed with historical fonts.

This element is actually the set of several sub-elements.

- **Formula.** We chose to save the recognized mathematical formulas in the form of a LaTeX script. Popular and even official, for example in Linux, LaTeX technical writing system ensures the satisfaction of any requests for mathematical formulas recognition.
- **Technical drawings.** To present in digital form schemes, block schemes, and graphs, we chose to convert them to vectors. Such representation allows including scheme-like vectors directly to LaTeX or converting them to LaTeX drawings using the wide set of free converters both online and locally.
- **Tables.** Despite a table being a separate heterogeneous document element, today table images are successfully recognized by general-purpose software, in particular by FineReader.
- **Charts.** Charts and plots are important elements of heterogeneous documents. Unlike simple illustration, charts and plots contain digital information and can be reconstructed from this digital data.

**3. Music scores.** The recognition of sheet music (OMR) both printed and handwritten is a well-developed domain. HeDy has to provide the specific OMR of historical document scans, mostly of the fancy decorated music scores. We obtained the domain requirements for these elements from our past projects (Cojocaru et al., 2021), (Cojocaru et al., 2008).

**4. Chemical structures.** The task of recognizing images of chemical (molecular) structures is one of the oldest known text recognition tasks. Many both free and commercial solutions for OCR drug discovery have been developed during the last years. Currently, another application revives the chemical structures recognition task. This application springs from the modern establishment of standards of stereochemistry (Ahluwalia, 2008) for any organic chemistry publication. To re-edit books or just for the usage of figures from previous papers, scientific researchers need to recognize chemical molecular structure images in chemistry vector formats (SMILES, Mol, etc.). Due to the existence of a large number of converters of chemical vector

formats, we selected for the HeDy backend tools IMAGO (Imago url, 2022) and MolVec (MolVec url, 2022); both recognize Mol format. The reason for the choice was that they are implemented as jar which means cross-platform and easy-to-customize usage. The choice of tool for the front-end was more complicated. Today there are several free and full-functioned online editors of chemical structures. Making the choice, we consulted with organic chemistry domain scientific researchers. As the result, the set of requirements was obtained to edit recognized chemical structure vectors. The set includes some general possibilities: a toolbar of stereo elements and 3D editing mode. Moreover, the correction of recognition results requires specific possibilities like molecule structure normalization. To meet the majority of the requirements, the Ketcher (Ketcher url, 2022) editor was selected for the HeDy front-end.

HeDy chemical structures recognition workflow is illustrated in Fig. 2. Part a) presents the original image. The image is shown inside the IMAGO interface for demonstration purposes, while the HeDy backend uses the command line. Part b) shows the recognition result. The resulting Mol file is shown in the Open window of the Ketcher editor. Part c) shows the visualization of the resulted Mol in Ketcher. Part d) demonstrates the toolbar with the stereochemistry tools. Part e) presents the resulting molecule structure after applying the normalization tool.

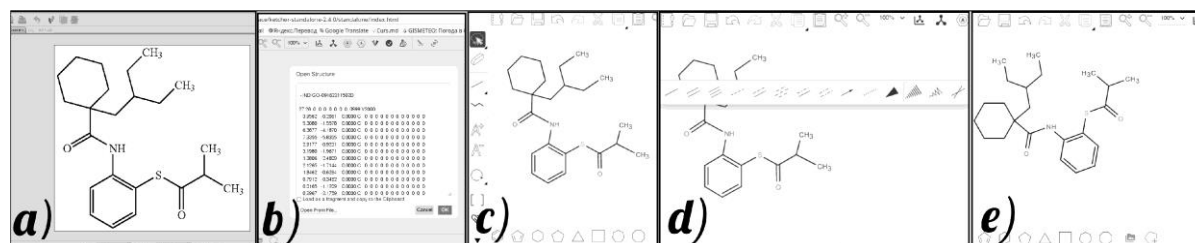


Figure 2. Workflow at chemical structures recognition

## 4 Steps of Digitization

The HeDy heterogeneous content platform integrates a variety of tools for such processing and allows users to convert images of printed pages into a set of texts and scripts, depending on the type of content. Non-recognizable portions of the page remain in image format. The resulting fragments of the page can be reconstructed from the original image.

The following types of fragments are provided: images, text, musical notes, mathematical formulas, chemical formulas and structures, chess diagrams, etc.

The platform implements a user interface, serving as a bridge between the user and a variety of tools for solving specific subtasks (see below), including third-party development. The platform is extensible and uses convergent technology to seamlessly integrate external subsystems (Cojocaru et al., 2020).

The platform supports a step-by-step workflow for processing heterogeneous documents. Each step performs a complete operation, the results of which can be transferred to the next step or used directly. Some steps may be skipped or repeated.

Document processing begins by loading one or more page images. PNG, JPEG, GIF, TIFF, and PDF files are supported. To select files, one can use the standard dialog box or drag and drop. The loaded pages are shown on the screen.

The next step is to refine the images so that they are ready for recognition. The tools for such processing are external programs Open CV, FineReader, ScanTailor, Gimp, and ImageMagick, selectable through the menu. The following Table 1 shows some operations over the page image and the corresponding external tools.

If there are pages with heterogeneous content, image fragmentation is performed. At this step, image areas with homogeneous content are selected and their type is determined. To do this,

ABBYY FineReader Engine is called through an intermediate module in Python.

Table 1. Some external tools integrated into the image preprocessing

| Function                                 | External modules                  |
|------------------------------------------|-----------------------------------|
| Image binarization                       | ScanTailor, FineReader 15, OpenCV |
| Manually setting the image resolution    | ScanTailor, OpenCV                |
| Automatic page orientation correction    | FineReader 15, OpenCV             |
| ISO noise removal                        | FineReader 15, OpenCV             |
| Splitting the image into multiple pages  | FineReader 15                     |
| Automatic image resolution correction    | FineReader 15                     |
| De-skewing                               | FineReader 15, ScanTailor         |
| Manual page orientation correction       | ScanTailor                        |
| Despeckle                                | ScanTailor                        |
| Correcting the illumination in the image | ScanTailor                        |
| Management of character thickness        | ScanTailor                        |

The obtained fragments are shown on the screen. A file with the coordinates of each fragment is also shown. Fragmentation can be corrected manually. Fragment type is chosen from a drop-down list: text, musical notes, mathematical formulas, chemical formulas and structures, and chess diagrams (Fig. 3). The remaining fragments get the “image” type by default.

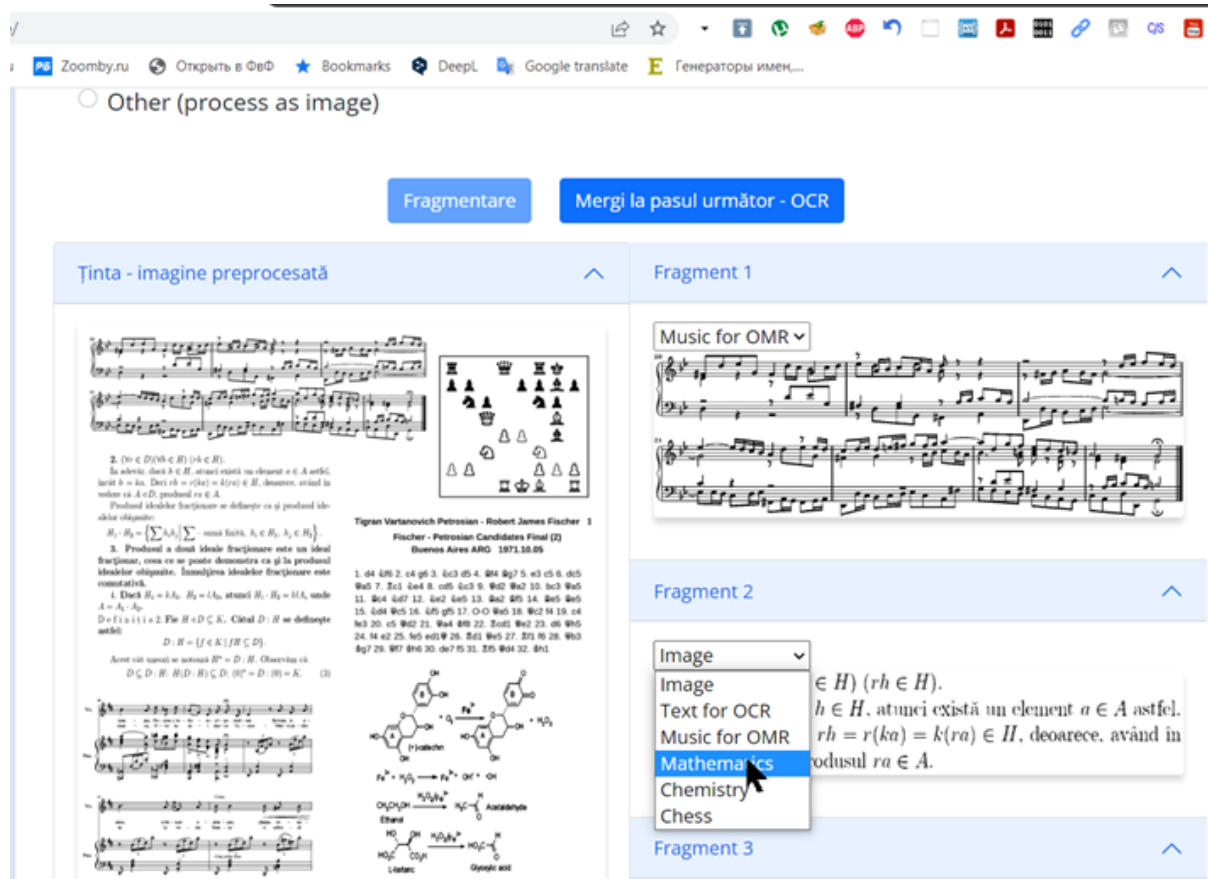


Figure 3. Manual selection of page fragment types

Specialized programs are used for fragment recognition according to the content type (FineReader, Mathpix, etc.). The result will be a text or script representation of the fragment: text in natural language,  $L^A T_E X$  script for math, MusicXML for notes, etc.

Recognition results are shown and can be checked and corrected. In some cases, one may need

to transliterate the recognized text, e.g., from the old Romanian Cyrillic to the modern Latin alphabet. A special AAconv application is called for this task.

The results of the recognition, along with images of the original pages, are assembled into a PDF file. The PDF file can contain other files as attachments. All the recognition results are appended to each page. Page maps, containing the coordinates of each fragment on the page, are also attached.

The assembled file, combining the original images with the recognition results and the page maps, is available to the user and can be used for various purposes, e.g., to recreate a graphic image of the original document.

## 5 Integration of External Programs

The digitization platform integrates a set of external programs to execute image preparation, recognition, transliteration, and other tasks. The web application is written in Django, which allows the development of new Python modules and the integration of external programs through the Python layers.

All necessary operations of the platform are executed by integrated external applications like Imagemagick, ScanTailor, GIMP, FineReader, etc.

We use a specific convergent technology of application integration (Cojocaru et al., 2020).

Let us discuss one of the applications processing image sets available in the HeDy platform.

Page makeup repeats into documents of the same origin, e.g., collected issues of a magazine for several years. Each book, magazine, or newspaper uses only a restricted number of page makeups. These are makeups for even and odd pages, the beginning and end of chapters, table of contents, etc. Regarding the re-edition of books, this means that page fragments are located approximately in the same place. Page makeups are also repeated in series of books.

A subsystem based on a neural network with deep learning is implemented and trained to recognize page makeups. Then fragmentation of pages according to their makeup follows. This greatly facilitates further extraction of sub-fragments with homogeneous content, since the possible content of each element of the given makeup is more or less fixed. The proposed subsystem is naturally included in our HeDy platform.

## 6 On the Evaluation of the HeDy Platform Productivity

HeDy platform is actively developing right now; therefore, not all aspects of its productivity have yet been assessed. Moreover, HeDy is the integrator of several external applications and the productivity of the platform as a whole depends largely on the efficiency of these components. For these reasons, we will discuss below only some aspects concerning the productivity and quality of the platform.

**Case 1.** For the OCR evaluation of 17th-century documents printed in Romanian Cyrillic characters, we used ABBYY FineReader PDF 15 OCR Editor (hereafter FR 15).

The dataset was prepared with pages from the New Testament (1648). 10 pages were selected for the training set. For the test set, 5 pages were selected which include about 7 090 characters and about 1280 words. A single page in the test set contains on average 1400 characters and 260 words. These pages have already been recognized, checked, and manually corrected in a previous training process. We used in some experiments a dictionary consisting of 4,582 words previously extracted from recognized pages from the 17th-18th centuries. The words in the dictionary were checked and corrected before being added to FR 15.

In this evaluation, the criteria investigated were OCR accuracy at both character and word levels. The results of the evaluation on several test/training pages will be summarized to obtain an overall OCR accuracy. The overall accuracy (GA) for a given experiment will be calculated according to the formula described in the paper (Heliński et al., 2012):

$$GA = \frac{\sum_1^n c_i}{\sum_1^n t_i}$$

where  $n$  is the number of test pages in each experiment,  $c_i$  is the number of characters or words correctly recognized from an arbitrary page, and  $t_i$  is the total number of characters/words in the test set used in the experiments.

The results are shown in Tab. 2. Each row in the table represents an experiment performed on the given training and test datasets consisting of a defined number of pages. Columns of the table contain identification of the experiment and the accuracy of recognition in characters or in words in two variants: with the dictionary or without it.

*The first experiment.* We trained the FR 15 model with a page from the training set. From this page, 662 glyphs were extracted in the training process using the GUI interface provided by FR 15. The most encountered characters are 'n' and 'a' which were encountered 51 and 48 times, and the minimum number of glyphs is the character 'o' which was encountered only once in the training set. In this iteration, we evaluated the OCR model with and without the word dictionary. With the dictionary, we counted 132 wrong characters out of a total of 1460 characters, and out of 256 words in total, 69 have at least one wrong character. Without the word dictionary, we have 171 wrong characters and 108 words have at least one wrong word.

*The second experiment.* We trained the FR 15 model with 2 pages from the training set. From these pages, 1275 glyphs were extracted. The glyph 'a' occurs 95 times in this training set, and on average there are more than 30 glyphs per character. In dictionary recognition of a page from the test set, we counted 99 wrong characters out of 1460 characters, and 56 words have at least one wrong character. Without the word dictionary, we counted 134 wrong characters and 76 wrong words out of 256 words.

*The third experiment.* We further trained the OCR model with 5 pages from the training set with over 2500 glyphs. In dictionary recognition, out of 1460 characters tested 73 are misrecognized, and out of 256 words evaluated, 50 are misrecognized. Without a dictionary, we have 86 misidentified characters and 69 words with errors. Most of the errors made in this experiment are due to ligatures, where 44 out of 56 ligatures are erroneous.

*The fourth experiment.* The OCR model was trained with 7 pages from the training set. The number of glyphs in the set is over 3600. With the dictionary, 59 characters out of 1460 were misrecognized, and 52 words out of 256 have at least one wrong character. Compared to experiment number 3, the character-level accuracy increased by approx. 1%, but the word-level accuracy remained the same, and we even got 2 more wrong words than in the previous experiment when we also used the dictionary.

Table 2. OCR accuracy on 17th-century documents

| Experiment |                       |        | With dictionary |           | Without dictionary |           |
|------------|-----------------------|--------|-----------------|-----------|--------------------|-----------|
| Nr.        | Training              | Test   | $A_{ch}$        | $A_{wrd}$ | $A_{ch}$           | $A_{wrd}$ |
| 1          | 1 page (662 glyphs)   | 1 page | 0.91            | 0.73      | 0.88               | 0.57      |
| 2          | 2 pages (1275 glyphs) | 1 page | 0.93            | 0.78      | 0.90               | 0.70      |
| 3          | 5 pages (2540 glyphs) | 1 page | 0.95            | 0.80      | 0.94               | 0.73      |
| 4          | 7 pages (3668 glyphs) | 1 page | 0.96            | 0.796     | 0.95               | 0.75      |

**Case 2a.** In 2017, we re-published in the modern Latin Romanian script the book (Andrunachievici et al., 1979). Modern systems for mathematical formula recognition were not used. The resulting timeline of the work was the following:

- automated scan, OCR, and transliteration took 3 man-hours;
- manual text editing took 2 man-days;
- manual retype of formulas in LaTeX (formulas/text = 50/50) took 2 man-months of a highly qualified mathematician.

**Case 2b.** In 2021, the development of techniques based on Deep Learning made mathematical formula recognition widely available. We tested the recognition of the same book (Andrunachievici et al., 1979) with the commercial online system Mathpix, which converts formulas and text from images into LaTeX scripts. We show in Fig. 4 the result of recognition checking for page 110. The page was scanned, the image was recognized in LaTeX, the resulting script was compiled into the graphical formula presentation, and then this reconstructed image was visually compared with the original page image. We found on the page only two errors: 1) There were no “\$” signs around the “\beta” command for the letter “β”; 2) The letter «БІ» in the word «БІнсэ» was recognized as «Б» because of bad image quality. It should be noted that Mathpix does not provide the Cyrillic Moldavian script for Romanian; nevertheless, the textual part was recognized with only one error.

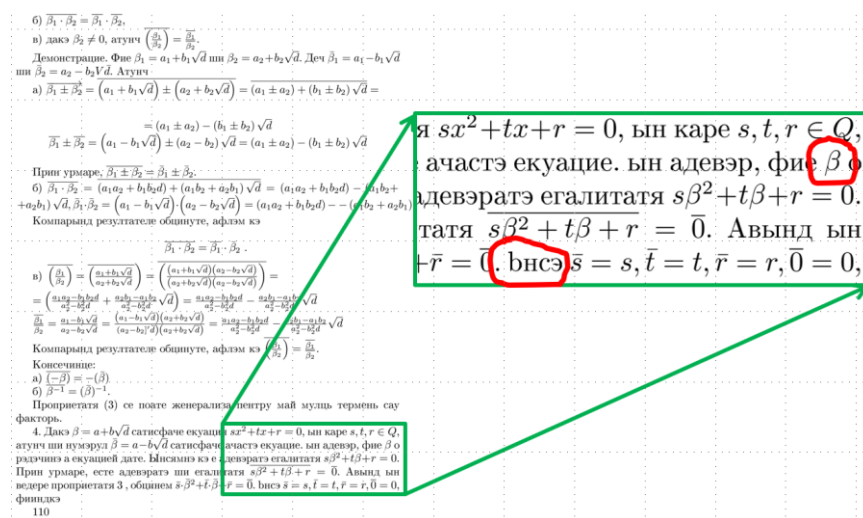


Figure 4. Page checking after recognition by Mathpix

## 7 Conclusions

HeDy platform includes modules needed to perform three main tasks related to the digitization of heterogeneous documents: image pre-processing, recognition, and post-processing including transliteration of recognized Romanian text from Cyrillic to Latin. The platform works as a web application. A scanning cycle includes eight processing steps. In the end, the user gets a digitized document, containing the recognized text and scripts, and the processed images.

HeDy platform presents a useful tool for libraries, publishers, and researchers in various fields. Along with these, the application can also be useful to a wide circle of users interested in the digitization of printed cultural heritage.

## Acknowledgments

This work was prepared as part of the research project 20.80009.5007.22 “Intelligent information systems for solving ill-structured problems, processing knowledge and big data”.

## References

- AAconv: <https://translitera.cc/>
- M. Z. Afzal, K. A. Hashmi, A. Pagani, M. Liwicki, & D. Strieker. (2022, January 6). *DeHyFoNet: Deformable Hybrid Network for Formula Detection in Scanned Document Images*. <https://doi.org/10.20944/preprints202201.0090.v1>
- V.K. Ahluwalia. (2022). *Stereochemistry of Organic Compounds*. Springer International Publishing,.

- V. Andrunachievici, & I. Chitoroaga. (1980). *Numere si ideale. (Numbers and ideals.)* Chisinau, “Lumina”, 1979. 224 p. – In Romanian (the Cyrillic script). – Russian edition: Chisinau, “Lumina”. – Reprinted in Romanian (the Latin script): 2017.
- S. Bakkali, Z. Ming, M. Coustaty, & M. Rusinol. (2020). Visual and Textual Deep Feature Fusion for Document Image Classification. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 2394-2403).
- C. Clausner, A. Antonacopoulos, & S. Pletschacher. (2020). Efficient and effective OCR engine training. *Int. J. Document Anal. Recognit.* 23(1): 73-88.
- S. Cojocaru, L. Burteva, A. Colesnicov, V. Demidova, L. Malahov, & M. Petic. (2008). Instrumentar Multimedia pentru Facilitarea Promovarii Arte Muzicale. *UTM “Studia Universitatis”, seria Stiinte exacte si economice*, Chisinau, Moldova, no.3(13), pp.82-85.
- S. Cojocaru, A. Colesnicov, L. Malahov, & L. Burtseva. (2020). On convergent technology in development of information systems for processing of documents with heterogeneous content. In: *Proceedings of the Workshop on Intelligent Information Systems*, 04-05 December, 2020, Chisinau, Republic of Moldova, pp. 61-68, ISBN 978-9975-68-415-6.
- S. Cojocaru, A. Colesnicov, L. Malahov, & L. Burtseva. (2021). “Chapter 6. Digitization Technology of Old Romanian Documents Printed in the Cyrillic Script”, *Horizons in Computer Science Research*. Volume 21, Thomas S. Clary, Editor. Nova Science Publishers, Inc. ISBN: 978-1-53619-103-5, ISBN: 978-1-53619-107-3, pp. 185-217.
- A. Colesnicov, L. Malahov, S. Cojocaru, & L. Burtseva. (2020). Semi-automated workflow for recognition of printed documents with heterogeneous content. *Computer Science Journal of Moldova*, 28(3), 223-240.
- A. Colesnicov, L. Malahov, S. Cojocaru, & L. Burtseva. (2021). Development of a platform for processing heterogeneous printed documents. In *Proceedings of the Conference on Mathematical Foundations of Informatics MFOI-2020*, January 12-16, Kyiv, Ukraine, pp.108-122, ISBN 978-966-999-143-0.
- D. Cristea, P. Rebeja, C. Padurariu, M. Onofrei, & A. Scutelnicu. (2021). Data Structure and Acquisition in DeLORO – a Technology for Deciphering Old Cyrillic-Romanian Documents. *Proceedings of ConsILR* (2021), pp. 59-74.
- FineReader. <https://www.abbyy.com/ocr-sdk/features/image-processing/>
- HeDy. <https://github.com/bumbutudor/PlatformaDigitizare>
- M. Heliński, M. Kmieciak, & T. Parkoła. (2012). *Report on the comparison of Tesseract and ABBYY FineReader OCR engines*: [https://www.digitisation.eu/fileadmin/Tool\\_Training\\_Materials/Abbyy/PSNC\\_Tesseract-FineReader-report.pdf](https://www.digitisation.eu/fileadmin/Tool_Training_Materials/Abbyy/PSNC_Tesseract-FineReader-report.pdf)
- Imago OCR, chemical image recognition toolkit. <https://github.com/ggasoftware/imago>
- IRIS - Intelligent Information Management. (2022, May 20). IRIS. <https://irisdatacapture.com/>
- F. Josi, C. Wartena, & U. Heid. (2022). Preparing Legal Documents for NLP Analysis: Improving the Classification of Text Elements by Using Page Features. 17-29. doi:10.25968/opus-2161.
- G. Kallemudi, K. A. Hashmi, A. Pagani, M. Liwicki, D. Stricker, & M. Z. Afzal. (2022). Toward Semi- Supervised Graphical Object Detection in Document Images. *Future Internet*, 14(6), 176. MDPI AG. Retrieved from <http://dx.doi.org/10.3390/fi14060176>.
- Ketcher overview. <https://lifescience.opensource.epam.com/ketcher/index.html>
- Kofax Intelligent Automation for Digital Workflow Transformation. (n.d.). (2022). Kofax Intelligent Automation for Digital Workflow Transformation. Retrieved September 28, 2022, from <https://www.kofax.com/>
- L. Lenc, J. Martmek, P. Kral, A. Nicolao, & V. Christlein. (2020, May 20). HDPa: historical document processing and analysis framework. *Evolving Systems*, 12(1), 177-190. <https://doi.org/10.1007/s12530-020-09343-4>
- W. Lin, Z. Sun, C. Ma, M. Li, J. Wang, L. Sun, & Q. Huo. (2022, August 9). TSR- Former: Table Structure Recognition with Transformers. arXiv.org. Retrieved September 28, 2022, from <https://arxiv.org/abs/2208.04921>
- C. Ma, W. Lin, L. Sun, & Q. Huo. (2022, March 17). Robust Table Detection and Structure Recognition from Heterogeneous Document Images. arXiv.org. Retrieved September 28, 2022, from <https://arxiv.org/abs/2203.09056>
- J. K. Mandivarapu, E. Bunch, Q. You, & G. Fung. (2021, June 25). Efficient Document Image Classification Using Region-Based Graph Neural Network. arXiv.Org. <https://arxiv.org/abs/2106.13802>

MolVec, A feeble attempt at molecular recognition. <https://github.com/ncats/molvec>

OpenCV. <https://opencv.org/>

K. Rajan, A. Zielesny, & C. Steinbeck. (2021, August 17). DECIMER 1.0: deep learning for chemical image recognition using transformers. *Journal of Cheminformatics*, 13(1). <https://doi.org/10.1186/s13321-021-00538-8>

Mouli Rastogi, Syed Afshan Ali, Mrinal Rawat, Lovekesh Vig, Puneet Agarwal, Gautam M. Shroff, & Ashwin Srinivasan (2020). Information Extraction from Document Images via FCA based Template Detection and Knowledge Graph Rule Induction. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2377-2385.

ScanTailor. <https://scantailor.org/>

N. Subramani, A. Matton, M. Greaves, & A. Lam. (2020, November 27). A Survey of Deep Learning Approaches for OCR and Document Understanding. arXiv.Org. <https://arxiv.org/abs/2011.13534>

F. Visalli, A. Patrizio, & M. Ruffolo. (2021). A Two Step Fine-tuning Approach for Text Recognition on Identity Documents. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence*. <https://doi.org/10.5220/0010252208370844>

Xingjiao Wu, Yingbin Zheng, Tianlong Ma, Hao Ye, & Liang He. (2021). Document image layout analysis via explicit edge embedding network. *Information Sciences*, 577, 436-448.



# HOW TO CREATE AN ADAPTIVE E-LEARNING SYSTEM

ALEXANDR PARAHONCO<sup>1</sup>, MIRCEA PETIC<sup>2</sup>

<sup>1</sup>*Junior research, Vladimir Andrunachievici Institute of Mathematics and  
Computer Science,*

*Alecu Russo Balti State University, Republic of Moldova,  
alexandr.parahonco@usarb.md*

<sup>2</sup>*Leading researcher, Vladimir Andrunachievici Institute of Mathematics and  
Computer Science,*

*Alecu Russo Balti State University, Republic of Moldova,  
mircea.petic@math.md*

## Abstract

Nowadays, there exist e-learning systems that combine adaptive testing with adaptive content for each student. Altogether, they make adaptive learning. Adaptive educational systems are educational information and communication technologies that react in real-time to the actions of the student and, in accordance with the information received, provide him with individual support. First of all, three key questions have to be answered: what is modeled (what variables are put into the adaptation model), how it is modeled (based on what interactions between variables the student receives educational material), and how the work of the adaptation model is supported. Then one of three scenarios is implemented, where the object of adaptation can be: content, tasks (assignments/items), or the order of presentation of educational materials. This paper answers the main question "How?" and proposes our solution.

## 1 Introduction

The twenty-first century has witnessed the emergence of groundbreaking information technologies that brought changes in our life. Since the mid-1990s, the Internet gave a start to methods, tools, and gadgets that covered all academic disciplines and business sectors. Soon, afterward, we witnessed a chain of web 2.0 technologies like e-commerce, which started social media platforms, e-business, e-learning, e-government, cloud computing, and more other in 2021 (Yamin Mohammad, 2019).

E-learning has evolved into a learning management system (LMS) or software applications for the administration, documentation, tracking, reporting, automation, and delivery of educational courses, training programs, or learning and development programs since the 1990s and 2000s. These systems may be used in distance learning and blended (hybrid) learning concepts.

Over the last decade, IT specialists have focused on the implementation and adaptation of the e-learning process to the needs of pupils. This idea has led to the creation of programs and special LMS that should improve learners' competencies.

The aim of the article is to examine the background of the subject and create some prototypes of adaptive e-learning systems.

Previous research showed that there are numerous studies about adaptive technologies, theories, and successful prototypes of applications that have been introduced to the world. The paper presents some common methodologies for developing adaptive platforms on the basis of conducted research as a common framework. Then a comparative analysis of well-known platforms is proposed with technical details. Finally, the paper presents our example of an adaptive e-learning application.

## 2 State-of-the-Art

Generally, adaptive learning is a broad term, also known as adaptive teaching, and is an educational method that uses computer algorithms and artificial intelligence to orchestrate the interaction with the learner and deliver customized resources and learning activities to address the unique needs of each learner (Andreas Kaplan, 2021).

From a pedagogical point of view, adaptation touches upon three aspects of education: personalization, individualization, and differentiation. Individualization allows students to progress through the curriculum at different speeds according to their learning needs. Differentiation encourages teachers to apply various methods or approaches according to the preferences of each student. And personalization refers to instruction that is paced to meet learners' needs, tailored to their preferences, and tailored to the specific interests of different learners (Amanda Stedke, 2017).

Any programmer has to decide what strategy to choose for further implementation into the application. This helps to understand which object of adaptation to work with. The next step is to answer 3 questions: what is modeled (what parameters are put into the adaptation model), how it is modeled, and how the work of the adaptation model is supported. Then one of three scenarios is implemented, where the **object of adaptation** can be (Fig. 1.): **content**, **tasks** (assignments/items), or **the order of presentation of educational materials** (Vilkova K.A. and Lebedev, D.V., 2020).

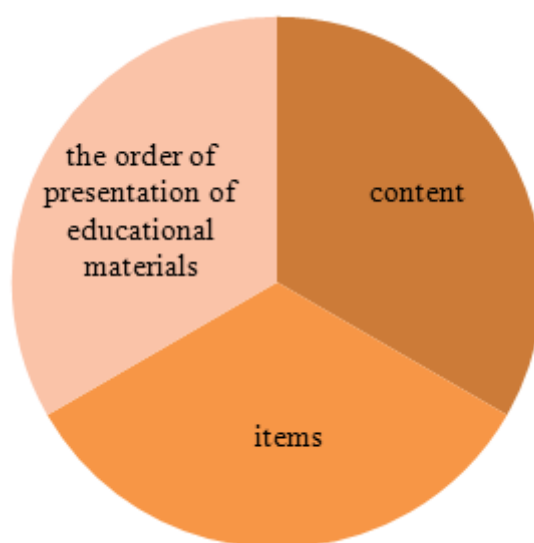


*Figure 1. Adaptation scenarios (adaptation objects)*

If in the educational system, the object of adaptation is **content**, then it functions according to the following algorithm: first, it analyzes the student's response to the task and, in the case of an incorrect answer, offers him feedback, tips, or additional educational materials.

Adaptation of **tasks** throws light on the selection of tasks that correspond to the test taker's level of knowledge. Strategies for presenting test items in adaptive testing can be divided into two-step and multi-step, according to which different technologies for forming adaptive tests are used (Fig. 2).

A two-step strategy involves two stages. In the first stage, all subjects are given the same input test, the purpose of which is to carry out a preliminary differentiation of students along the axis of the measurement variable. According to the results of differentiation in the second stage, an adaptive mode is organized, and adaptive tests are built.

As a result of the development of the Item Response Theory (IRT), which provides a unified interval scale for assessing the parameters of subjects and the difficulty of test items, it became possible to optimize the task selection procedure in a new way for modeling effective adaptive tests. Multi-step adaptive testing strategies began to develop, in which, in the process of completing sets of tasks, each subject moves along his own individual trajectory (Glebov V.A., 2012).

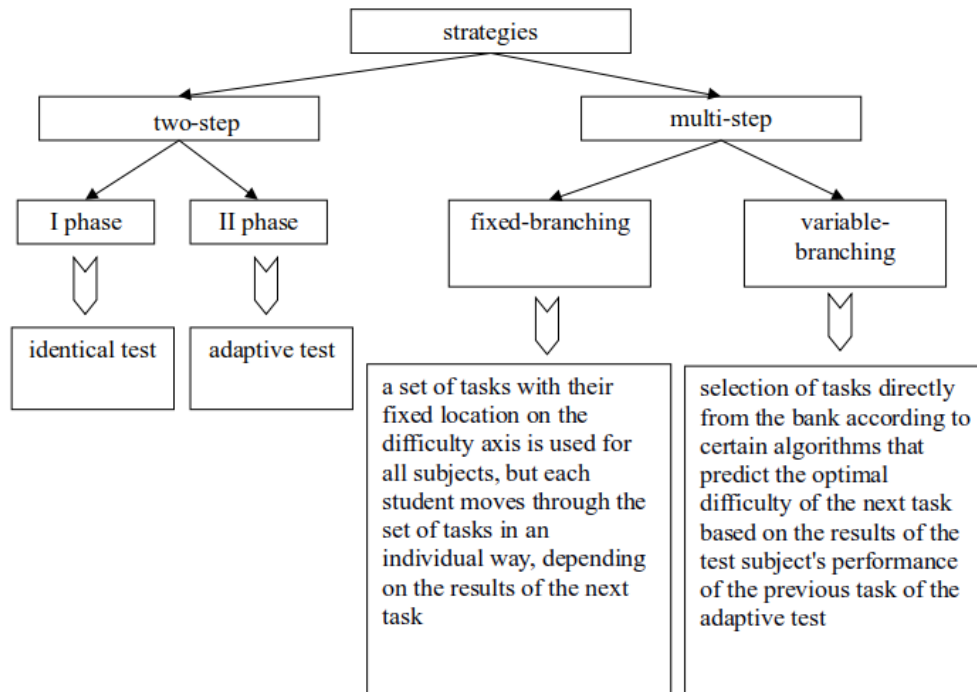


Figure 2. Adaptive testing strategies

**The order of presentation of educational materials** refers to the curriculum modification for each learner, depending on his/her achievements. In this case, it is possible to escape “unnecessary” subjects that the learner is supposed to know or to jump over some subjects if he or she decides to. It depends on the adaptation level and specific tasks.

After modeling the object of adaptation, one needs to think of

- the expert - a graphical model of the "ideal" state, of everything the person could learn using this system;
- the learner - a model of the learner's current state, which shows how likely the learner is to know each of the nodes in the expert graph;
- the tutor - the algorithms that determine what content to show and when. The expert model and the learner model inform the tutor. The tutor seeks to optimize content for relevance, challenge, and efficiency;
- the interface - which is how to display the learning experience to the learner. In many adaptive learning experiences, the interface changes based on the learner model and the tutor's goals.

A sizable connected graph with all the information you want learners to know makes up the **expert model**. As the name implies, the model is created by an expert on the subject (or experts on subjects). It is a static model. Only when the scope of learning outcomes changes does the expert model alter. Alternatively, when challenges and chances to improve the adaptive learning system appear.

The **student model** represents the learner's current level of competence. Therefore, the learner model assigns a probability of 1-99% to each node in the expert graph. This graph is updated by the system each time a learner completes an activity. The likelihood rises if a student provides a proper response to a question. If the learner provides an incorrect response, the chance falls. Even-

ry student uses a different learner model. Therefore, a new learner model is created whenever a new learner enters the system. The tutor will later select how to arrange the learning materials using the learner model. The learner model can be updated using a variety of algorithms. According to **knowledge spaces theory**, the probabilities for linked skills should change when a learner gains proficiency in one area.

For updating skill probability, some adaptive learning systems use straightforward heuristic models. The probability changes in IRT along a sigmoid curve. This curve has a more traditional shape when applied to **Bayesian knowledge** tracing. One requires a way to estimate these parameters for both Bayesian knowledge tracing and item response theory. Researchers are also developing models using traditional **machine-learning** techniques, such as **neural networks**.

The order in which the student will complete the exercises is decided by the **tutor**. The tutor will modify the path it will take to optimize for that learner following each change to the learner model.

The tutor's objective is to complete the learner's expert graph in the shortest period of time possible. Some systems let students ignore the rest, and concentrate just on a few topics. The approach the tutor will take is as unique for each learner as the learner model itself. The instructor is algorithms with some data, whereas the expert and learner components are data with some algorithms. The learner's activities and the tutor's choice of skills to emphasize are both up to them.

Some systems for adaptive learning will modify the user **interface**. The interface would be simpler and would concentrate more on the task at hand as the learner became less adept at a skill. More of the complete interface comes together as the learner's skill increases. Many refer to this procedure as "scaffolding" (Kevin Heis, 2019).

### 3 Comparative Analysis of Adaptive Platforms

In order to create a more detailed guide for creating an adaptive learning system, we have analyzed the three most famous platforms: Geekie, Knewton Alta, and Smart Sparrow.

**Geekie** is a paid learning platform powered by artificial intelligence (AI) to prepare Brazilian students for their high school final exams. Geekie evaluates each action of the wards and transfers them to the teacher. At the beginning of the training, students take a short test and a survey about the goals of education, the desired university, and what subject the student is striving for. Geekie then creates curricula, selecting the essentials according to each student's needs: more economics, for example, for aspiring math candidates. As a consequence of working with the Geekie platform, the system rearranges the curriculum in an individual manner, adapting over time to the degree of development of the student's abilities and how the algorithm understands him.

The program constantly collects data and learns from it (Big Data and Data Mining technologies). Geekie also collects data on required competencies that are required for more complex subjects, generating material that can be analyzed and incorporated by Geekie education experts. This platform provides **adaptation at the level of the order of presentation of educational materials** (Gil N, 2016).

The **Knewton Alta** educational Web platform is a platform on which programs and applications with an adaptive function are developed. As a result of many years of work and various experiments, the Knewton team managed to create universal algorithms and develop an extensive infrastructure for collecting, analyzing, and using information about the progress of students, including (Fig. 3):

- The data collection infrastructure;
- Inference infrastructure;
- Personalization infrastructure.

It contains two elements of adaptation: **tasks** and **the order of presentation of educational materials** (Bogdanova D.A., 2014).

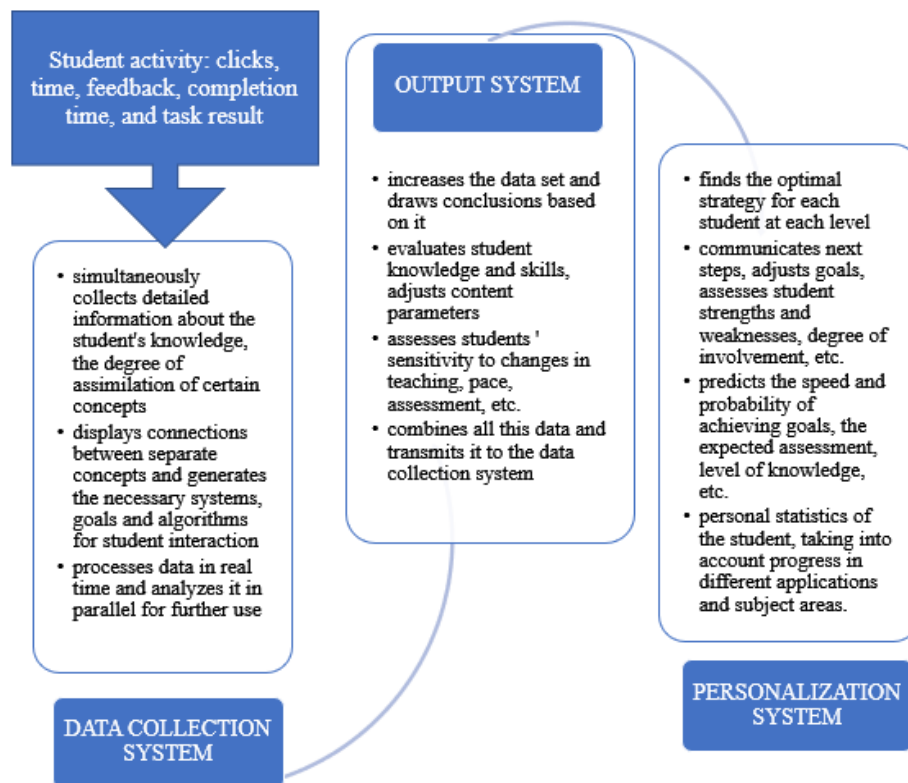


Figure 3. The Knewton Alta's infrastructure

Knewton's technology combines data from diverse sciences: statistics, psychometrics, and machine learning data, as well as unified participant histories. In the course of work, the algorithms of the Knewton platform analyze all educational materials according to hundreds of parameters and, based on the selection of key ideas, theories, and concepts, offer the optimal structure, information format, and level of complexity. At the same time, the educational platform analyzes the student's knowledge, the rhythm of his work, the ability to process information, and other characteristics of the student (Thomas A. Conklin, 2016). Based on this data, the system decides what to offer the student next: an interactive exercise, a test, a game, a training video, or something else. The platform is paid and allows you to study courses in **four groups of disciplines**: chemistry, economics, mathematics, and statistics.

In terms of theories and approaches, Knewton Alta employs Item Response Theory (IRT), Probabilistic Graphical Models (PGMs), and Hierarchical agglomerative clustering. Instead of aggregate-test performance, IRT models students' abilities using question-level performance. Rather than assuming that all questions contribute equally to our understanding of students' abilities, IRT takes a more nuanced approach to the information each question provides about a student. PGMs, in turn, which encompass statistical methods such as Bayesian networks and Markov random fields, allow data scientists to code and manipulate probability distributions over multi-dimensional spaces in which there are hundreds or even thousands of variables at play. Thus, PGMs allow Knewton analysts to build complex models one effect at a time, relating the many learning activities they observe to estimations that are useful for a recommendation. Hierarchical clustering is a data mining analysis method that aims to build a hierarchy or structure of clusters. The technique is used at Knewton to detect latent structures within large groups and build algorithms that determine how and by what features students should be grouped (Knewton adaptive learning).

Within the Web platform, **Smart Sparrow's** adaptability consists of feedback on any interaction with educational materials. There are **three levels of adaptation** on the platform: feedback, modification of the curriculum, and the ability of the educator to adapt the knowledge transfer

process. The functionality of this platform allows teachers to create interactive content that can be adapted for any group of students in accordance with the topic of the subject under study and the specific requirements for the group's learning process. Educators can define individual learning paths for students, interact with them in real time, and use a range of ready-made templates to save time during content elaboration.

The designed adaptivity of Smart Sparrow covers a common algorithm Bayesian Knowledge Tracing (BKT), which estimates the rate at which learning occurs and IRT. By using preset rules or algorithms to shape content to the needs of an individual student — just as an instructor does with students in one-to-one conversations — technology can provide necessary remediation, extra content, words of encouragement, and more (What is adaptive learning).

#### 4 Adaptive System Prototype

Our adaptive system offers adaptation at the level of personalized learning that widens learning needs and provides the possibility to expand learning materials and test feedback. It also provides differentiation opportunities that depend on teacher experience and practice. It uses two objects of adaptation: content and evaluation.

Therefore, content object refers to the adaptation of learning content to the student's interests and capacity for studying; evaluation – refers to a computer-based test that adapts to the examinee's ability level.

We decided to develop the adaptive system on the basis of the Moodle E-learning platform as it is an open-source project, widely spread and simple in usage. Bearing in mind that the Moodle platform consists of various plugins, we have developed an adaptive learning module. Thus, we used the equivalent of a closed system (module), which contains adaptive learning mechanisms and can interact both with the core of the Moodle platform and with individual elements of the course (Fig. 4).

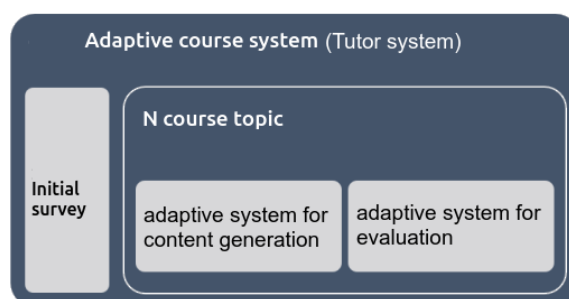


Figure 4. Adaptive course model for the Moodle platform

Based on the Moodle architecture, such a module is a **course plugin**. As can be seen from the picture above, the course plugin represents the *tutor* element from the theory. At the present moment, it is not yet finished. The same situation is with the adaptive system for content generation.

However, an adaptive system for evaluation is developed and validated in scientific experiments and different tests. Its name is TestWid (from the test widget). As it was said, every object of adaptation has to implement 4 elements:

- the expert model;
- the learner model;
- the tutor system;
- the interface.

Thus, the expert model of the test is the ideal result of passing through 9 items out of 15 in a short amount of time (Fig. 5). It should be mentioned, that the adaptive system for evaluation applies a *multi-step fixed branching strategy*. It uses a *Bank of items* and each time the learner takes a test, new items are randomly selected (15 items in total). The *learner model* represents the line

chart of all items that have been proposed to resolve (Fig. 6). Plugin's *tutor system* interacts with both models and decides what item to propose next and when the test should be finished. The *interface* of the evaluation system takes the pattern of Moodle.

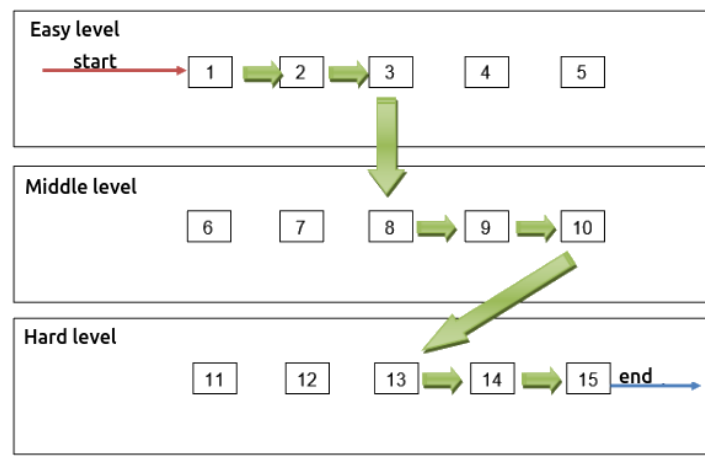


Figure 5. The expert model of the test

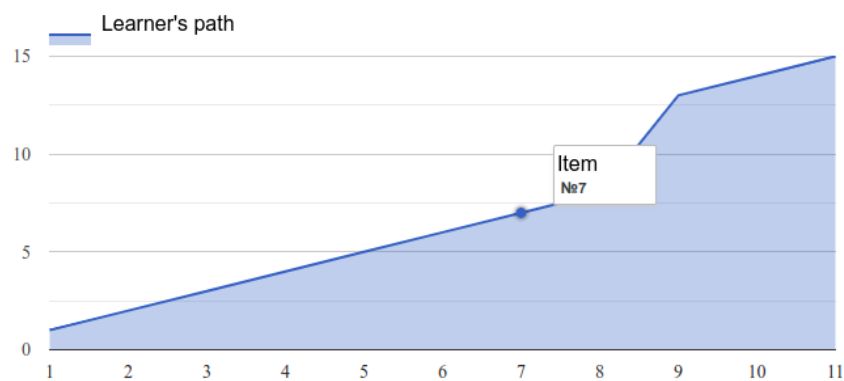


Figure 6. The learner model of the test

Currently, we are working on an adaptive system for content generation. Our research in the domain led us to natural language processing (NLP) and **Data mining**. NLP is a subfield of linguistics, computer science, and artificial intelligence concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of natural language data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. It is a component of artificial intelligence (AI). Using a combination of machine learning, **deep learning** and neural networks, natural language processing algorithms hone their own rules through repeated processing and learning.

Data mining is the process of analyzing a large batch of information to discern trends and patterns. It associates with methods of machine learning, statistics, and database systems. Data mining is an essential component of data analytics as a whole and one of the fundamental subfields of data science, which makes use of sophisticated analytics methods to unearth informational content in data sets. More specifically, data mining is a step in the knowledge discovery in databases (KDD) procedure, which is a data science approach for obtaining, processing, and evaluating data. Although they are frequently considered to be separate concepts, data mining and KDD are occasionally used interchangeably.

## 5 Final Thoughts

The current system of education requires improvements. Today's level of IT development promotes adaptive learning as the solution to the problem. This article proposes a guide for developing an adaptive system and backs it up with evidence from the analysis of such adaptive systems as Geekie, Knewton Alta, and Smart Sparrow. Moreover, we implement this methodology in our application for adaptive e-learning.

First of all, the developer should decide what level of adaptation to implement in the application. It increases from differentiation and personalization to individualization. Having in mind the level, it is necessary to define the object of adaptation. Each of which, content, items, or the order of presentation of educational materials, means certain application architecture and approaches/theories to apply. Beyond that, there should be created and managed expert and learner models under tutor control with a user-adopted interface.

To realize all these adaptive objects and develop the tutor system, there are lots of theories and approaches. We gathered several of them and explained how to get them together.

## Acknowledgements

This article was written as part of the research project 20.80009.5007.22 Intelligent information systems for solving ill-structured problems, processing knowledge and big data.

## References

- BOGDANOVA D.A. (2014). About an adaptive platform for individual learning. In: *XI International Scientific and Methodological Conference New Educational Technologies in Higher Education*, Ekaterinburg, February 18-20. Ekaterinburg: UrFU, 2014, pp. 202-207.
- CONKLIN T.A. (2016). Knewton. *Academy of Management Learning and Education*, 7. DOI: 10.5465/amle.2016.0206
- GIL N. (2016). How software that learns as it teaches is upgrading Brazilian education. *Guardian*, 1. [online]. [Accessed 08.09. 2022]. Access mode: <https://www.theguardian.com/technology/2016/jan/10/geekie-educational-software-brazil-machine-learning>
- GLEBOV V.A. (2012). Adaptive testing as a modern tool for control of learning outcomes. In: *Current issues of psychology and pedagogy*, 23, 78-87 [online]. [accessed 21 September 2022]. Access mode: <https://cyberleninka.ru/article/n/adaptivnoe-testirovanie-kak-sovremennoe-sredstvo-kontrolya-rezultatov-obucheniya>
- HEIS K. (2019). How to build an adaptive learning system. *freeCodeCamp*, 9, [online]. [accessed 29 September 2022]. Access mode: <https://www.freecodecamp.org/news/adaptive-learning-systems/>
- KAPLAN A. (2021). *Higher Education at the Crossroads of Disruption: The University of the 21st century*. Emerald Publishers. ISBN 9781800715042.
- Knewton adaptive learning. Building the world's most powerful recommendation engine for education*. [online]. [Accessed 03.01.2021]. Access mode: <https://cdn.tc-library.org/Edlab/Knewton-adaptive-learning-white-paper-1.pdf>
- VILKOVA K.A., LEBEDEV, D.V. (2020). Adaptive learning in higher education: pros and cons. In: *Modern Analytics of Education*. Moscow: HSE, 7(37), p.9. ISSN 2500-0608.
- YAMIN M. (2019). Information technologies of 21st century and their impact on the society. In: *International Journal of Information Technology*, 11(16-8), 759-766. DOI: 10.1007/s41870-019-00355-1
- STEDKE A. (2017). Differentiation, individualization, and personalization: What they mean, and where they're headed. In: *eSchool News*, no. 8. [online]. [accessed 5 September 2022]. Access mode: <https://www.eschoolnews.com/2017/08/01/differentiation-individualization/>
- What is adaptive learning*. [online]. [Accessed 03.09.2022]. Access mode: <https://www.smartsparrow.com/what-is-adaptive-learning/>

# ONTOLOGICAL STRUCTURING OF NATURAL LANGUAGE CONTENT IN ENCYCLOPEDIC WIKI-PORTALS AS A BASIS FOR SEMANTIC SEARCH

JULIA ROGUSHINA

*Institute of Software Systems of the National Academy of Sciences of Ukraine,  
40, Ave Glushkov, Kyiv, 03181, Ukraine*

## Abstract

We consider the theoretical principles used for organization of the knowledge representation system in semantic Wiki resources and their expressiveness. The specifics of the application of these principles for the development of the Web-oriented encyclopedic portal is analyzed. We propose to use an ontological approach to formalize the representation of domain knowledge used for semantic markup. The portal version of the Great Ukrainian Encyclopedia demonstrates the software implementation of the proposed approach.

## 1 Introduction

Every year, the volume of information resources (IR) created and used by mankind increases, and their structure becomes more complicated and heterogeneous. Here we define IR in the most general understanding as various data and infrastructure for its storage, retrieval, and processing that can be estranged from sources and authors. Now the majority of IRs are represented in digital forms that use computers, network equipment, and appropriate software. One of the promising directions for the effective use of information is the transition from data processing to knowledge processing: we consider knowledge as a special case of information that can be used for generating new information.

The utility and quality of IRs are determined significantly by the means of knowledge representation (KR), rules and formalisms, their expressiveness and interoperability. KR in a broad sense is a complex theoretical problem, and the use of computers in knowledge processing make these studies more practical by many additional requirements and limitations.

Encyclopedic IRs need convenient, clear, and unambiguous abilities for processing heterogeneous knowledge with a complex structure, and the use of appropriate KR becomes a key factor in creating of efficient intelligent information system (IIS) based on this IR, taking into account the needs of users.

Such a KR system has to reflect adequately the real world and its various regularities and to ensure an acceptable processing speed (execution of requests, search for information with conditions, etc.) These demands cause the study of theoretical principles and practical means of knowledge representation.

## 2 Information and Knowledge as a Resource

(Moody and Walsh, 1999) define some specific properties of the information that distinguish it from other resources:

- Information can be shared, and this resource is not exhaustive (although information sharing may reduce ownership benefits);
- The usefulness of information can increase with use;
- The value of information increases with accuracy;

- The value of information increases in combination with other information;
- The value of information is situational and depends on its actuality;
- More information is not better for problem solving, the usefulness of information depends on its relevance;
- Information is based on previous information, the combination of which often stimulates new understanding.

All these properties are also inherent in knowledge, but knowledge has some more unique properties that are very important for KR:

- Knowledge has a structure, which itself is an important element of IR;
- The usefulness of knowledge is determined not only by its formal characteristics but also by its pertinence to a specific task and an adequate representation of the real world by task-important characteristics;
- The value of knowledge is determined not only by its quality and expressiveness but also by connections with other knowledge;
- The ability to search into knowledge-oriented IR at the semantic level causes the effectiveness of its use.

Relational database systems are oriented on the closed world structure where information about the domain is known and limited, but they do not work well with the Web-oriented IRs in the open dynamic environment. Restructuring knowledge or adding new processing tools to an existing closed-world system can be difficult. Therefore, the development of open knowledge-oriented IRs needs models, methods, and tools that maintain a structure of relations between content elements of the knowledge representation system (KRS). Such a KRS should satisfy both the current needs of the IR users and be suitable for development and adaptation to their new activities.

### 3 Representation of Knowledge in Open Information Resources

Knowledge representation (KR) is a subfield of artificial intelligence (AI) that deals with the notation of information about the world in a form that a computer system can use to solve various tasks. Many researchers analyze KR from different points of view (Bergman, 2018) and propose various approaches to their use for knowledge management in intelligent information systems (IIS). Such approaches depend on IIS domain, purposes of analysis, role of KR, type of processed information, expressiveness requirements, etc. Peirce (1985) bases the study of signs for KR from the point of view of semiotics, where every intelligent operation includes a triad "object – representation – interpretation" where representation is one of the characteristics of information meaning.

Information is a statement conveyed by means of signs, most often symbols, about objects in a particular domain. The transition from information to knowledge occurs when we trust the information and intend to act on it. If the response to the observation and evaluation of symbols is a certain action or intention to act, then these symbols can be considered knowledge.

The criteria for evaluating certain information are its coherence represented by the state of logical sequential connections that provides a logical structure for the intelligent integration of various elements. Another criterion for evaluating information is its ambiguity, which is a frequent source of errors arising from incorrect identification of objects and relations. That is why disambiguation is such an important task in semantic systems. Context can be used to resolve ambiguities: the same information can be used in different ways or have different importance depending on the circumstances.

Structure and various patterns are the special cases of knowledge that can be acquired and recognized by analysis of KR for various IRs. KR of natural language (NL) texts can be enriched in addition to grammar rules, syntax, punctuation, etc. by more formal means for the

representation of sentence structure and semantics. However, language-structuring tools are not limited to grammar and need to take into account the meaning of language units and their aggregates. Semantics refers to those aspects of language processing that allow recording, characterizing, and analyzing its content. With the advent of thesauri, linguistic databases such as WordNet, and knowledge graphs that semantically link words and terms, automated processing of language at the semantic level by computers is becoming possible.

The structure in a broad sense can refer to objects (both physical and abstract) and their attributes. This term characterizes a variety of ways in which objects are arranged or related to each other, and is used as a means of describing components and relations (Rogushina, 2019). The concept of structure is close to the concept of patterns that is supported by various mathematical models.

In IT, pattern recognition is a check for certain sequences of features included in this pattern. Algorithms used for the recognition of various patterns depend on the domain (speech recognition, image recognition, cryptography, data compression, machine learning, etc.) and representation form (NL text, images, video, databases, etc.).

When considering the knowledge and choosing means of its representation, one must also take into account some other characteristics of data that need to be taken into account in systems of knowledge representation (KRSs):

- Knowledge is never complete – the acquisition and use of knowledge is a process that can never be considered complete, and therefore the KRS has to provide means to replenish and update knowledge;
- Knowledge can be stored in different forms: a large part of available information is contained in semi-structured sources (for example, in NL documents and hypertext), and this increases the importance of metadata processing;
- Knowledge can be obtained from various sources, from the environment or other IISs, and KRS has to support the import and acquisition of knowledge from large volumes of available information;
- Structures of knowledge develop as a result of receiving new information, and KRS requires means of developing and improving these structures, as well as checking previously created knowledge, expanding their description and clarifying definitions: it is necessary to clearly define how KRS provides an expansion of the set of concepts and complicating the structure of their relations;
- Knowledge is a specific type of information that allows one to establish meaningful connections between disparate fragments of relevant information, and the number of such connections increases with the growth of the knowledge base, i.e., KRSs should support the possibility of adding new connections without negatively affecting the characteristics of existing knowledge.

#### **4 Knowledge Representation Roles**

Davis proposes to define KR in terms of five different roles of knowledge use (Davis et al., 1993):

1. a surrogate, i.e., a substitute for the thing itself, used to enable an entity to determine consequences by reasoning about the world, rather than acting in it.
2. a set of ontological obligations, i.e., an answer to the question about the terms for reasoning about the world.
3. a theory of intelligent reasoning.
4. an environment for pragmatically efficient computing.
5. environment for expression of the beliefs about the world.

Each role provides different requirements for KR by different sets of representation properties. KR roles provide a structure that can be used to characterize various representations. This approach can be useful both for theoretical research and practice. In this work, we consider the second role associated with the use of ontologies in the analysis of KR expressiveness.

All representations of knowledge are certain approximations to reality, and each of them pays attention to some properties and ignores others. Such a choice of important properties means accepting a set of ontological commitments about the world.

These commitments and their focus are essential because the relevant choice of commitments allows us to focus on those aspects of the world that we consider relevant to some particular task. Each domain ontology offers its own way of seeing some part of the world.

Ontologies, in turn, can be represented in different languages and in different notations, but from the KR point of view the essential information is not the representation form, but its content defined by the set of concepts for reasoning and relations between them.

Ontological obligations, determined by the choice of domain ontology, can lead to different views of the task and ways of its solving. Rule-based systems view the world in terms of “attribute-object-value” triples and the plausible reasoning rules that connect them, while frames force us to think in terms of prototypical objects. For example, in medicine, rule-based KR (such as MYCIN) relates a symptom to a disease, and prototype-based KR (such as INTERNIST) matches symptoms to specific cases related to different types of diagnosis. Thus, each KRS provides its own perspective on what is important to pay attention to and what can be ignored. Frames suggest prototypes and taxonomies but do not specify exactly which domain objects are selected as prototypes, and rules suggest reasoning in terms of plausible inference, but do not specify which form of plausible inferences is relevant to the task.

It is important to remember that KR is not reduced to data structure because such structure is only one of the KR components.

For example, a semantic network is an example of KR, and a graph characterizing this network is a data structure. These are different entity types, even though one is invariably used to implement the other, precisely because the network expresses domain semantics that constrains the data structure: a graph of a semantic network that describes family relations cannot contain cycles, although, in general, the topology of graphs (i.e., data structures) has no such restrictions and can contain arbitrary cycles.

Practical decisions about the organization of knowledge in IISs are based on the assumption that information is clearly interconnected in many different ways and not structured as a set of independent or only implicitly related facts. For example, frame theory recommends reasoning by matching stereotypes that are described in the structure of frames and in the system of interaction between frames, and matching stereotypes with specific cases.

Development of IISs uses KR roles to evaluate knowledge models according to various criteria:

- conformity of the surrogate – modeling of the real-world objects in KR that provides the requirements for characteristics that are necessary for solving practical problems;
- definition of ontological obligations of KR that allow reflecting adequately the semantics of the real world;
- the acceptability of KR for performing intelligent inference that solves the user's problem;
- the acceptable computational complexity of intelligent inference and retrieval based on KR;
- sufficient expressiveness of KR for formulating ideas about the world and requests that can be performed for intelligent inference.

## 5 Practical Use of KR Based on Wiki Technology

Let us consider in more detail the approaches to the KR organization for the creation of semantic encyclopedic Wiki portals. Various Wiki-encyclopedias (the most famous example of which is Wikipedia) are widely used for various purposes, but most of them do not use semantic extensions, and this significantly limits both the expressiveness of information representation and the possibilities of retrieval of context elements.

Wiki technologies provide distributed processing of information on the Web and allow users to edit the content. One of the most popular software for Wiki systems is MediaWiki, which is used by Wikipedia, Wikidata, and Wikibooks projects. MediaWiki is distinguished by a high-quality and convenient content editor. From the KR point of view, Wiki technologies provide the following syntactic constructs for knowledge structuring:

- categories that allow grouping Wiki pages according to arbitrary principles (each page can belong to any number of categories, and the relations of partial ordering allow creating sets of taxonomies of these categories);
- links between Wiki pages (the semantics of these links is not described formally and can be determined only by context);
- page namespaces;
- templates that specify the representation of information about objects of similar types, but do not formalize the properties of these objects.

Such KR significantly limits the retrieval possibilities: user can search for information by keywords (in the names of pages or in their content), and the result of such a search is a list of pages without their characteristics, or the user can find pages of a certain category (but it is impossible to find pages of several categories simultaneously).

KR of Wiki resources can be enriched by semantic plug-ins that add more instruments for the formal representation of content meaning.

Now various semantic plug-ins are used for semantic markup of NL Wiki components that support links between content elements with domain concepts with pertinent meaning.

For example, Semantic MediaWiki (SMW) is a MediaWiki plug-in that allows annotating Wiki content with explicitly defined information suitable for automatic processing. The main element of SMW is a semantic property that connects the Wiki page with data values or other Wiki pages. SMW allows the integration of information from different pages and supports semantic search based on an arbitrary set of categories and values of semantic properties (Rogushina and Grishanova, 2020). Results of knowledge-level search in SMW can be performed as ontological structures (Guarino, 1998) that can be used by other systems. Each SMW semantic property has a type, name, and value, as well as its own Wiki page in a special namespace that allows defining its place in the property hierarchy and documenting how this property should be used.

KR of semantic Wiki resources is enriched by:

- semantic properties of Wiki pages with values of various types of constants and links with Wiki pages;
- semantic queries (their parameters can be both categories and values of semantic properties, and the query results are not only pages that meet these conditions, but also values of the selected set of semantic properties of these pages);
- templates of typical information objects (TIOs) that provide a unified semantic markup (values of the semantic properties relevant to TIO are defined as parameters of the TIO template).

The model based on IO templates provides flexible development of the Wiki resource schema. Each template defines a class with freely chosen attributes, and instances of these templates can be freely embedded in other objects.

The use of templates for KR is appropriate to describe real-world objects with a similar structure, between which there are relations with different semantics. This situation is especially characteristic of encyclopedic IRs that contain information from a selected set of domains and with a clearly defined structure of elements.

Semantic properties and built-in queries of SMW, together with semantic forms and templates, is a powerful tools that can support modeling for a variety of project needs. But, from the KR point of view, they have certain limitations:

- you can build semantic queries only regarding the characteristics of one page (its properties and links to other pages), but it is impossible to analyze more complex chains of relations between pages;
- it is impossible to fix formally the range of values and the range of definitions for semantic links between Wiki pages by their categories of such pages or more complex conditions regarding the admissibility of their use – for example, we can specify that the value of the semantic property "Birthplace" should be a Wiki page, but SMW has no tools that would limit these values to pages of categories "City" and "Country");
- there are no means of determining the characteristics of semantic relations from the point of view of logical inference (transitivity, symmetry, reflexivity, etc.).

Thus, a certain part of domain knowledge used in the process of the IR knowledge system based on cannot be formalized explicitly and interoperably.

## 6 Use of Ontologies for KR of Wiki

We can develop some domain ontology (or use existing ones) that reflects the knowledge structure of the Wiki resource: ontology-based KR is much more expressive and allows both importing external knowledge bases and generating created descriptions in commonly accepted standardized formats (OWL and RDF). The use of ontologies requires much deeper theoretical research in the field of knowledge representation in order to guarantee obtaining query results (for example, SPARQL) in an acceptable time. In addition, the creation of ontologies requires the use of appropriate tools (such as the Protégé ontology editor and description languages. The requirements for such ontologies and methods of their construction for semantic Wiki resources are considered in (Rogushina, 2018).

The most important function of domain ontologies in KR of the Wiki resource deals with the formalization of such features of relations between Wiki pages that can be used by complex semantic queries. Ontological relations defined in the thesaurus for real domains (Rogushina and Gladun, 2021) can be grouped by similar meanings:

- taxonomic relation,
- cause and effect relation,
- equivalence relation,
- dependency relation,
- structural relation,
- topological relation,
- functional relation,
- chronological relation,
- similarity relation,
- conditional relation,
- target relation.

Relations from one group have similar properties that can be used for logical inference. SMW expressiveness is not sufficient to define these characteristics, but we can do it by ontological representation. Semantic properties used for SMW semantic markup of NL content can be

classified into one of these groups. It is important to understand that characteristics of different relations from one group can not be used for inference with individuals linked by these relations.

However, the importance of different groups of relations is not equal, and we can mark out fundamental relations that can be used as a base of some formal system expressing basic mathematical concepts. The examples of fundamental relations are: 1. the relation between a function, its argument, and the result (von Neumann set theory); 2. the naming relation ("is\_a"); 3. the relation "part-whole" (mereology); 4. the membership relation (ZF and NF set theory). We can define the formal properties of these classes of relations. For example, mereological relations are defined by such axioms: reflexivity – every object is a part of itself; transitivity – every part of a part of a whole is itself a part of that whole; antisymmetry – two separate objects cannot simultaneously be part of each other. We distinguish seven separate subclasses of mereological relations: 1. Component-Object; 2. Member-Collection; 3. Part-Mass; 4. Material-Object; 5. Property-Activity; 6. Stage-Process; 7. Locality-Region.

Therefore ontological representation of these groups of relations is more advisable and can be used for definitions of their subclasses. For example, semantic properties "is a member" and "is a subregion" are a subclass of mereological relation and have all characteristics of this relation. Inheritance of characteristics is an instrument for the definition of their subclasses, and use of domain-specific names for semantic properties helps the user in practical semantic markup of NL content (for example, by searching for relevant words in the content of the Wiki pages and their automated replacement by SMW constructions).

## **7 Relations of Elements as an Important KR Component**

The semantization of Wiki technologies provides an opportunity to define explicitly relations between Wiki pages, but each domain (and, accordingly, each information resource (IR) that deals with this domain) is defined by a fixed set of domain-specific relations. Some domain relations are the same for all areas (for example, "class-subclass"), and some others are very specific and are used to individual domains only and have no formally defined characteristics (for example, relation "has publications in journal").

But we can assign a group of relations of the "part-whole" type which are characteristic of various domains but contain different subtypes with similar characteristics and different semantics. Their definitions make an important component of the description of various domains (Rogushina and Gladun, 2022). The theoretical basis for determining such subtypes is mereology – a scientific direction that deals with the formalization of the description of relations between parts and the whole. In the creation of modern Wiki-based encyclopedic IRs, focused on functioning on the Web and interaction with other intellectual applications, it is advisable to use this theoretical basis to expand the functionality of such IRs.

## **8 Practical Application of Knowledge Representation in e-VUE**

The effective functioning of modern encyclopedias requires the organization of the system of its content in accordance with modern theoretical approaches in the management of distributed knowledge. An example of a Wiki resource based on SMW is the portal version of the Great Ukrainian Encyclopedia – e-VUE (vue.gov.org). This encyclopedia contains a lot of TIOs and their individuals connected by various semantic relations. The main part of e-VUE content is represented by NL articles, but most of them contain various multimedia elements – images, video, audio, etc. It integrates knowledge from various subject areas that are refreshed and enlarged continually. Therefore organization and management of e-VUE knowledge is a complex problem that needs advanced KR.

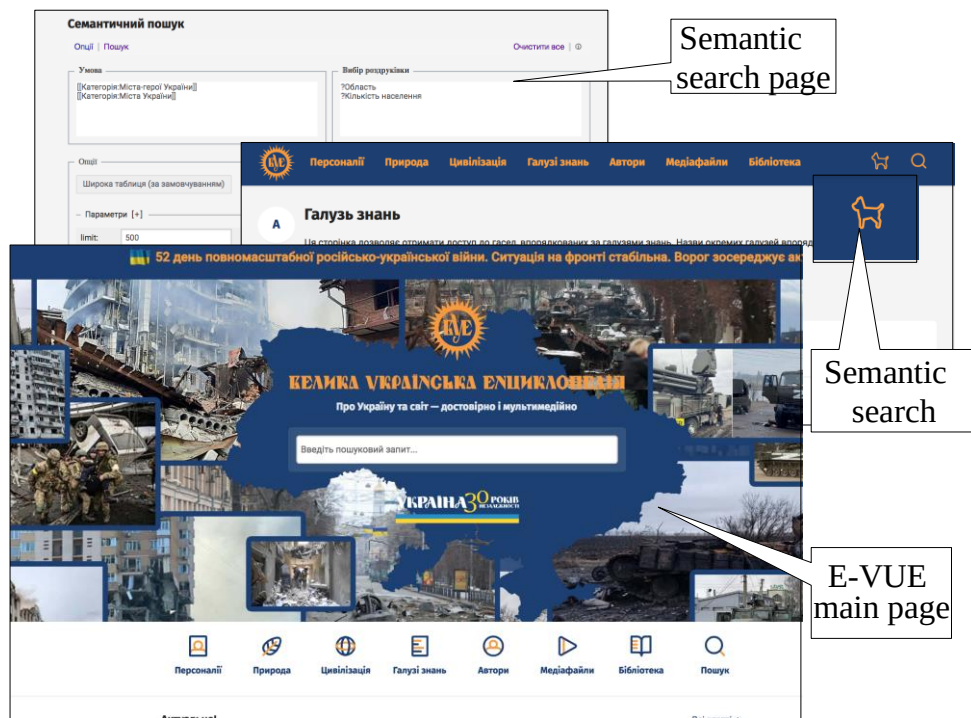


Figure 1. e-VUE semantic search

The specificity of the KR choice for e-VUE should take into account the fact that this encyclopedia covers many fields of knowledge characterized by TIOs, but terms with the same names used in e-VUE as TIO parameters can have different meanings for different domains. Relations between e-VUE concepts are very diverse but their names can be the same too. In different fields, the same terms can have completely different meanings, and therefore in the semantic markup, they need to be distinguished in some way. On the other hand, very similar concepts and relations can be named in different fields by distinct names. Moreover, the creation of e-VUE content is a cooperative process, and many experts can use different terminological systems.

Therefore, management of the e-VUE knowledge base requires models and tools for the KR organization. Such KR has to be more expressive, and we propose to use domain ontologies to formalize knowledge about the structure and restrictions of relations between e-VUE concepts.

The roles of KR for e-VUE are specified as follows:

1. the main replacement for the real-world concept in e-VUE is an encyclopedic article represented as a Wiki page that reflects only the most stable and objective characteristics of the corresponding concept (in addition to NL Wiki pages, e-VUE also includes various types of multimedia objects, but such objects are associated with certain articles and can be considered as their components); and typical IOs are supported by semantic templates.

2. the set of the e-VUE ontological obligations is represented by: 1. standard structural elements of Wiki-technologies (categories and hyperlinks); 2. SMW elements (semantic properties and templates of TIOs); and 3. Wiki-ontology that contains detailed characteristics of TIOs and semantic properties.

3. information from the Wiki-ontology is used for logical inference and execution of complex semantic queries that require greater expressiveness, and this ontology can be supplemented with new rules and axioms in the process of e-VUE content development (knowledge is acquired from semantic markup of NL texts). For example, if two Wiki pages, A and B, are linked by some semantic property, but Wiki ontology does not have a relation between respective concepts, A-concept and B-concept, then we can add it to ontology or remove it from Wiki IR.

4. e-VUE KR ensures the suitability of this environment for pragmatically effective calculations, that is, semantic queries (formed by users or built in the Wiki pages) based on markup elements can be performed in an acceptable time.

5. e-VUE contains a set of semantic relations between individual encyclopedia articles, which is sufficient to reflect the properties of these articles that are necessary for semantic search, that is, the markup language allows users to formulate their information needs.

e-VUE provides users with semantic access to information on the basis of KR in three ways:

1. the special page "Semantic search", where query parameters are entered in the appropriate fields, and users have not to know the specific syntax of the search language (it is enough to know fields for query and names of query requirements);

2. semantic queries in the specialized SMW search language and embedded in other pages;

3. queries that use API (by the development of a special code).

Queries based on API extend search possibilities into NL content of Wiki resources, but we have to understand that development of such queries is much more difficult as compared with SMW queries, and their execution requires significantly more computational power due to the need to analyze all unstructured NL content instead of elements of SMW semantic markup. Therefore, it is advisable to turn to such queries only in those cases when they become a unique source of the necessary information that cannot be obtained in other ways.

The use of ontologies in e-VUE allows describing individual semantic properties of encyclopedic articles and the relations between them much more clearly and formally. For example, if you build a hierarchy of e-VUE TIOs and categories in Protégé, it is easy to visualize it in the form of a tree (Fig. 2). In addition, the use of this ontology greatly simplifies the creation of semantic queries – this ontology contains all correct names of relations between pages and other semantic properties of pages that can be searched.

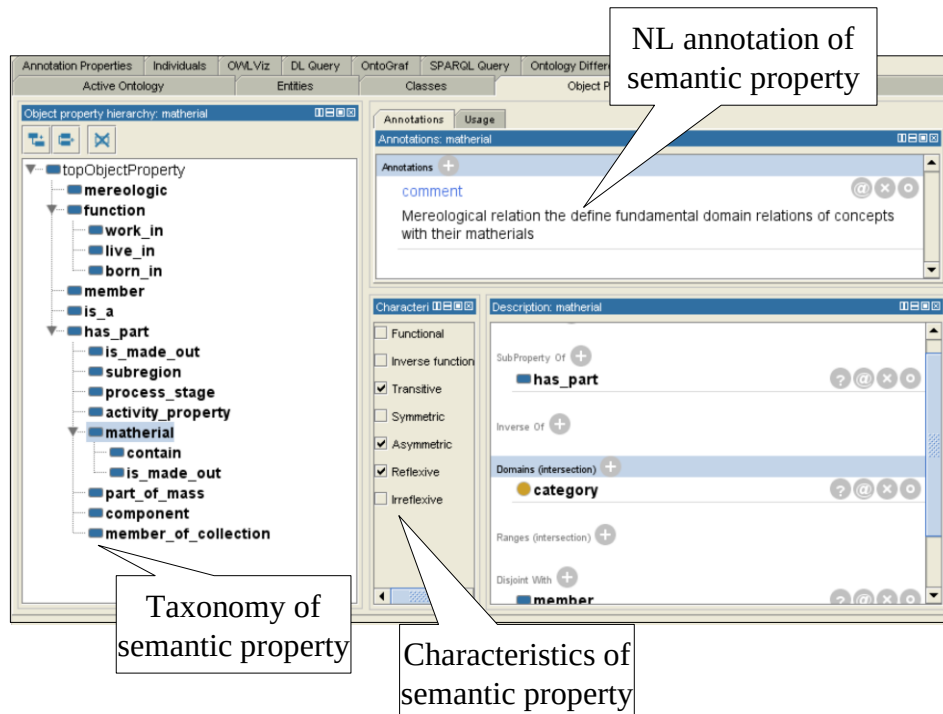


Figure 2. Characteristics of e-VUE semantic properties by Wiki ontology

## 9 Conclusions

Development of complex IRs that integrate the knowledge of different domains needs in relevant KR means that, on the one hand, have sufficient expressive ability to formalize all important

characteristics of the domain and support the search for information at the semantic level, and on the other hand, allow ensuring reasonable request execution time and be quite understandable for developers and users of the resource.

The experience of the e-VUE highlighted the problems related to knowledge management and proved the need for a detailed analysis of the KR roles in this project that affect the improvement of the e-VUE and correspond to the specifics of the semantic Wiki technology. This study, represented by the analysis of the KR expressive means of NL component of the e-VUE content, substantiates the possible ways of retrieval and integrating of information, as well as evaluates the expediency and effectiveness of various ways of their practical implementation. The proposed approach to the classification of relations between concepts by their characteristics can be used for the development of other multidiscipline information resources that use semantic Wiki technologies for NL content structuring.

### References

- Michael K. Bergman. (2018). *A Knowledge Representation Practitioner*. Springer International Publishing, 464 pp., ISBN 978-3-319-98091-1.
- Randall Davis, Howard Shrobe, and Peter Szolovits. (1993). What is a knowledge representation? *AI magazine*, 14(1): 17-28.
- Nicola Guarino. (1998). Formal Ontology in Information Systems. In *Proc. of FOIS'98*, pages 3-15, Trento, Italy.
- Daniel L. Moody and Peter Walsh. (1999). Measuring The Value Of Information: An Asset Valuation Approach. In *Proceedings of the Seventh European Conference on Information Systems (ECIS'99)*, pages 496-512, Frederiksberg, Denmark.
- Charles S. Peirce. (1990). Objects of Thought and Representation, *Nous*, Vol. 24, No. 3, pages 375-395. <https://doi.org/10.2307/2215771>
- Julia Rogushina. (2018). Theoretical foundations of the application of ontologies for the semanticization of Web resources. *Problems of programming*, 2-3: 197-203.
- Julia Rogushina. (2019). Use of Semantic Similarity Estimates for Unstructured Data Analysis. In *CEUR Vol-2577, Selected Papers of the XIX International Scientific and Practical Conference "Information Technologies and Security" (ITS 2019)*, pages 246-258, Kyiv, Ukraine. <http://ceur-ws.org/Vol-2577/paper20.pdf>.
- Julia Rogushina and Anatoly Gladun. (2021). Task Thesaurus as a Tool for Modeling of User Information Needs. In *New Perspectives on Enterprise Decision-Making Applying Artificial Intelligence Techniques*, pages. 385-403, Springer, Cham. [https://doi.org/10.1007/978-3-030-71115-3\\_17](https://doi.org/10.1007/978-3-030-71115-3_17).
- Julia Rogushina and Anatoly Gladun. (2022). Mereological Approach for Formation of "Part-Whole" Relations between Pages of a Semantic Wiki-resource. In *Information Technologies and Security ITS 2021, CEUR*, 3241: 237-247. <http://ceur-ws.org/Vol-3241/paper22.pdf>.
- Julia Rogushina and Irina Grishanova. (2020). Ontological methods and tools for semantic extension of the media WIKI. *Problems of programming*, 2-3: 61-73. <http://pp.isoftware.kiev.ua/ojs1/article/download/398/437>. DOI:10.15407/pp2020.02-03.061.
- T. Tudorache, C. Nyulas, N. F. Noy, and M. A. Musen. (2013). WebProtégé: A collaborative ontology editor and knowledge acquisition tool for the web. *Semantic web*, 4(1), 89-99. – <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3691821/>.

## Index of Authors

|                           |                                |
|---------------------------|--------------------------------|
| Arvinte Rareș, 73         | Micu Roxana, 49                |
| Avram Andrei-Marius, 7    | Mincu Eugenia, 123             |
| Bejinariu Silviu-Ioan, 81 | Mitkov Ruslan, 1               |
| Bibiri Anca-Diana, 93     | Moldovan Andrei, 15            |
| Bumbu Tudor, 141          | Moruz Maria, 93                |
| Burțeva Lyudmila, 141     | Moruz Mihai Alex, 93, 105, 131 |
| Ciora Radu A., 27         | Nevaci Manuela, 81             |
| Cojocaru Svetlana, 4, 141 | Ogrodniczuk Maciej, 2          |
| Colesnicov Alexandru, 141 | Orăsan Constantin, 3           |
| Costea Daria, 15          | Pădurariu Cristian, 115        |
| Cristea Dan, 115          | Păiș Vasile, 7, 15, 49         |
| Dănciulescu Daniela, 35   | Parahonco Alexandr, 153        |
| Druță Georgiana, 35       | Petic Mircea, 153              |
| Fellbaum Christiane, 2    | Rebeja Petru, 115              |
| Floarea Carmen Irina, 81  | Rogushina Julia, 161           |
| Gasani Carol Luca, 49     | Saramandu Nicolae, 81          |
| Gîfu Daniela, 27          | Shvedova Mariia, 1             |
| Gînsac Ana-Maria, 105     | Strățianu Bianca Ionela, 59    |
| Iftene Adrian, 35, 59     | Trandabăț Diana, 73            |
| Iftene Florin, 81         | Tușiș Dan, 7                   |
| Luca Ramona, 81           | Ungureanu Mădălina, 105, 131   |
| Malahov Ludmila, 141      |                                |