

**PROCEEDINGS
OF THE 13TH INTERNATIONAL CONFERENCE “LINGUISTIC
RESOURCES AND TOOLS FOR PROCESSING THE ROMANIAN
LANGUAGE”
IAȘI, 22-23 NOVEMBER 2018**

Editors:

Vasile Păiș

Daniela Gîfu

Diana Trandabăț

Dan Cristea

Dan Tufiș

Organisers

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

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

Institute for Computer Science, Romanian Academy, Iași

Romanian Association of Computational Linguistics

Under the auspices of the Academy of Technical Sciences

ISSN 1843-911X

SCIENTIFIC COMMITTEE

Mihaela Colhon, Faculty of Mathematics and Natural Science, University of Craiova

Dan Cristea, Faculty of Computer Science, “Alexandru Ioan Cuza” University and Institute for Computer Science, Romanian Academy, Iași

Daniela Gîfu, Faculty of Computer Science, “Alexandru Ioan Cuza” University and Institute for Computer Science, Romanian Academy, Iași

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

Verginica Barbu Mititelu, Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, Bucharest

Mihai Alex Moruz, Faculty of Computer Science, “Alexandru Ioan Cuza” University and Institute for Computer Science, Romanian Academy, Iași

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

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

Elena Isabelle Tamba, “A. Philippide” Institute for Romanian Philology, Romanian Academy, Iași

Horia-Nicolai Teodorescu, ARFI-IIT & Faculty of Electronics, Telecommunications and Information Technology, “Gheorghe Asachi” Technical University of Iași

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

Ștefan Trăușan-Matu, Computer Science Department, “POLITEHNICA” University of Bucharest & Research Institute for Artificial Intelligence “Mihai Drăgănescu”, Romanian Academy, Bucharest;

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

Marius Zbancioc, Institute for Computer Science, Romanian Academy, Iași

ORGANISING COMMITTEE

Anca Diana Bibiri, Department of Interdisciplinary Research – Humanities and Social Sciences, “Alexandru Ioan Cuza” University of Iași

Dan Cristea, Faculty of Computer Science, “Alexandru Ioan Cuza” University and Institute for Computer Science, Romanian Academy, Iași

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

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

Daniela Gîfu, Faculty of Computer Science, “Alexandru Ioan Cuza” University and Institute for Computer Science, Romanian Academy, Iași

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

Mihaela Onofrei, Institute for Computer Science, Romanian Academy, Iași

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

Andrei Scutelnicu, Faculty of Computer Science, “Alexandru Ioan Cuza” University and Institute for Computer Science, Romanian Academy, Iași

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

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

TABLE OF CONTENTS

TABLE OF CONTENTS.....	V
FOREWORD	VI
CHAPTER 1 SPEECH PROCESSING	1
A Psycholinguistic View in Romanian Intonation Interpretation	3
<i>Doina Jitcă</i>	
Nonlinear System for Romanian Voice Processing	13
<i>Carmen Grigoraş, Victor Grigoraş, Vasile Apopei</i>	
Comparison of I-Vector And GMM-UBM Speaker Recognition on a Romanian Large Speech Corpus.....	25
<i>Alexandru-Lucian Georgescu, Horia Cucu, Corneliu Burileanu</i>	
A Comparison Between Traditional Machine Learning Approaches and Deep Neural Networks for Text Processing in Romanian.....	33
<i>Adriana Stan, Mircea Giurgiu</i>	
Prosodic Phrasing between Subjective Perception and Objectivity	43
<i>Vasile Apopei, Otilia Paduraru</i>	
CHAPTER 2 TEXT PROCESSING	51
On Text Processing after OCR.....	53
<i>Alexandru Colesnicov, Svetlana Cojocaru, Ludmila Malahov, Lyudmila Burtseva</i>	
Romanian Diacritics Restoration using Recurrent Neural Networks	61
<i>Stefan Ruseti, Teodor-Mihai Cotet, Mihai Dascalu</i>	
TEPROLIN: an Extensible, Online Text Preprocessing Platform for Romanian.....	69
<i>Radu Ion</i>	
How to Find out Whether the Romanian Language Was Influenced by the Two Historical Unions?	77
<i>Dan Cristea, Daniela Gîfu, Svetlana Cojocaru, Alexandru Colesnicov, Ludmila Malahov, Marius Popescu, Mihaela Onofrei, Cecilia Bolea</i>	
CHAPTER 3 LINGUISTIC RESOURCES	89
More Romanian Word Embeddings from the ReTeRom Project	91
<i>Vasile Păiş, Dan Tuftiş</i>	
Valence Dictionary for Romanian Language in Printed Version and XML Format	101
<i>Ana-Maria Barbu</i>	
Nonstandard vs. RRT: Comparison of Two Romanian Corpora in UD	113
<i>Cătălina Mărănduc, Victoria Bobicev</i>	
Interlinking and Extending Large Lexical Resources for Romanian	125
<i>Mihai Alex Moruz, Andrei Scutelnicu, Dan Cristea</i>	
CHAPTER 4 APPLICATIONS.....	133
README - Improving Writing Skills in Romanian Language	135
<i>Maria-Dorinela Sirbu, Mihai Dascălu, Daniela Gîfu, Teodor-Mihai Cotet, Adrian Tosca, Stefan Trausan-Matu</i>	
New GIS Based Approaches for the Linguistic Atlases.....	147
<i>Silviu-Ioan Bejinariu, Vasile Apopei, Manuela Nevaci, Nicolae Saramandu</i>	
Let's Teach Computer Poetry: Automatic Rhyme Detection	159
<i>Victoria Bobicev, Cătălina Mărănduc</i>	
Extracting Actions from Romanian Instructions for IoT Devices.....	169
<i>Bianca I. Nenciu, Stefan Ruseti, Mihai Dascalu</i>	
Identifying Fake News on Twitter Using Naïve Bayes, SVM and Random Forest Distributed Algorithms	177
<i>Ciprian-Gabriel Cuşmaliuc, Lucia-Georgiana Coca, Adrian Iftene</i>	
INDEX OF AUTHORS.....	190

FOREWORD

At its inception, in 2001, the ConsILR Conference (traditional acronym for *Consortiul de Informatizare pentru Limba Română – Consortium of Informatization for Romanian Language*) was an initiative born in the Section of Information Science and Technology of the Romanian Academy, meant to bring together at the same table linguists and computational linguists, researchers in different fields of humanities, PhD students and master students in computational linguistics, pure linguistics and philologists, all those that have a major interest in the study of the Romanian language from a computational perspective. The series of events have run, with few exceptions, once every year, first in the format of a workshop, and since 2010 – as a conference. In order to reach wider visibility, the organisers decided to make the Conference itinerant and to publish its Proceedings in English. Thus, ConsILR is not strictly addressed to researchers working on Romanian language but also to other scientists, from any part of the world, which could find sources of inspiration in the models and techniques developed for our language and apply them for their own languages. Opening the gate for researchers working on languages other than Romanian to participate in the Conference and publish their results in this series of Proceedings, a reverse influence is also expected, namely that their work inspire scientists dealing with Romanian language.

As always, this year too, the organisers of the Conference *Linguistic Resources and Technologies for Romanian Language* are: the Faculty of Computer Science of the “Alexandru Ioan Cuza” University of Iași and two institutes of the Romanian Academy: the Research Institute for Artificial Intelligence “Mihai Drăgănescu”, in Bucharest, and the Institute for Computer Science, in Iași. We are honored that the Academy of Technical Sciences has offered again its high auspices to run the event under and, for the first time, the Romanian Association of Computational Linguistics was among the organisers.

In 2018, we brought the Conference again in Iași, the beautiful city located in the North-eastern Romania, in the heart of the province of Moldavia, a city full of students, and whose buildings and streets, stretching over seven rolling hills (just like Rome), offer visitors memorable history lessons at their strolling through the town. ConsILR-2018 is the 13th in the series and, because this year Romania commemorates 100 years since the union of Transylvania, Bessarabia, and Bukovina with the Romanian Kingdom, we decided to dedicate it to this major anniversary.

As usual, our conference aims to advance the level of computerisation of the Romanian language with new resources and improved processing tools. The volume includes 18 papers, covering topics of 4 major areas, which give also the names of the 4 chapters: Speech Processing, Text Processing, Linguistic Resources and Applications.

Just like in other volumes of the series, the first chapter is dedicated to Speech technology. The five papers are contributed by authors affiliated to three main academic centres of speech processing research in Romania: Institute of Computer Science, in the Iași branch of the Romanian Academy, Speech and Dialogue Research Laboratory, in the University Politehnica of Bucharest, and Communications Department, in the Technical University of Cluj-Napoca. The topics cover psycholinguistic modelling applied to intonational contour comprehension, a comparison of analogue nonlinear systems with the aim of improving information extracted from speech signals, a comparison of the identity vectors against the traditional Gaussian Mixture Model – Universal Background Model applied to speaker verification and identification, a proposal for a new perspective on the relationship between speech rhythm and prosodic phrasing, and a comparison between traditional and deep learning techniques applied to text processing issues involved in text2speech synthesis techniques.

Chapter 2 includes 4 papers in the area of Text Processing, contributed by researchers from Chișinău – Institute of Mathematics and Computer Science, Bucharest – Computer Science Department of the University Politehnica of Bucharest, Faculty of Mathematics and Computer Science of Bucharest University and Research Institute for Artificial Intelligence “Mihai Drăgănescu” of Romanian

Academy, and Iași – Faculty of Computer Science of “Alexandru Ioan Cuza” University and Institute of Computer Science of Romanian Academy. Their contributions range from developments of post-OCR processes for recognition of old Romanian texts written in the Cyrillic script, to diacritics restoration using recurrent neural networks, a presentation of the TEPROLIN text pre-processing platform, part of the ReTeRom project, and a theme that resonates well with the dedication of the present volume: the design of a methodology of studying the influence of the two historical acts of unification of Romanian provinces on the evolution of the modern Romanian language.

The 3rd Chapter, Linguistic Resources, is contributed by researchers from Bucharest, Romanian Academy – “Mihai Drăgănescu” AI Institute and Institute of Linguistics, from Iași – Faculty of Computer Science in A.I. Cuza University, and from Chișinău – Technical University of Moldova. The topics of the 4 papers are: an open-access large library of ready-to-use word embedding sets, the encoding structure of a verb valence dictionary for Romanian, the new release of the non-standard Romanian Treebank in comparison with the Romanian Reference Treebank, and new research directions for the extension and improvement of CoRoLa (the Corpus of Contemporary Romanian Language): adding of new layers of annotation and a standard that allow interlinking of different types of lexical resources for Romanian.

Finally, Chapter 4, dedicated to Applications, groups 5 papers signed by researchers in Bucharest – Computer Science Department of Politehnica University, “Mihai Drăgănescu” AI Institute, “Iorgu Iordan – Al. Rosetii” Institute of Linguistics and Cognos Business Consulting S.R.L., as well as from Iași – Faculty of Computer Science in A.I. Cuza University, Institute of Computer Science of the Romanian Academy, and from Chișinău – Technical University of Moldova. As always, a diversity of fresh topics can be found in the Applications part of the volume: the description of a new release of the rule-based extendable engine ReadMe, intended to do automatic writing skills evaluations, two approaches of geographical information systems, one developed for the Linguistic Atlas of the Aromanian Dialect and the other for the European Linguistic Atlas, an experimental study on how rhyme may be detected automatically in well formed poems, a speech2text interface implanted in an action uttering mobile application for IoT, and an analysis of different techniques of detecting fake news on Twitter.

We hope that the quality of the selected papers makes the present volume, alongside the volumes from previous editions, an interesting source of information on what is happening in the scientific community dealing with natural language, especially Romanian, a collection of articles very useful for researchers on AI, NLP and Linguistics, for professors and students and for anybody who is concerned with language use in the electronic media.

Acknowledgements

Part of the work published in this volume was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818 / 73PCCDI (ReTeRom), within PNCDI III and by the README project "Interactive and Innovative application for evaluating the readability of texts in Romanian Language and for improving users' writing styles", contract no. 114/15.09.2017, MySMIS 2014 code 119286. Many people contributed to the success of this event, and we express our sincere thanks to all of them. The Scientific Committee made up of international experts in the area of Linguistic Resources and Tools for Processing Romanian Language, spent time carefully reviewing all the proposals submitted to ConsILR-2018 to insure a qualitative improvement of the papers. The members of the Organizing Committee have enthusiastically assured the settings were appropriate for scientific networking.

November 2018
The editors

CHAPTER 1

SPEECH PROCESSING

A PSYCHOLINGUISTIC VIEW IN ROMANIAN INTONATION INTERPRETATION

DOINA JITCĂ

Institute of Computer Science of the Romanian Academy, Iasi-Branch
doina.jitca@iit.academiaromana-is.ro

Abstract

The paper presents an Information Structure (IS) model that can explain how F0 contours organize utterances at the information packaging level. On the IS model basis utterances are divided into binary units having a hierarchical organization. Binary units or contrast units (CUs) consists in a pair of two functionally contrasted elements marked by prosodic words with different phonetic cues. The psycholinguistic model defines two levels of Information Structure: CU_topic-CU_focus and CU_predicat – CU_argument. These pragmatic IS functions have neuronal equivalence at the cortex level and each category of semantic models has to be related to one of these pragmatic IS functions in order to understand how a semantic event of a certain category is implemented at the information packaging level. The paper introduces the IS model view in intonational contour analysis in order to improve the intonational contour comprehension. The paper exemplifies IS analysis by using two Romanian Yes-No question contours varieties. IS descriptions of IS partition are based on the IS functional categories defined by the model. The paper concludes that despite the diatopic variation of Romanian YNQ contour at the phonological level, the two variants shows invariable IS features related to functional element distribution and nuclear element position.

Key words — IS partition, predicate-argument structure, psycholinguistic IS model, topic-focus.

1. Introduction

The paper introduces information structure (IS) analysis in intonational analysis and explains the role of intonation in building information units which pack words within utterances. Words are marked as functional elements at the IS level by using prosody. The semantic approaches of information structure apply them semantic functions by analysing the linguistic content of utterance. The semantic IS functions have problem in explaining different linguistic contexts; e.g. the presence of given referents with semantic “focus” function which is related to new elements as it is presented in (1). The pronoun *him* is a given element but it is under the semantic “focus” produced by focus-sensitive particle only. The author in (Krifka and Mussan, 2012) claims that “given constituents can be in focus...it is possible to focus on pronoun”

(1) Mary only saw HIMF.

In this paper a psycholinguistic IS approach will be presents where IS functions have equivalent at the cortical speech production mechanism. Despite their psycholinguistic determination, functional IS constituents are marked accordingly to syntactic, semantic and discourse aspects. Thus, we will show that “focus” events are produced by a combination of

psycholinguistic IS functions. One combination may be that of nuclear and given information functions.

In the psycholinguistic IS model information unit is always a binary one on any level of utterance tree. It has the constituents into a functional contrast. That's why we call it contrast unit (CU). If the tonal group of one prosodic phrase has more than two constituents, words are grouped in a hierarchical manner by a binary tree. Binary information units and their hierarchical architecture lack in semantic IS models and IS description can not be detailed under lowest IS levels. Halliday (1967) or Steedman (1991; 2000) for English intonation relates the whole tonal group to one information unit and select only one constituent as being relevant ("focused") due to its acoustic salience.

Phonological approaches describe intonational phrases by sequences of events related to accented syllables or edge tones. The IS approach based on the psychological model treats words related to their prosodic words as functional events. IS functional constituents are identified by pragmatic analyses of prosodic words at the intonational contour level.

The paper introduces the psychological IS model view in analysing intonational contours in order to improve the description and the comprehension of their architecture. The intonational analysis from this point of view is also used in (Jitcă, 2017), (Jitcă, 2015). The papers do not modify the claims from previous works related to nuclear stress positions in Romanian intonational contours (Dascălu-Jinga, 1998), (Dascălu-Jinga, 2001), (Jitcă *et al.*, 2015). The psycholinguistic IS model gives an information structure basis in thinking IS functions and nuclear positions of utterance constituents.

The paper is structured as follows: Section 2 contains the IS model presentation with three aspects: the two information structure levels, the IS partition description system and nuclear accent rules for nuclear position identification. In section 3 the rules are used to analyse two Romanian Yes-No question varieties from the psycholinguistic IS viewpoint, aiming to compare there IS pattern descriptions. The two varieties are produced by a speaker from Moldova (Iași) and the second by a speaker from Transylvania (Bistrița). The author proposes that IS psycholinguistic features deduced from intonational contours to be taken into account in studying the unitary character of the Romanian intonation system.

2. The psycholinguistic model of Information Structure

The psycholinguistic model of Information Structure proposed in the paper redefines IS categories from semantic models where they are related to linguistic aspects and gives them sense within CUs that pack words at the utterance level.

Semantic models beginning with Prague School (Daneš, 1970) define two IS levels: i. The first level illustrates the "aboutness" principle which claims that sentence has to say "something", named comment or rheme, about an element, named "topic" or theme; ii. The second IS level conveys the relation between old (given) information (topic function) and new information (focus function) related to givenness and newness categories.

IS functional elements have roles at the information unit level. Units have two constituents and we name them contrast units (CU). Their constituents bear functions related to the following IS categories: CU_theme, CU_Rheme, CU_topic and CU_focus. Both IS levels are overlapped within each their information unit. In previous IS models, e.g. (Halliday, 1967)

(Steedman, 1991; 2000; 2014), the two IS levels are realized at different level of utterance tree: background-focus level is analysed at the tonal group level and theme-rheme at the linguistic phrase level (morpho - syntactic phrase, clause and sentence). In these approaches the topic-focus structure is replaced by background-focus structure to emphasize the importance of new information element due to its acoustical prominence. Topic elements may be present in the background part of respective unit. In (1) is presented the case of given information element which is under a “focus” function.

The psycholinguistic model reintroduces the pair of topic and focus functional elements at the information unit level, as CU_Focus and CU_Topic elements, relating them to specific acoustic features at the pragmatic level. CU_Focus and CU_Topic elements are not related to the existence or non-existence of acoustical prominence because any of the two constituents may be acoustically prominent elements in different contexts. CU_Topic and CU_Focus constituents are related to givenness and newness categories if the respective element is anchored within a concrete discourse context.

The acoustic prominence of CU_topic or CU_focus elements depend on the second function carried by them at the second IS level which implement the “aboutness” principle. The psycholinguistic IS model redefines the theme-rheme structure in terms of (von der Gabelentz, 1869)’s model. Thus, the theme becomes the psychological subject (PS) defined as "that to which the speaker directs the hearer's attention", and the rheme becomes the psychological predicate (PP) as "that which the speaker makes the hearer thinking about". Theme and rheme categories become CU_theme and CU_rheme in the psycholinguistic IS model view because they are the two parts of any CU.

2.1. The IS categories of the psycholinguistic model

The overlapping between the CU_topic-CU_focus and CU_theme and CU_rheme structure within CUs is possible because the two functions of one constituent are encoded by different acoustic features: the function at the CU_topic-CU_focus structural level is conveyed by temporal features of its F0 contour shape (slope, duration, etc.) and the function at the PS-PP or CU_theme-CU_rheme level, by the level of the dominant tone within the prosodic word of the respective element.

The CU_topic and CU_focus IS constituents are two paired and contrasted elements at the information packaging level that mark information unit. The CU_topic and CU_focus elements can anchor information unit within discourse context by also bearing givenness and newness meaning. The information packaging mechanism at the cortex level have two areas with different speech processing parameters for marking CU_topic and CU_focus elements with different temporal features.

The van der Gabelentz’ PS-PP structure is redefined within this psycholinguistic IS model by using the more general predicate-argument structure which can be implemented at the neurobiological level. It is used the term of CU_predicate to refer the predicate element (PS) at the CU level. The CU_predicate is carried by the constituent which bear the first reference of communication object within CU. The model uses the CU_argument function related to the PP function of the von der Gabelentz’ model that bears the added (specific) reference of communication “object” at the respective CU level. By using an analogy with the linguistic theme-rheme structure, we can say that CU_predicate is the “thematic” element and

CU_argument is the “rhematic” element within CU. The CU_predicate-CU_argument structure implements the “aboutness” principle within CUs.

The two contrasts, CU_topic-CU_focus and CU_predicate-CU_argument, within the same CU lead to constituents with two functions. This is possible at the neurolinguistic level by using two kinds of pitch features of contrasted prosodic marks: i. temporal features which involve shape and slope pitch variations. ii. Tonal levels which refer target (dominant) tones of the two prosodic marks.

The constituents with CU_topic function have F0 patterns with slow pitch movements and longer duration during accented syllable or they do not stop their rising movement in the end of the accented syllable. On the contrary, the CU_focus marks are abrupt pitch movement and shorter duration of syllable nucleus and tend to stop rising pitch movement in the end of the accented syllable. Other focus patterns have constant level pitch accent. The CU_predicate and CU_argument functions modulate the CU_topic and CU_focus realizations giving lower target (dominant) tones to the CU_predicate elements and higher target tones to the CU_argument element. This model is more general because it combines the two levels of IS functions in an independently manner and can describe semantic “focus” elements in the case they have new information function but also in the case of their given information function.

2.2. *The IS partition description system*

In the psycholinguistic IS model view utterance can be partitioned into CU hierarchy each CU having two-dimensional IS partition. One partition has two paired elements each of them having two functions at the two IS levels. The IS partition description system defines four labels related to the IS functional categories: T and F labels for CU_topic and CU_focus categories, and P and A labels for CU_predicate and CU_argument categories.

At the partition annotation level each element is associated with two functional labels linked by “+” and closed between round parentheses. IS partition of CUs relies into a sequence of two round parenthesis separated by “/”. In this view IS partition of CUs may be of one of the types described in (2).

(2) (A+T)/(P+F), (A+F)/(P+T), (P+T)/(A+F), (P+F)/(A+T).

All four IS partition variants are possible because the two structural levels are independent and each function of one level can be combined with any of the two functions of the second level. Thus, by using this IS model, analyses of different intonational realizations of the same text will always relate them different IS partition descriptions. If utterance tree has more than one CU, at each of them one IS partition must be applied. The coherence of IS partitions within utterance is implemented at the pragmatic (neural) level by nuclear attribute. The category of nuclear attribute is subsumed to the category of “focus” within semantic IS models. This is another advantage of the psycholinguistic model where the function of one element at the CU_topic-CU_focus structure level is dissociated from its function at the CU_predicate-CU_argument level and its nuclear attribute. This explains why in (1) the pronoun him can be annotated by CU_topic, CU_argument and nuclear function instead of applying semantic “focus” label which means CU_focus and CU_argument and nuclear function.

2.3. *Nuclear accent in the psycholinguistic IS model view*

At the linguistic level nuclear accent is related to the word to which the sentence refers or to the most important word. The IS model gives a psycholinguistic role to nuclear accent in organizing utterances as CU hierarchies. Nuclear element at one lower IS partition projects its IS functions to the whole respective CU, at the higher level IS partition. In this manner a large CU count as a single CU element within a higher CU. We exemplify in (3) with the partitioning of a SVO utterance. The utterance is structured by two nested CUs.

(3) $\{(F1+F2)^{\text{Subject}} / [(F1+F2)^{\text{Verb}} / (NF1+NF2)^{\text{Object}}]_{NF1+NF2}\}$ where $F1=P$ or A and $F2=T$ or F

The embedded CU corresponds to verbal phrase which has the verb as an element with generic $F1$ and $F2$ functions, and the object, as an element with contrasted $NF1$ and $NF2$ functions. At the global level the subject is an element with $F1$ and $F2$ functions and the whole verbal phrase has to be an element with $NF1$ and $NF2$ functions. This is possible because the nuclear accent within the imbricate IS partition is the element with $NF1$ and $NF2$ functions (the object). This example explains the importance of nuclear accent in organizing utterances in a coherent manner.

The nuclear accent marking in IS partition is another problem in intonational contour interpretation. There are two modalities used at the psycholinguistic level for encoding nuclear mark: by a negative prominence (a minimal low tone) or by a positive prominence (a maximal high tone) on one of accented syllables of utterance. We formulate in (4) and (5) two corresponding rules which identify nuclear words in the two prominence cases related to non-emphasized and emphasized contours.

(4) NSR_NE: In non-emphasized contour the nuclear accent is assigned to the global CU_predicate element related to the “low prominence” produced by the lowest target tone.

(5) NSR_E: In emphasized contour the nuclear accent is assigned to the global CU_argument element related to the “high prominence” of an emphasis produced by the highest target tone.

The prominence type of nuclear accent within $F0$ contour depends on the ratio between the tonal spaces of the two elements of IS partition. IS partitions with coordinated CU constituents (partial or total overlapping of their tonal spaces) produce nuclear event on the negative prominent element while those with subordinated CU constituents (non-overlapping of their tonal spaces) produce nuclear event on positive prominent element of partition.

3. IS analysis of two varieties of Romanian YNQ contours

In this section two varieties of Romanian Yes-No question (total interrogation) contours exemplify how IS partitions can be deduced from intonational contours. IS partition descriptions contain the two IS levels and nuclear position related to each CU of CU hierarchy.

The analysed contours correspond to information seeking (neutral) YNQ. Two intonational realizations are analysed representing two varieties from Moldavia (Iași) and Transylvania (Bistrița). The contours are very different at the phonological level and we aim to compare their characteristics at the IS level.

3.1. IS analysis of two varieties of Romanian YNQ contour

Two controversial opinions are presented in (Ladd, 2008) related to nuclear position (sentence-final vs. non-final position) in Romanian information-seeking YNQ contour. Xu

(2005) does not accept a nuclear accent without an acoustical prominence and thus, he can not accept the nuclear accent in the sentence-non-final position while Ladd (2008) admits that nuclear accent may have sentence-non-final position in Romanian contour case. Dascălu-Jinga (2001) uses the term of interrogative emphasis to characterize the phonological event with negative “prominence” on the verb. During the verb F0 contour holds constant low level.

In this section we explain how a nuclear accent can be produced without an acoustical salience. The IS descriptions resulted after the IS analysis of the two YNQ contours are compared in order to conclude about their common or different IS characteristics.

3.1.1. Information seeking YNQ contour- Moldavian variety

Information seeking YNQ contours is illustrated in Fig. 1 by using an utterance of the sentence *Aveți marmeladă?* (‘Do you have marmalade’) from the database of the paper (Jitcă *et al.*, 2015). The contour keeps low levels until the last accented syllable and then it rises up to very high levels during the object *marmeladă*. The contour has a falling pitch movement during the last non-accented syllable. The two prosodic words are structured within an IS partition described in (6).

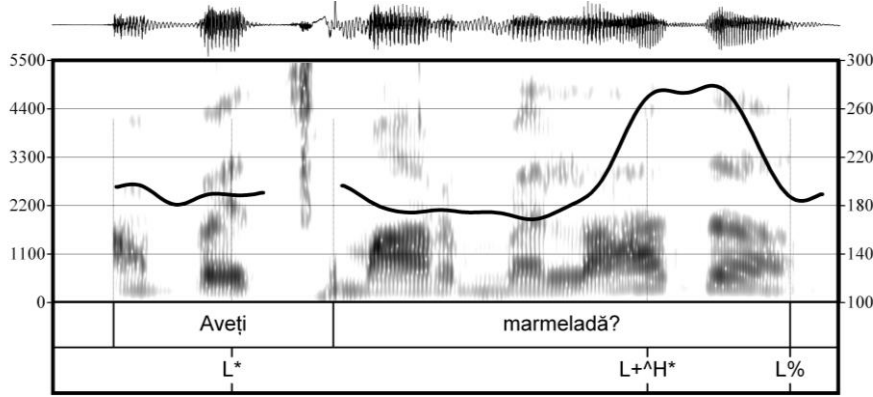


Figure 1. The F0 contour and the spectrogram of the utterance *Aveți marmeladă?* of one speaker from Moldova (Iași).

$$(6) \quad \{ (P+F)_N^{\text{Aveți}} L^* (A+T)^{\text{marmeladă}} L+^H L\% \}$$

The first constituent, the verb *aveți* has a low level pitch accent which gives him CU_predicate and focus functions within IS partition and it is annotated with P+F in (6). The second constituent of the noun *marmeladă*, has an L+^H* pitch accent with a very high target tone (extra-high tone) which gives him CU_argument function. CU_topic is the second functions and the object becomes an A+T element as it is described in (6).

An overlapping of their tonal spaces occurs as it can be seen in Fig. 1 where the tonal spaces of the two constituents are delimited by two rectangles. The minimal tone of the final constituent is smaller than that of the first constituent. Thus, the contour does not produce emphasis on the element with the highest target tone. In agreement with the NSR_NE rule nuclear accent is related to the negative prominence of the P+F element. In all three analysed F0 contours produced by different speakers a tonal space overlapping was identified. There is no valid neutral Romanian YNQ contour without this tonal characteristic to be accomplished.

This explains why the verb in sentence-non-final position bears nuclear accent which is named in (Dascălu-Jinga, 1998; 2001) interrogative emphasis. Romanian speakers feel the sentence accent on the verb even if it is realized by negative prominence (Ladd, 2008).

3.1.2. Information seeking YNQ contour- NE Transylvanian variety

The information seeking YNQ contours, the NE Transylvanian variety, is illustrated in Fig. 2 by using an utterance of the sentence *Aveți marmeladă?* It is extracted from the database of the paper (Jiță et al., 2015). The characteristic of this contour relies in a rising pitch movement after the low level during the accented syllable of the verb. The rising occurs on the non-accented syllables of the noun *marmeladă*.

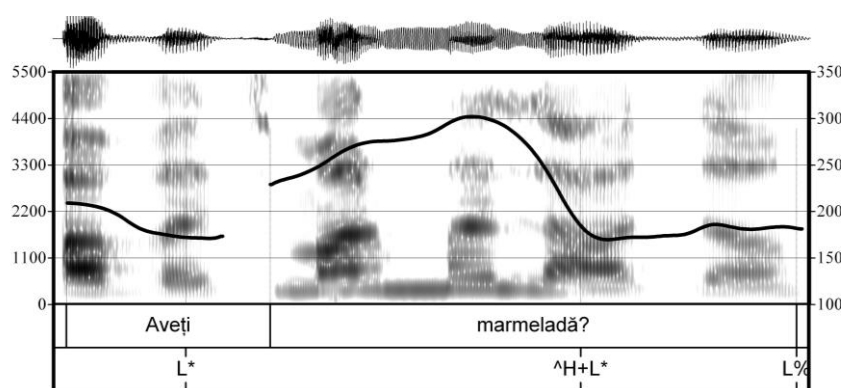


Figure 2. The F0 contour and the spectrogram of the utterance *Aveți marmeladă?* of one speaker from NE Transylvania (Bistrița).

In Fig. 2 it can be seen the overlapping of the two tonal spaces. This relates the nuclear accent to the negative prominence of the CU_predicate element. The CU_predicate function is carried by the verb which keeps low level during the whole accented syllable while the noun begins its accented syllable at high level of 250 Hz. During 0.77 msec. of the ^H+L* pitch accent, the F0 contour keeps higher levels than the low level on the verb. With other words, the negative prominence of the verb is more salient than that of the last word. This explains why the verb bears the nucleus and the sentence-final element has a high dominant tone becoming a CU_argument element as it is described in (7). The slow last falling marks the noun for the topic function. The negative prominence gives the CU_predicate function to the verb and the abrupt rising pitch movement after the accented syllable marks it for the focus function.

$$(7) \{ (P+F)_N^{Aveți} L^* / (A+T)^{marmeladă} \wedge H+L^* L\% \}$$

The Romanian YNQ contours were also analysed by foreign linguists interested to explain the low prominence of nucleus and the role of acoustical salient pitch accent. In all cases phonological views are used. Ladd (1983) describes the contour with ascending phrase-final contour (Moldavia) by L* H-L% sequence and the Transylvanian variant by the L*+H H-L% sequence. Grice et al. (2000) gives the same description for both Moldavian and Transylvanian contours by using the L* H- L%. In both papers the last high pitch accent is interpreted as a phrase accent, a phonological event which is considered to have an implicit or explicit presence before the boundary tone. The authors were interested to give general descriptions for similar YNQ contours in many languages and they do not intend to deduce

the IS structural meaning. They do not perceive the two constituents of the verb and of the object, as thematic and rhematic elements. They are more interested by the high pitch accent and they try to find it a role at the phonological level.

The psychological IS model view gives an IS role to the last high pitch accent. It produces a dominant high tone which marks the last word for the CU_argument function leading to a negative prominence and a CU_predicate function on the first word.

A comparison between the descriptions of the two contours in (6) and (7) leads to the conclusion that despite their phonological differences, they have the same IS partitions and nuclear positions. This fact can be an argument that they belong to the intonational system of a unitary language which have for information-seeking YNQ, IS partitions of (P+F)/(A+T) and sentence-non-final position nucleus.

4. Conclusions

The paper present a IS model that explains the information packaging within utterances and then use the related IS annotation system in producing Romanian intonational contour descriptions at the IS level. The IS model perspective is a psycholinguistic one that defines IS functions at the information packaging unit level. The IS functions are directly deduced at the pragmatic level from intonational pattern of IS constituents. IS partitions contain two overlapped contrasts: CU_predicate-CU_argument and CU_topic-CU_focus. The IS model contains nuclear accent rules related to the two modalities in marking nuclear accent: by low and high prominence.

Two Romanian YNQ contours are analysed in the IS model view that justifies the sentence-non-final position of their nuclear accents and describes their IS partitions. Foreign linguists were interested to give an explanation of the non-nuclear attribute of the last pitch accent with a very high target tone. They fitted the pitch accent with the phonological category of accent phrase (H- in the ToBI annotation).

In the IS view the F0 contours of the two YNQs have a prominent contrast between the low-level CU_predicate constituent (the verb) and the high level CU_Argument constituent (the object) which is marked by the last pitch accent. Intonational analyses as those presented in section 3 can be performed on the main Romanian contour types in standard Romanian and different other varieties in order to deduce their IS pattern characteristics. The conclusions may offer psycholinguistic arguments in studding the unitary character of Romanian intonational system.

References

- Daneš, F. (1970). One Instance of Prague School Methodology: Functional Analysis of Utterance and Text. In *Method and Theory in Linguistics*, P. Garvin (ed.), Paris: The Hague, 132-146.
- Dascălu-Jinga, L. (1998). Intonation in Romanian. In *Intonation Systems: A survey of twenty languages*, D. Hrist and A. Di Cristo (Eds), Cambridge: Cambridge University Press, pp. 239-261.
- Dascălu-Jinga, L. (2001). *Melodia vorbirii în limba română*. București: Univers Enciclopedic.

- Frota, S., Prieto, P. (2015). Intonation in Romance. In *Intonation in Romance*, Frota, S. & Prieto, P. (Eds), Oxford: Oxford University Press.
- von der Gabelentz, G. (1869). *Ideen zu einer vergleichenden Syntax. Wort und Satzstellung Zeitschrift für Völkerpsychologie und Sprachwissenschaft*, 6, 376-384.
- Grice, M., Ladd, D. R., Arvaniti, A., (2000). On the place of phrase accents in intonational phonology. *Phonology*, 17, 143-18.
- Halliday, M.A.K. (1967). *Intonation and grammar in British English*. The Hague: Mouton.
- Jitcă D., Prosodic Phrases and Contrast Units in Intonation, *IEEE Proceedings of the 9th Conference on Speech Technology and Human-Computer-Dialogue (SpeD 2017)*, Bucharest, Romania, July 9-11, 2017.
- Jitcă D., Information Structure in Romanian Utterances with contrast relation, *IEEE Proceedings of the 8th Conference on Speech Technology and Human-Computer-Dialogue (SpeD 2015)*, Bucharest, Romania, Oct 14-17, 2015.
- Jitcă,D., Apopei, V., Păduraru, O., Marușcă, S. (2015). Transcription of Romanian Intonation. In Frota, S. & Prieto, P. (Eds), *Intonation in Romance*. Oxford: Oxford University Press, 284-316.
- Ladd, D.R. (2008). *Intonational Phonology*. Cambridge: Cambridge University Press.
- Pană-Dindelegan, G. (2005). ‘Predicatul’. In V. Guțu Romalo (coord.), *Gramatica limbii române II: Enunțul*, București: Editura Academiei Române, 241–266.
- Krifka, M., Musan, R. (2012). Information Structure: Overview and Linguistic Issues. In Krifka, M. & Musan, R. (Eds.), *The expression of information structure* in The Expression of Cognitive Categories (ECC) 5, Berlin: Mouton de Gruyter, 1-44.
- Steedman, M. (1991). Structure and intonation, *Language*, 68, 260–296.
- Steedman, M. (2000). Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 34, 649–689.
- Steedman, M. (2014). The Surface-Compositional Semantics of the English Intonation. *Language*, 90, 2–57.
- Xu, Yi & Chung X. (2005). Phonetic realization of focus in English declarative intonation, *JPhon* 33, 159-97.

NONLINEAR SYSTEM FOR ROMANIAN VOICE PROCESSING

CARMEN GRIGORAŞ^{1,2}, VICTOR GRIGORAŞ³, VASILE APOPEI²

¹*”Gr.T. Popa” University of Medicine and Pharmacy of Iaşi, 700115, Romania*

²*Institute of Computer Science of the Romanian Academy, Iaşi, 700481, Romania*

³*”Gheorghe Asachi” Technical University of Iaşi, 700506, Romania*

Corresponding author: Victor Grigoras, E-mail: grigoras@etti.tuiasi.ro

Abstract

Nonlinear systems with complex behaviour are deeply studied, in order to use their rich properties for signal generation, transmission and other processing. Chaotic systems are especially of great interest due to their corresponding natural implementations and complex processing capabilities. Chaos has been associated to turbulence, and from this point of view to inconsistencies in voice disorders. Nonlinear dynamics and mostly chaos theory has been used to assess pathological or emotional phonation. In the present contribution, the authors make comparison among several analogue nonlinear systems, for understanding their possibilities of extracting information from speech signals. Due to limited speech generation modelling, only control synchronization is approached, other methods, such as emitter partitioning and inverse system receiver, being less efficient. The obtained results suggest that complex *signum* nonlinearity systems and higher order ones can lead to more efficient processing structures, compared to classical, product nonlinearity ones.

Key words — chaotic dynamics, nonlinear synchronization, nonlinear systems, voice signal processing.

1. Introduction

Speech analysis is important for a large variety of applications, related with many research domains, such as medicine, linguistics, forensics or engineering, being one of the main forms of communication between people. Nonlinear dynamics has been used to study and understand the complexity of the normal or pathological speech, of different languages prosodies. Nonlinear methods seem to be suitable to explore the dynamics of the systems that produce voice signals, based on the observed nonlinear phenomenon which is turbulence during speech production (Narayanan and Alwan, 1995), (Johnson, 2011). Several nonlinear parameters were investigated for their performance in characterizing the mechanics that produces special phonemes, or discriminating among pathological and normal speech. Among these, we mention maximum Lyapunov exponent (LE), correlation dimension (CD), Kolmogorov entropy (KE) (Pitsikalis and Maragos, 2002), (Orozco-Arroyave *et al.*, 2013), (Henríquez *et al.*, 2009) and other more robust entropy measures, such as relative entropy (RE) (Scalassara *et al.*, 2009), permutation entropy (PE), or state space correlation entropy (SSCE) measures (Cao *et al.*, 2004), (Tripathy *et al.*, 2017). CD and KE measures, being sensitive to non-stationarity and noise contamination, produce ambiguous results mainly due to the presence of silent blocks in trained speech of stuttering subjects, whilst maximum LE performs slightly better (Kokkinos and Maragos, 2005). PE shows more efficiency in indicating the improvement in speech performance after the therapy, due to its robustness of

this measure to noise contamination and non-stationarity, although it does not consider the amplitude variations in a time series. RE and SSCE are better indicators of the increased dynamical complexity of pathological voice signals (Scalassara *et al.*, 2009), (Tripathy *et al.*, 2017).

Other nonlinear methods investigate speech nonlinearities by means of developing adequate nonlinear system models in order to detect such phenomena and extract related information and nonlinear parameters (Grigoraş *et al.*, 1997), (Teodorescu, 2006), (Teodorescu *et al.*, 2006). Models of jitter and shimmer may be used to investigate the presence of pathologies related to the voice (Cataldo and Soize, 2017). Other nonlinear methods related to this subject include glottal closure instant (GCI) detection, with application in prosody modification analysis, or pathological voice classification, among others (Silva *et al.*, 2009), (Daoudi and Kumar, 2015). This method is based on the precise computation of local geometrical and statistical parameters called the “singularity exponents” at every point in complex signal domain.

Our approach is based on the type of nonlinear processing specific to chaos synchronization. From the three basic methods, emitter partitioning, feedback control and inverse system, we have concentrated only on the second one. The first and third synchronization methods need in depth knowledge of the nonlinear state equations for the speaker emitter, including precise estimation of the system parameters. Thus, trying to apply these two methods might lead to errors especially to process signals from one speaker to another. The feedback control method is more flexible and it can adapt even to imprecise parameter knowledge of the system generating the analysed signal. Upon several system simulations we could also make this method fast enough. The signal analysis we performed is based on comparing three basic nonlinear systems: the Lorenz system, a three multiplier nonlinear system and a comparator nonlinearity system we introduced in a previous work (Grigoraş and Grigoraş, 2015). Making several simulations we concentrated on the error signal of the control loop to conclude on the signal parameters. The obtained results suggest that complex comparator systems of the third order can lead to more efficient processing structures, compared to classical multiplier systems with two and three nonlinearities.

2. The nonlinear system

The presented signal analysis is based on nonlinear signal processing. Basically, the chaos synchronization approach tries to highlight Romanian voice parameters. The main nonlinear synchronization methods are: receiver design based on emitter partitioning, nonlinear system feedback control and inverse system approach. All exposed methods are based on emitter system detailed modelling. The control method is more flexible and is able to adapt different emitter-receiver pairs and that is why this method is used in the following analysis.

2.1. The proposed general settings

Nonlinear synchronization was first proposed for communication applications. That is why the system generating the analysed signal is named emitter and the analyser, receiver, as suggested in Figure 1.

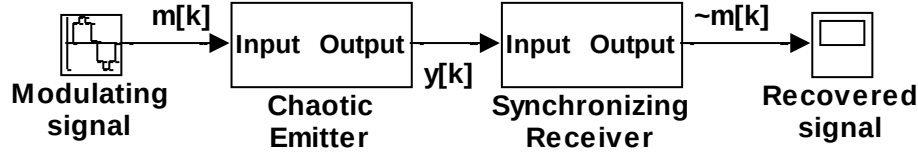


Figure 1. Nonlinear synchronization emitter-receiver pair

The control synchronization method is based on nonlinear feedback to obtain a synchronizing receiver to the chaotic emitter. In order to easily achieve synchronization, usually a nonlinear feedback loop is designed, aiming at nonlinear compensation of the direct path system, which is similar to the emitter. In our case, the voice signal is generated by an uncompleted model system, which is also time variant. As such, we chose to preserve the nonlinearity of the direct path system and use a simpler, linear feedback block, characterised by the feedback matrix of coefficients, as depicted in Figure 2 under the name ‘Matrix Feedback’.

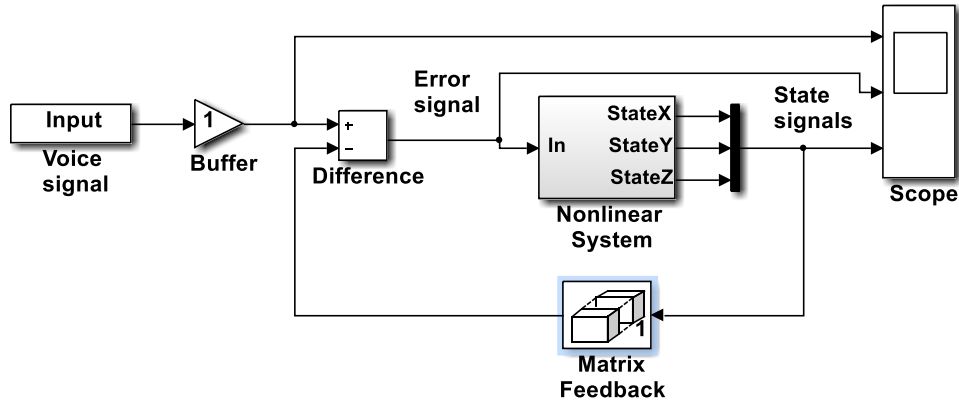


Figure 2. Block diagram of the feedback control system

In some complex dynamic behaviour of the nonlinear system, the feedback matrix is sometime even uni-dimensional, thus simplifying the analysis estimation.

In the following we intend to analyze the behaviour of several chaotic nonlinear systems by interpreting the ‘Error signal’ shape to detect some ‘Voice signal’ properties.

2.2. The Lorenz system

A classical approach is based on the Lorenz system (Lorenz, 1963), described by the third order state equations system:

$$\begin{cases} x' = \sigma(y - x) \\ y' = \rho \cdot x - y - x \cdot z \\ z' = x - \beta \cdot z + x \cdot y \end{cases} \quad (1)$$

The state equations of the Lorenz system (1), show that some state transition function is based on linear terms and algebraic nonlinear functions that are the two multiplications, in equations

two and three. The system parameters used in the following are the ones proposed by Lorenz (2), to obtain chaotic behaviour:

$$\sigma = 10; \quad \rho = 28; \quad \beta = 2.66; \quad (2)$$

As a result, the time evolution of the state vector is the one depicted in Figure 3.

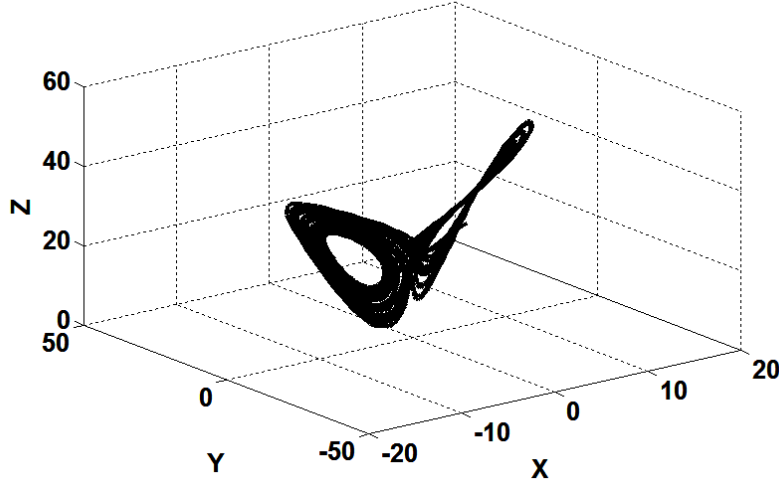


Figure 3. The two wing butterfly Lorenz system attractor

2.3. The three multiplier system

In order to increase the complexity of the nonlinear dynamics obtained, the number of analogue multiplications can be increased to three, as analyzed in (Grigorş and Grigoraş, 2014). The used system is also an order three one, but with more coefficients to increase adaptive flexibility:

$$\begin{cases} x_1' = -x_1 - \alpha \cdot x_2 \cdot x_3 \\ x_2' = -x_2 + a \cdot x_3 - \beta \cdot x_1 \cdot x_3 \\ x_3' = -b \cdot x_1 - a \cdot x_2 + x_3 + \gamma \cdot x_1 \cdot x_2 \end{cases} \quad (3)$$

The equilibrium points of the nonlinear system result in the form (4):

$$\begin{cases} x_1 = (\alpha\beta \cdot \rho^2 - ab\alpha \cdot \rho + a^2 - 1) / (a \cdot \gamma) \\ x_2 = a \cdot \rho - \rho\beta (\alpha\beta \cdot \rho^2 - ab\alpha \cdot \rho + a^2 - 1) / (a \cdot \gamma) \\ x_3 = \rho \end{cases} \quad (4)$$

where ρ is a root of the polynomial:

$$P(z) = z^4 - \frac{a \cdot b}{\beta} \cdot z^3 + \frac{a^2(\beta - \gamma) - 2\beta}{\alpha\beta^2} \cdot z^2 + \frac{a \cdot b}{\alpha\beta^2} \cdot z - \frac{(a^2 - 1)}{\alpha^2\beta^2} \quad (5)$$

The fact that the proposed nonlinear system is dissipative can be easily verified from the form of the state equations, as resulting from:

$$\nabla f = \frac{\partial f_1}{\partial x_1} + \frac{\partial f_2}{\partial x_2} + \frac{\partial f_3}{\partial x_3} = -1 \quad (6)$$

Moreover, this property does not depend on the system parameters, as defined in the system state equations. On the contrary, the ergodicity and sensitivity to initial conditions are dependent on these coefficients and, as such, they were justified by numerical simulations.

An analyzed set of equation coefficients, leading to chaotic dynamics, are (7):

$$a = 5; \quad b = 9; \quad \alpha = 50; \quad \beta = 20; \quad \gamma = 4.1; \quad (7)$$

The system state portrait is highly different from the previous one, suggesting that it evolves from two unstable limit cycles, as depicted in Figure 4.

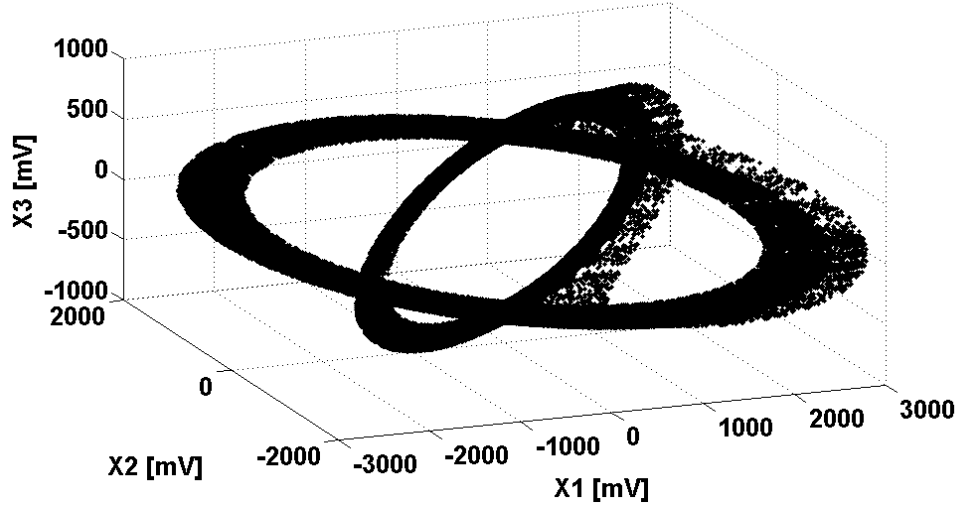


Figure 4. Three multiplier system state vector attractor

This suggests that the three multiplier system may synchronize onto signals generated by a two frequency torus nonlinear behaviour, not necessarily the most exact model for voice signals.

2.4. The signum nonlinearity system

For reducing implementation complexity, we can use a nonlinear system architecture based on a previously reported system, (Grigoraş and Grigoraş, 2015), which was described by the state equations:

$$\begin{cases} dx/dt = a \cdot y - z \\ dy/dt = -a \cdot x + z \\ dz/dt = -x - y - a \cdot z + \text{sign}(x) \end{cases} \quad (8)$$

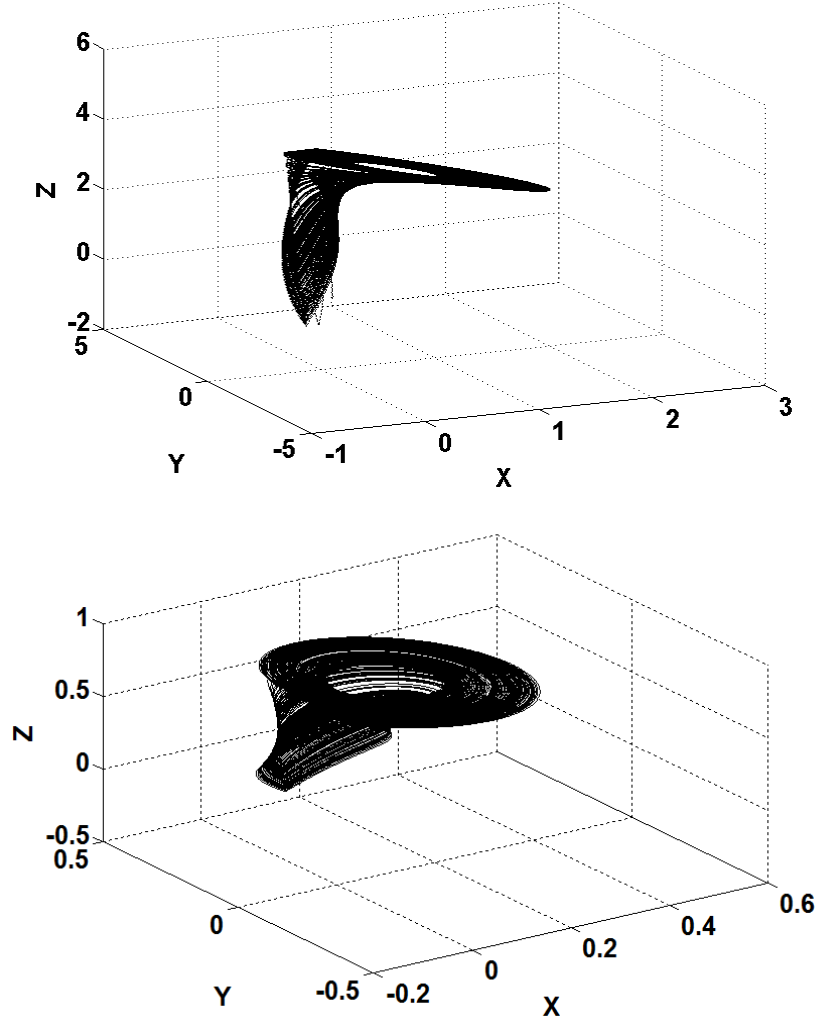


Figure 5. The time evolution of the *signum* nonlinearity system state vector for different coefficients values

As seen in equation (8), this third order system is based on a simple linear prototype, with few nonzero terms, characterized by the state transition matrix, A :

$$A = \begin{bmatrix} 0 & a & -1 \\ -a & 0 & 1 \\ -1 & -1 & -a \end{bmatrix} \quad (9)$$

The only nonlinear algebraic function is in the third state equation, comparator type nonlinearity, $\text{sign}(x)$,

$$u = \text{sign}(x) = \begin{cases} 1 & x > 0 \\ 0 & x = 0 \\ -1 & x < 0 \end{cases} \quad (10)$$

The approached system is characterised by chaotic evolution, with several strange attractors, depending on the parameter ' a '. As examples, state vector time evolution depicted in Figure 5, show the attractor for $a = 1.23$ (left graph) and $a = 2.21$ (right one).

This rich set of strange attractors, with different state-space shapes, suggests the possibility of this nonlinear system to highlight different voice signal properties.

3. Romanian voice data

We based our simulations on two recordings from Radio România Cultural, freely available on the following Youtube online address https://www.youtube.com/watch?v=YXs1Pz_beH0. These are ones of the first audio recordings of Romanian personalities, Henri Coanda and Nicolae Titulescu, who lived during the period of The Great Reunion. They were previously analysed with PRAAT, and with some algorithms developed for prosodic phrasing and diatopic intonational variation, as described in (Apopei, Păduraru, 2015) and (Jitcă *et al.*, 2015).

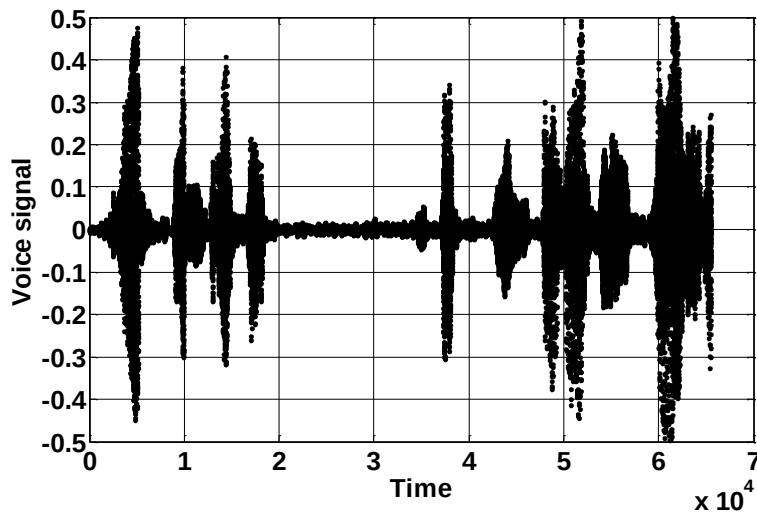


Figure 6. Example of Romanian voice signal

Examples of processed voice signal and its frequency spectrum are presented in Figure 6, for the time domain representation and Figure 7 for the frequency domain.

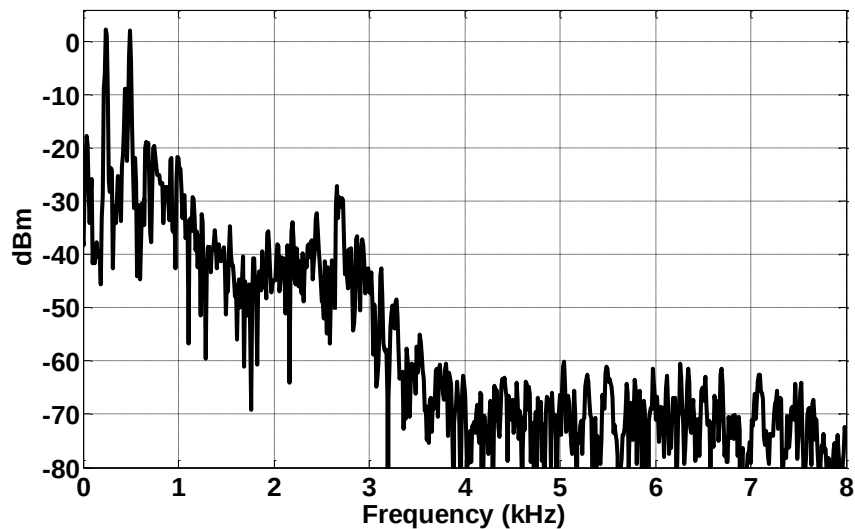


Figure 7. Example of voice frequency spectrum

4. Nonlinear processing results

The following simulation results try to compare the capability of the previously presented chaotic systems to help show some Romanian voice signals properties.

If the Lorenz system is taken as the basis nonlinear direct path block, the results may be taken as interpretative, as depicted in Figure 8.

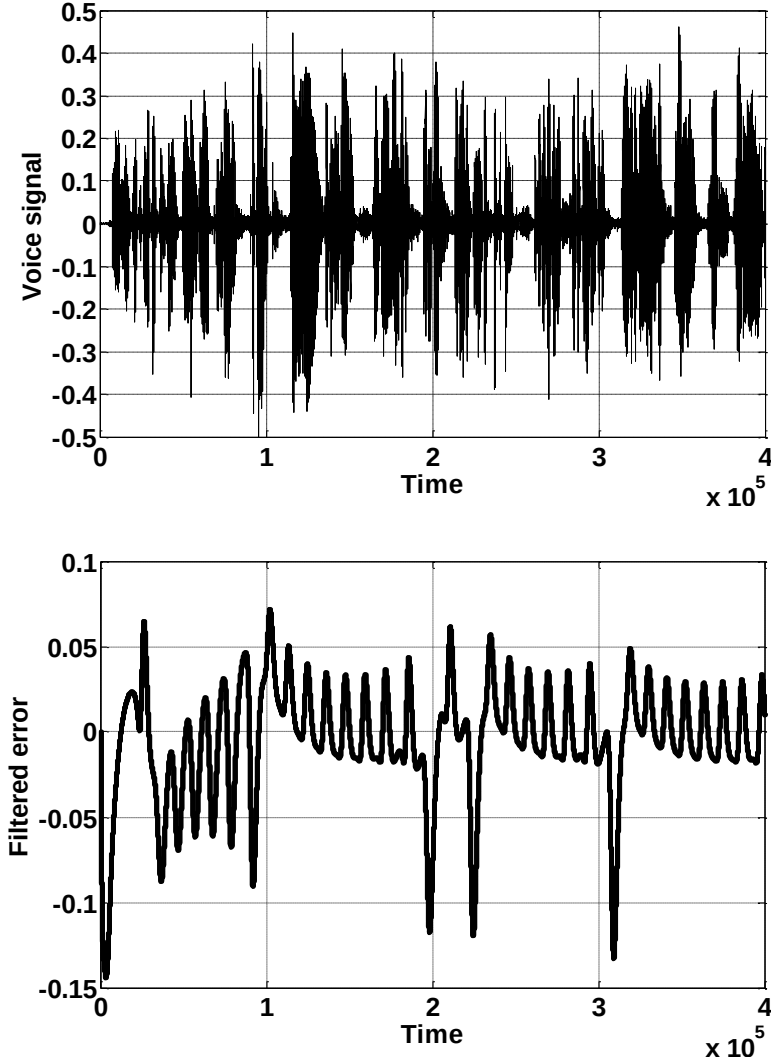


Figure 8. Voice signal (up) and filtered error signal (down)

The feedback matrix, denoted by K , used to build the feedback control system, as introduced in Figure 2, is:

$$K = [0.5 \quad -0.3 \quad -0.1]^T \quad (11)$$

For this simulation the error signal is filtered with the analogue transfer function:

$$H(s) = \frac{s + 0.1}{s + 10} \quad (12)$$

As seen from the filtered error signal, the obtained signal peaks are correlated to the voice information time periods.

We also tried the three multiplier nonlinear system and made several feedback matrices and filters alternatives, but all showed noisy error signals, not useful for automated decisions. For instance, using the feedback matrix:

$$K = [0.0001 \quad -0.0001 \quad 0.0003]^T \quad (13)$$

and:

$$H(s) = \frac{s + 0.23}{s + 100} \quad (14)$$

For the same voice signal, the filtered error signal is depicted in Figure 9, highlighting the undesired noisy shape. Another structure used in our comparison is based on the signum nonlinear system, leading to the results in Figure 10. The used voice signal is the same as depicted in figure 8, but the feedback matrix is modified to:

$$K = [0.1 \quad 0.3 \quad 0.3]^T \quad (15)$$

and the error signal is filterer using a different transfer function filter:

$$H(s) = \frac{s + 1}{s + 35} \quad (16)$$

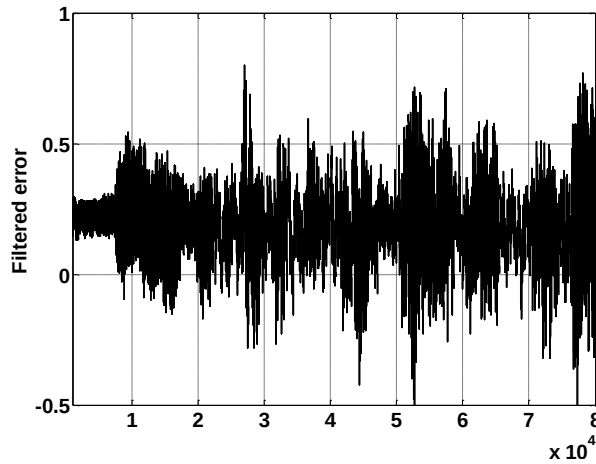


Figure 9. Three multiplier system filtered error signal

Expectedly, the filtered error signal highlights in a more observable way the speech and pause time periods, but by the negative valued spikes.

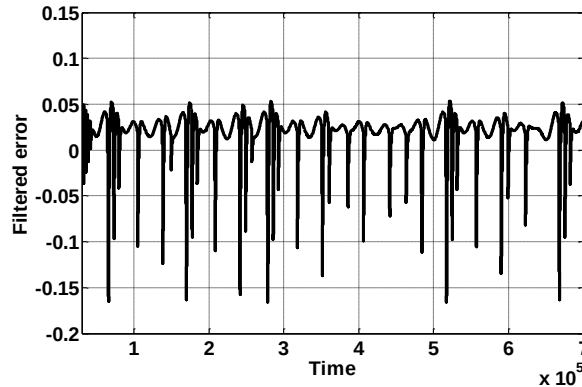


Figure 10. Filtered error signal for the signum nonlinear system

5. Conclusions

In this paper we applied chaos synchronization to speech analysis. We have chosen to implement the feedback control method, because of its flexibility and because it can adapt even to imprecise parameter knowledge of the system generating the analyzed signal. We performed several simulations in order to improve the speed of the computation. The signal analysis is based on comparing three basic nonlinear systems, and our results highlight the fact that the comparator nonlinearity system gives the best results regarding the pointing of the voice signal characteristics. Making several simulations we focused on the error signal of the control loop to conclude on the signal parameters. The obtained results suggest that complex comparator systems of the third order can lead to more efficient processing structures, compared to classical multiplier systems with two and three nonlinearities.

A precision analysis regarding the correlation between voice pauses and nonlinearly generated pulses by the dynamic systems excited by the audio signal was not performed. Such a detailed result needs deeper research taking into account more variant voice signals, larger and more complex analyzed nonlinear systems and precise measurement methods. More research is also needed to improve our method and to obtain results on other nonlinear systems. Further aspects of interest that could be approached are connected to the areas of voice activity detection and nonlinear voice processing under perturbation and measurement imperfection.

Acknowledgements

We are grateful to Professor H.N. Teodorescu, member of The Romanian Academy, for his suggestions and support.

References

- Narayanan, S., Alwan, A.(1995). A nonlinear dynamical systems analysis of fricative consonants, *J. Acoust. Soc. Am.*, 97:4, 2511-2524.
- Johnson, K.(2011). Chapter 4: Basic Audition. In *Acoustic and auditory phonetics*, Wiley-Blackwell Publishers, 3rd Ed., 82-98.

- Pitsikalis, V., Maragos, P.(2002). Speech analysis and feature extraction using chaotic Models. In *Proc. Int'l Conf. Acoustics Speech and Signal Processing (ICASSP-2002)*, Orlando, USA, 533-536.
- Orozco-Arroyave, J.R., Arias-Londono, J.D., Vargos-Bonilla, J.F., Nöth E.(2013). Analysis of speech from people with Parkinson's disease through nonlinear dynamics. In *Advances in Nonlinear Speech Processing: 6th International Conference Nolis2013 Proc.*, Springer, Eds. T.Drugman, T. Dutoit, 112-119.
- Henríquez, P, Alonso, J.B., Ferrer, M.A., Travieso, C.M., Godino-Llorente, J.I., Díaz-de-María, F.(2009). Characterization of Healthy and Pathological Voice Through Measures Based on Nonlinear Dynamics, *IEEE Trans.Audio, Speech, and Lang. Proc.*, 17:6, 1186–1195.
- Scalassara, PR, Dajer, ME, Maciel, CD, Guido, RC, Pereira, JC (2009). Relative entropy measures applied to healthy and pathological voice characterization. *Applied Mathematics and Computation*, 207:1, 95-108.
- Cao, Y., Tung, W., Gao, J.B., Protopopescu V.A., Hively L.M. (2004). Detecting dynamical changes in time series using the permutation entropy. *Physical Review E*, 70: 046217.
- Tripathy, R.K., Deb, S., Dandapat, S.(2017). Analysis of physiological signals using state-space correlation entropy. *Healthcare Technol. Lett.*, 4:1, 30-33.
- Grigoraş, F., Teodorescu, H.N., Apopei, V. (1997). Analysis of nonlinear and nonstationary processes in speech production. *IEEE ASSP Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York.
- Teodorescu, H.N. (2006). Occlusal sound analysis revisited. *Advances in Medical, Signal and Information Processing. MEDSIP 2006. IET 3rd Int. Conf.*, Glasgow, UK.
- Teodorescu, H.N., Feraru, M., Trandabăţ, D.(2006). Nonlinear Assessment of the Professional Voice 'Pleasantness'. *Proceedings of the 18th Biennial International EURASIP Conference BIOSIGNAL 2006*, J.Jan, J.Kozumplik, I. Provaynik (Eds.), Vutium Press, 63-66, Biosignal 2006, Brno, Czech Republik.
- Kokkinos, I., Maragos, P.(2005). Nonlinear Speech Analysis Using Models for Chaotic Systems. *IEEE Trans. Audio, Speech, and Lang. Proc.*, 13:6, 1098-1109.
- Cataldo, E., Soize, C. (2017). Voice signals produced with jitter through a stochastic onemass mechanical model. *Journal of Voice*, Elsevier, 31:1, 111.e9-111.e18.
- Silva, D.G., Oliveira, L.C., Andrea, M (2009). Jitter estimation algorithms for detection of pathological voices. *Eurasip Journal on Advances in Signal Processing—Special issue on analysis and signal processing of oesophageal and pathological voices*, Article No. 9.
- Daoudi, K., Kumar, A.J. (2015). Pitch-based speech perturbation measures using a novel GCI detection algorithm: Application to pathological voice classification. In *INTERSPEECH'2015 - 16th Annual Conference of International Speech Communication Association*, Dresden, Germany.
- Lorenz, E. N. (1963). Deterministic nonperiodic flow. *Journal of the Atmospheric Science*, 20:3, 130-141.
- Grigoras, V., Grigoras, C. (2014). A Novel Chaotic System for Random Pulse Generation. *Advances in Electrical and Computer Engineering*, 14:2, 109 – 112.

- Grigoras, V., Grigoras, C. (2015). Dynamic and Statistic Analysis of a Simple Chaotic Generator. In *12th International Symposium on Signals, Circuits and Systems (ISSCS2015)*, Iași, Romania, 77 - 80.
- Jitcă, D., Apopei, V., Păduraru, o., Marusca, S. (2015). Transcription of Romanian intonation. In *Sónia Frota and Pilar Prieto (eds.), Intonational variation in Romance*, 284-316. Oxford: Oxford University Press.
- Apopei, V., Păduraru, O. (2015). Towards prosodic phrasing of spontaneous and reading speech for Romanian corpora. In *2015 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Bucharest, Romania.

COMPARISON OF I-VECTOR AND GMM-UBM SPEAKER RECOGNITION ON A ROMANIAN LARGE SPEECH CORPUS

ALEXANDRU-LUCIAN GEORGESCU, HORIA CUCU, CORNELIU BURILEANU

*Speech and Dialogue Research Laboratory,
University Politehnica of Bucharest,
lucian.georgescu@speed.pub.ro,
{horia.cucu, corneliu.burileanu}@upb.ro*

Abstract

Anatomical features are more resistant in front of falsification or theft, which give them an important advantage in biometric systems, comparing to the classical ones, based on usernames and passwords. Thus, non-intrusive biometric systems have gained a major edge in recent years, speaker recognition being one of them. This study focuses on speaker verification and identification based on the UBM-ivector framework. The results are compared with a baseline GMM-UBM system trained and evaluated in the same manner. To the best of our knowledge, the speaker recognition experiments presented in this paper are performed on a Romanian-speech dataset that is more than 7 times larger than the ones used in any other papers. Two datasets comprising more than 200 hours of read and spontaneous speech and one set of connected-digits spoken by 154 speakers, with a total duration of 38 hours represents the audio data used in this paper. Both systems were evaluated in terms of Equal Error Rate (EER). For the speaker verification task, the obtained EER is around 1% or bellow, while for the speaker identification task, the Identification Error Rate (IER) is very close to 0%. The focus of the experiments is on the number of files used for speaker enrollment.

Key words — GMM-UBM framework, RoDigits, speaker verification, speaker identification, UBM-ivectors framework.

1. Introduction

The last few years have brought a deep interest in the context of data security, biometrics being a very active field, dealing with the statistical study and the measurement techniques applied to living organisms. Biometric systems have the sole property of relying on the anatomical characteristics of the individual, which are much harder to counterfeit or steal, unlike conventional password-based or access tokens systems.

One of the most fashionable biometric technologies is speaker recognition, which is one of the main branches in digital signal processing, along with speech recognition. Human communication is mostly based on speech and the voice contains a multitude of speaker-specific features. These features can be extracted quite simply from the vocal signal, avoiding direct interaction with the speaker, thus being a non-invasive method. The shape and size of the vocal tract are some of the physiological characteristics that differentiate speakers. The behavior of a person could be also a factor that characterizes a person's speech. The manner of speaking, depending of the accent, rhythm, intonation, pronunciation and more (Kinnunen and Haizhou, 2010) is specific for each individual. Speaker's intentions could classify them into cooperative speakers, wanting their identity to be recognized or non-cooperative speakers. The first situation is found in some applications, such as access control, transaction

authentication (telephone banking, e-commerce) or device customization (retrieve personal settings based on speaker identity) (Reynolds, 2001). In the second situation, the speaker does not want his identity to be determined, this being a common case in forensics or law enforcement.

Speaker recognition usually implies two different tasks: speaker verification and speaker identification. Speaker verification consists in verifying if the claimed identity of a speaker matches his real identity, being a binary decision. Speaker identification involves determining the identity of the speaker, without having previous knowledge about his possible identity. Both tasks could exist in two scenarios, depending on how speakers relate to the system: close-set scenario and open-set scenario. A closed-set system always presumes that the tested speaker is one of the enrolled speakers. An open set system could receive as input a voice from an enrolled or not enrolled speaker.

Depending on the content of the utterance, automatic speaker recognition systems can also be categorized into text-dependent or text-independent systems (Kaur and Jain, 2015). Text-dependent supposes that the speaker has to pronounce a predefined word or sentence. This is especially common in access control systems where, in order to be verified, the speaker has to utter a password. The disadvantage of this kind of system is given by the possibility of fooling the system using an audio recording with the legitimate speaker while he utters the password. Text-independent has no constraints on what speaker needs to pronounce. These systems are very useful in forensics. However, from the point of view of accuracy, text-dependent systems are more efficient.

This paper is structured such as following: Section 2 presents some related work and specific algorithms applied in speaker recognition field. Section 3 introduces the speaker recognition framework used in our experiments. Section 4 describes the experimental setup, while the Section 5 presents the results. Conclusions are drawn in Section 6.

2. Related work

The National Institute of Standards (NIST) organizes since 1996 special sessions in order to evaluate speaker recognition techniques. The enhancement of technological advance and encouraging the emergence of new methods was their main goals (Martin and Przybocki, 2001). NIST has dealt with the standardization of speaker recognition concepts and performance assessment metrics (Doddington *et al.*, 2000). Currently, NIST provides standard speech databases containing training and evaluation sets. In this way, direct comparisons can be made between different algorithms.

For a long time, the GMM-UBM framework (Reynolds *et al.*, 2000) (Gaussian Mixture Model – Universal Background Model) has been state-of-the-art in the speaker recognition systems. The last decade has brought popularity increase of neural networks. However, errors have persisted because of channel variability. Feature mapping (Reynolds, 2003) is one of the techniques used to solve this problem. Features originating from different channels are mapped together into a common channel, independent of the feature space. Joint Factor Analysis (JFA) (Kenny *et al.*, 2007) and identity vectors (i-vectors) (Dehak *et al.*, 2011) represents other successful methods for the channel compensation approach.

Neural networks (NN) and deep neural networks (DNN) have had a major impact in recent years. A speaker verification system using DNN, where the input layer processes raw waveform is presented in (Variani *et al.*, 2014). This system outperforms an i-vector system by 14%-15% (depending on speech type, clean or noisy). A DNN used for bottleneck features

extraction is used in [11]. The gain in terms of equal error rate over MFCCs is around 60%, evaluated on NIST SRE 2010 database.

Several speaker recognition attempts for Romanian language were the subject of study for a couple of research groups. Most of them were based on machine learning algorithms, such as GMM-UBM or Support Vector Machines (SVN), but no one used DNNs. Our approach presented in (Georgescu and Cucu, 2018) was the second one that uses GMM-UBM for speaker recognition on Romanian speech. An important factor in our research is represented by the speech corpus, which is more than seven times larger than the largest Romanian speech corpus used for this task in the past.

A hybrid classification method, which involves a Hidden Markov Model (HMM) to model each vowel from each speaker and a SVM to classify them, was proposed in a recent study on Romanian speaker verification (Răstoceanu and Militaru, 2010). The experiments were carried out on a corpus containing 10 speakers with 50 utterances from each one.

Two automatic speech recognizers, one speaker-dependent used for the client model and another speaker-independent used for the world model, are the components of a speaker verification system presented in (Munteanu and Toma, 2010). The decision is made by comparing the acoustic scores output during Viterbi forced-alignment. For those experiments, a corpus of 11 speakers, containing around 370 utterances for each one was used.

In (Răstoceanu and Lazăr, 2011) the experiments were carried out on a corpus comprising 120 phrases per each of the 22 speakers using support vector machines applied on a fusion of speech features.

The study presented in (Segărceanu and Zaharia, 2013) uses a corpus with 26 speakers, each one pronouncing between 24-66 phrases. The authors employ a DTW-based similarity method applied on the speaker feature sequences. In a subsequent study (Segărceanu *et al.*, 2014), the same authors use a GMM-UBM system for the same task.

A voice-based authentication system is described in (Sandulescu and Halcu, 2015). The authors used a corpus of 10 speakers, 20 words each, together with 5 minutes of speech. Classification was performed using a SVM trained on LPC and MFCC features.

3. The UBM-ivectors Framework for Speaker Recognition

The UBM-ivectors framework is a common approach in the speaker recognition field. The principle behind this kind of system is to represent the characteristics of each speaker through a fixed dimension vector, called i-vector. Equation (1) expresses the i-vector framework:

$$S = m + Tw \quad (1)$$

where S is the speaker and channel dependent GMM supervector, m is the UBM mean supervector and T is the total variability matrix, referred also as the i-vector extractor, consisting in speaker variability and channel variability and w is the i-vector, a low-dimensional variability factor which controls an eigen-dimension of T .

But, to obtain such a vector, same previous steps are required. First of all, the training stage implies the creation of the Universal Background Model (UBM), which models general speech, without being specific to a certain speaker. The UBM is a statistical model, composed by a weighted sum of Gaussian Densities and characterized by mean and variance. The UBM is trained using the Expectation-Maximization (EM) algorithm over a large set of acoustic data. The main goal of the UBM is to estimate probability densities that cover the variety of speech. The training stage also involves the training of the i-vector extractor.

Enrollment represents the next step in this pipeline. The i-vectors are extracted from each audio file of the speakers considered known by the system. An averaging operation is performed, obtaining a single i-vector per speaker.

The test stage consists in determining the identity of a speaker, with or without prior information about his possible identity, by computing the similarity between his i-vector and the i-vectors of the enrolled speakers. Several techniques are used to obtain this score, such as simple Euclidean distance, cosine distance and some more advanced as Linear Discriminant Analysis (LDA) or Probabilistic Linear Discriminant Analysis (PLDA). The last method is used as a pattern recognition algorithm and it can be considered a particular case of Joint Factor Analysis (JFA) with a single Gaussian component. PLDA estimate the likelihood for the two i-vectors were or were not generated by the same speaker.

4. Experimental Setup

The experiments in this paper were performed using RoDigits (Georgescu *et al.*, 2018), a connected-digits corpus in the Romanian language, available online¹. This corpus consists of 154 speakers, each one pronouncing 100 utterances. Each utterance contains 12 random digits (0 to 9) in Romanian language. The corpus was recorded using an online application by students of Faculty of Electronics, Telecommunications and Information Technology from University “Politehnica” of Bucharest. The total size of the corpus is about 38 hours.

The corpus was split into an enrollment set and two test sets (one for the open-set scenario and one for the closed-set scenario). The enrollment set comprises 11,120 utterances: 80 utterances from each of the 139 enrollment speakers. The close test set consists of 2,780 utterances: 20 utterances from the same 139 enrollment speakers. The open test set consists of 1489 utterances: around 100 utterances from each of the other 15 speakers.

Some other Romanian corpora were used for these experiments. The first one, RSC-train, is composed by 145k read speech utterances from 157 different speakers, summing a total of 100 hours of speech. The second and the third ones contain read speech from news broadcasts and also spontaneous speech, extracted from radio and TV shows, sometimes affected by background noises. They comprise together around 224k utterances (about 130 hours of speech). The details for all speech corpora are presented in Table 1.

The acoustic parameters extracted from all the audio data are the MFCC features, consisting of 20 dimensional vectors, computed on 25 ms length frames. A voice activity detector (VAD) is applied to keep only the speech frames and ignoring the non-speech segments.

For a faster training, the third part of RSC-train, SSC-train1 and SSC-train2 sets was used to train a diagonal covariance matrix UBM with 2048 Gaussian Densities. Then, using this UBM and the other two third parts from the same sets, a full covariance matrix UBM of 2048 components was trained.

Table 1: Speech Corpora

Usage	Set	Duration
Training (UBM, i-vector extractor, PLDA)	RSC-train	94 h, 46 m
	SSC-train1	27 h, 27 m
	SSC-train2	103 h, 17 m
Enrollment	RoDigits train-set	27 h, 02 m

¹ RoDigits corpus v1.0: <http://speed.pub.ro/downloads>

COMPARISON OF I-VECTOR AND GMM-UBM SPEAKER RECOGNITION ON A ROMANIAN LARGE SPEECH CORPUS

Closed set testing	RoDigits dev-set	6 h, 47 m
Open set testing	RoDigits eval-set	3 h, 43 m

The i-vector extractor of dimension 600 was trained after 5 Expectation-Maximization iterations using as input the three training sets and the full UBM.

The enrollment step implies extracting i-vectors from sets of 1, 2, 5, 10 up to 80 audio files, for all the enrolled speakers. The test step consists in i-vectors extraction for the audio files in the test corpus. Then, a PLDA scoring technique is applied to compute the similarity between i-vectors of tested speakers and i-vectors of enrolled speakers. The systems were evaluated in terms of equal error rate (EER), finding a threshold for which the false rejection and false acceptance rates are equal.

In this work we used as a baseline the GMM-UBM system presented in (Georgescu and Cucu, 2018). The experiments in (Georgescu and Cucu, 2018) were performed on a smaller version of the RoDigits corpus (v0.9).

Using the enrollment set and the closed set testing set, summing about 10k utterances from 100 speakers, several UBM models were created, varying the number of Gaussian densities (GD) as powers of 2: the simplest model comprises 4 GDs, while the most complex one comprises 256 GDs.

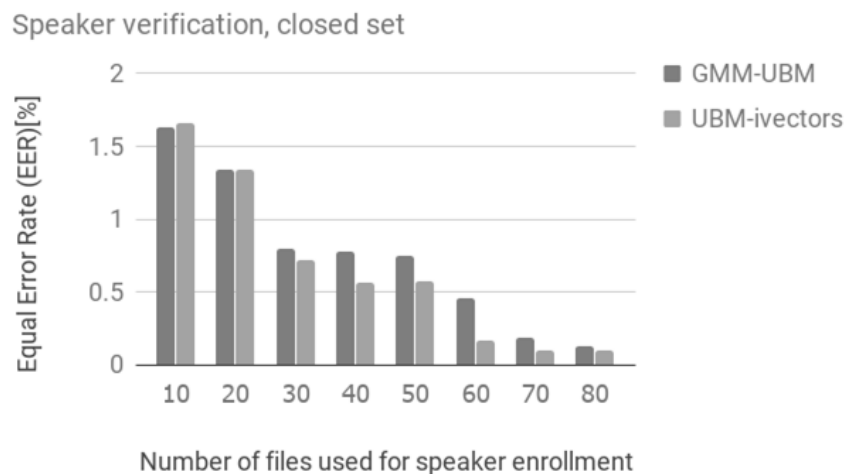
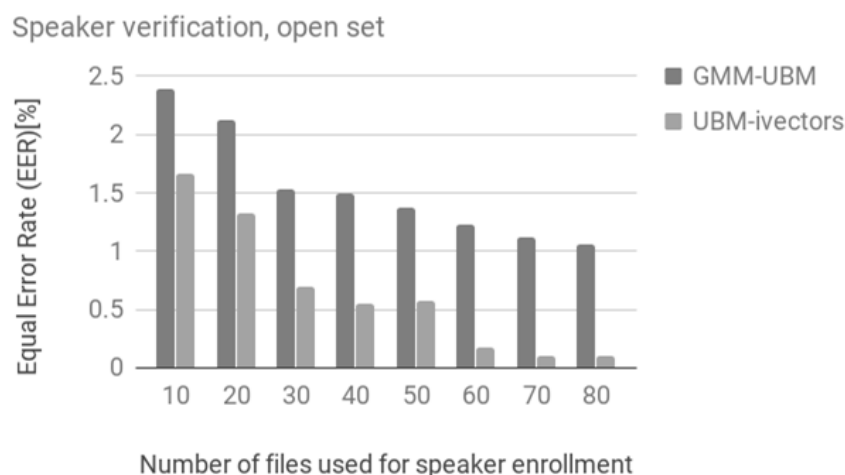
The enrollment step was similarly performed in both experiments. Between 10 and 80 utterances belonging to each speaker were used to adapt the UBM model, resulting individual GMM models.

5. Experimental Results

5.1. Speaker Verification

The first experiment we performed aimed to assess the performance of the speaker verification system in the closed set scenario. Figure 1 shows the variation of equal error rate (EER) as a function of the number of enrollment utterances, for both the evaluated systems: GMM-UBM and UBM-i-vectors. Although we would have expected to see much better results for the UBM-i-vectors system, the experiments show that the two systems are very similar. The UBM-i-vectors system is slightly better only when more than 40 utterances are used for enrollment.

The second experiment aimed to assess the performance of the speaker verification system in the open set scenario. Figure 2 shows the variation of equal error rate (EER) as a function of the number of enrollment utterances, for both the evaluated systems: GMM-UBM and UBM-i-vectors. The UBM-i-vectors system achieves similar EER values as in the closed-set scenario, clearly overpassing the GMM-UBM system. The experiment suggests that GMMs cannot cope with un-enrolled speakers as well as the i-vectors can. There are probably cases when un-enrolled speakers with voices similar to some of the enrolled speakers are able to fool the system.

**Figure 1:** Speaker verification in closed set scenario**Figure 2:** Speaker verification in open set scenario

5.2. Speaker Identification

Finally, our last experiment aimed to compare the error rates for speaker identification (closed-set) as opposed to speaker verification. Figure 3 presents the results. What is interesting here is that for a small amount of enrollment data, the i-vectors outperform the GMMs, while if more audio data is available for enrollment, the GMM-based system performs better.

6. Conclusions

This paper aimed to present speaker verification and speaker identification experiments using the UBM-ivectors framework, compared with a baseline system presented in one of our previous papers. A read speech corpus and a spontaneous speech corpus were used to train the UBM. Also, a spoken Romanian connected-digits corpus was used for speaker enrollment and for testing purposes. During the experiments, the number of the enrollment files was varied. As this number was higher, the performance was better.

COMPARISON OF I-VECTOR AND GMM-UBM SPEAKER RECOGNITION ON A ROMANIAN LARGE SPEECH CORPUS

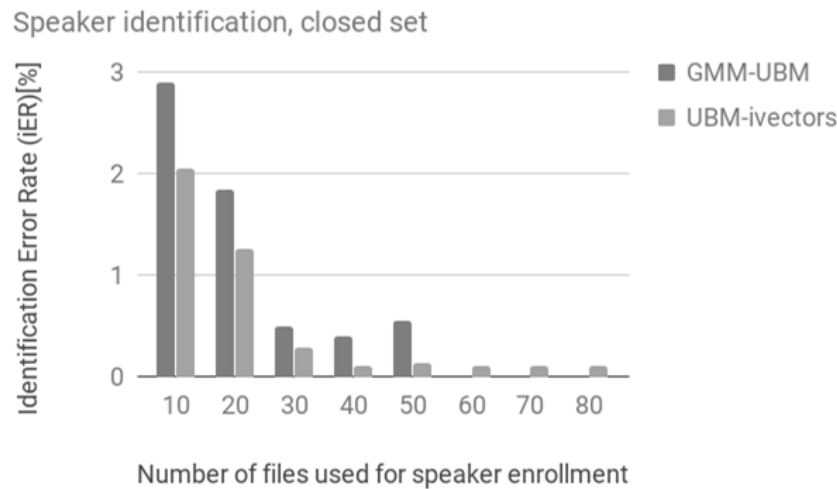


Figure 3: Speaker identification in closed set scenario

For the speaker verification task, the EER was computed for both closed and open set scenario. The best results were obtained for 80 enrollment files, the EER value being equal to 0.17%. Speaker identification was performed only in closed set scenario. For 80 enrollment files the identification error rate was equal to 0.11%.

Acknowledgement

This work was supported by the Ministry of Innovation and Research, UEFISCDI, project SPIA-VA, agreement 2SOL/2017, grant PN-III-P2-2.1-SOL-2016-02-0002.

References

- Kinnunen, T., Haizhou, L. (2010). An overview of text-independent speaker recognition: From features to supervectors. *Speech communication*, 52.1, 12-40.
- Reynolds, D. A. (2001). Automatic speaker recognition: Current approaches and future trends. *Speaker Verification: From Research to Reality*, 5, 14-15.
- Kaur, K., Jain, N. (2015). Feature Extraction and Classification for Automatic Speaker Recognition System-A Review. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5.
- Martin, A. F., Przybocki, M. A. (2001). The NIST speaker recognition evaluations: 1996-2001. In *Proceedings of the Workshop "A Speaker Odyssey-The Speaker Recognition Workshop"*, Crete, Greece, 39-43.
- Doddington, G. R., Przybocki, M. A., Martin, A. F., Reynolds, D. A. (2000). The NIST speaker recognition evaluation—overview, methodology, systems, results, perspective. *Speech Communication*, 31(2), 225-254.

- Reynolds, D. A., Quatieri, T. F., Dunn, R. B. (2000). Speaker verification using adapted Gaussian mixture models. *Digital signal processing*, 10(1-3), 19-41.
- Reynolds, D. A. (2003). Channel robust speaker verification via feature mapping. In *Proceedings of Acoustics, Speech, and Signal Processing (ICASSP'03)*. 2003 IEEE International Conference on, Vol. 2, II-53.
- Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P. (2007). Speaker and session variability in GMM-based speaker verification. *IEEE Transactions on Audio Speech and Language Processing*, 15(4), 1448.
- Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4), 788-798.
- Variani, E., Lei, X., McDermott, E., Moreno, I. L., Gonzalez-Dominguez, J. (2014). Deep neural networks for small footprint text-dependent speaker verification. In *Proceedings of Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 4052-4056.
- Matějka, P., Glembek, O., Novotný, O., Plchot, O., Grézl, F., Burget, L., Cernocký, J. H. (2016). Analysis of DNN approaches to speaker identification. In *Proceedings of Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference on*, 5100-5104.
- Georgescu, A. L., Cucu, H. (2018). GMM-UBM Modeling for Speaker Recognition on a Romanian Large Speech Corpora. In *Proceedings of 2018 International Conference on Communications (COMM)*, IEEE, 547-551.
- Răstoceanu, F., Militaru, D. (2010). A HMM/SVM hybrid method for speaker verification. In *Proceedings of 2010 International Conference on Communications (COMM)*, IEEE, 111-114.
- Munteanu, D. P., Toma, S. A. (2010). Automatic speaker verification experiments using HMM, In *Proceedings of 2010 International Conference on Communications Communications (COMM)*, IEEE, 107-110.
- Răstoceanu, F., Lazăr, M. (2011). Score fusion methods for text-independent speaker verification applications. In *Proceedings of Speech Technology and Human-Computer Dialogue (SpED)*, IEEE, 1-6.
- Segărceanu, S., Zaharia, T. (2013). Speaker Verification using the dynamic time warping, *UPB Sci. Bull., Series C*, Vol. 75, Iss. 1, 179-194.
- Segărceanu, S., Zaharia, T., Rădoi, A. (2014). Speaker verification using GMM modelling, *UPB Sci. Bull., Series C*, Vol. 76, Iss. 2, 117-130.
- Sandulescu, V., Halcu, I. (2015). Speaker Authentication for an Assistive Domotic System. In *Proceedings of Control Systems and Computer Science (CSCS)*, IEEE, 337-340.
- Georgescu, A.L., Caranica, A., Cucu, H., Burileanu, C. (2018). RoDigits – a Romanian Connected-Digits Speech Corpus for Automatic Speech and Speaker Recognition, *UPB Sci. Bull., Series C*, Vol. 80, Iss. 3, 45-62.

A COMPARISON BETWEEN TRADITIONAL MACHINE LEARNING APPROACHES AND DEEP NEURAL NETWORKS FOR TEXT PROCESSING IN ROMANIAN

ADRIANA STAN, MIRCEA GIURGIU

*Communications Department,
Technical University of Cluj-Napoca, Romania
{Adriana.Stan, Mircea.Giurgiu}@com.utcluj.ro*

Abstract

This paper presents a comparison between traditional machine learning approaches (decision trees and multilayer perceptron) and the latest trend in artificial intelligence, deep neural networks for three separate tasks of text processing in Romanian. The tasks we examine are: lexical stress assignment, syllabification and phonetic transcription. The evaluation is performed on large manually transcribed lexicons and uses simple input features derived strictly from the orthographic form of the words. Results show that, depending on the task, the performance of each of the algorithms can vary, and that in some limited cases, the decision trees can outperform the deep neural networks.

Key words — decision trees, deep neural networks, lexical stress assignment, multilayer perceptron, phonetic transcription, Romanian, text processing, syllabification.

1. Introduction

Text processing is an essential part for most of the speech processing applications. In text-to-speech synthesis systems, for example, it is essential to know the correct syllabification, or the stress assignment of each word in the text input, so that the acoustic models can generate speech that is as natural as possible. Or, in automatic speech recognition systems, the language model decoders need to take into account the phonetic transcription of the words, so that a correct phrase is generated as output. But all text processing steps require large lexicons as training datasets.

In this respect, Romanian is an under resourced language (Trandabăţ *et al.*, 2012) and does not benefit from readily available, high quality language resources and systems. However, in the recent years more and more resources and tools have been developed and published as open-source or for academic use, as described in (Barbu, 2008), (Toma *et al.*, 2017), (Domokos *et al.*, 2012), (Boroş *et al.*, 2012), (Stan *et al.*, 2017), (DEX online, 2018) or (Boroş *et al.*, 2018). With the availability of these resources, more and more studies and tools have been published, investigating either entire text processing systems, or only parts of them. The following paragraphs index some of the most important work with respect to the three text processing tasks we address in this paper: lexical stress assignment, syllabification and phonetic transcription. It is also worth mentioning a few papers which have introduced full systems, such as (Ungurean and Burileanu, 2011) which addresses the diacritic restoration, text normalisation, syllabification, phonetic transcription and lexical stress positioning; (Stan *et al.*, 2011) describes a complete text-to-speech synthesis system including the front-end text

processing and acoustic modelling. Boroş *et al.* (2012) and Boroş *et al.* (2018) present a full text processing tool and two newly developed Romanian speech corpora.

Lexical stress assignment was investigated by Oancea and Bădulescu (2002) by developing a set of rules which also use morphologic, lexical and phonetic information. Ciobanu *et al.* (2014) and Chitoran *et al.* (2014) use the consonant-vowel structure of the word and a cascaded model with averaged perceptron training to predict the syllable boundaries and lexical stress placement. Balc *et al.* (2015) performed a series of experiments based on Random Forest algorithm for stress assignment.

The topic of syllabification in Romanian has been studied by Toma *et al.* (2009) which present a rule-based approach for syllabification and phonetic transcription. Dinu *et al.* (2013) use sequence tagging, support vector machines and a set of rules. In (Balc *et al.*, 2015) the authors perform a comparison of Random Forest, SMO, Naive Bayes and Ada Boost algorithms for syllabification. Boroş *et al.* (2017) used modified decision trees for syllabification and phonetic transcription in order to improve the efficiency and speed of the two text processing steps as part of a complete system.

Phonetic transcription has seen more interest, as it is common to both the text-to-speech synthesis systems, as well as automatic speech recognition. Burileanu (2002) investigates the use of neural networks, a similar approach, based on ANNs, being discussed in (Jitcă *et al.*, 2003) and (Domokos *et al.*, 2011). On the other hand, Ordean *et al.* (2009) developed a large set of rules for the Romanian phonetic transcription and combined them with decision trees for exceptions. (Boroş *et al.*, 2012) introduces a maximum entropy classifier and a purposely built specialized algorithm for same processing task. Toma *et al.* (2013) compare 5 methods based on decision trees, neural networks, support vector machines, pronunciation by analogy and an expert system. A novel approach to the phonetic transcription task was presented in (Cucu *et al.*, 2014) where grapheme-to-phoneme conversion is performed using statistical machine translation principles.

However, none of the previous cited works included the recent and revolutionary technique of deep learning (DL) for text processing. Therefore, in this paper we perform a comparison between two of the most common traditional machine learning methods used in the previous works and DL.

Deep Learning has seen a large increase in its use across many research topics (Zhang *et al.*, 2018) and has revolutionised many of these fields, including computer vision, speech-enabled systems and information retrieval. Their main advantage over traditional machine learning algorithms is the fact that, due to their multiple hidden layers and training strategies, can learn non-linear high-dimensional partitions of the data spaces. But this comes at a cost, as multiple hidden layers require large amounts of training data. If this data is available, the accuracy of the networks is very high and their results are consistent across datasets.

Therefore, we also apply the deep learning models to the text processing tasks of lexical stress assignment, syllabification and phonetic transcription in Romanian and compare its results with two traditional machine learning methods: decision trees and multilayer perceptron.

The paper is structured as follows: Section 2 presents the text processing tasks and the selected algorithms. Section 3 introduces the performance analysis of the algorithms for each of the tasks, while Section 4 draws some conclusions and discusses future work.

2. Method overview

The most common tasks in speech-enabled applications refer to phonetic transcription, lexical stress assignment and syllabification. The following sections describe each of these tasks into more detail, highlighting the possible problems which may occur when basic rules or machine learning algorithms are used.

2.1. Lexical stress assignment

Stress assignment refers to the process of determining which syllable within a word is accentuated or stressed. The assignment is in general inherent to a word's orthographic form, but within the limited set of homographs, the semantics also play an important role.

In Romanian, the stress pattern is rather variable and depends mostly on the etymology of a word. As a general rule of thumb, the stress can be considered to pertain to the penultimate syllable (Franzen and Horne, 1997) in a word. However, this does not hold true for most derivational and inflexional morphologic forms of the words, as well as to neologisms (e.g. *ac-ce'nt* - accent; *'a-ri-pă* - wing).

2.2. Syllabification

Syllabification is the process of segmenting a word into syllables. Syllables are considered to be the building block of a language and can determine its rhythm and complexity. There are 7 basic rules for syllabification in Romanian, which correspond to various vowel-consonant groups of letters. However, compound and foreign words do not adhere to these rules. There is also the very important issue of hiatus-diphthong ambiguity (Dinu *et al.*, 2013) in Romanian (e.g. *bi-blio-gra-fi-á* - to annotate vs *bi-bli-o-gra-fî-a* - bibliography) .

2.3. Phonetic transcription

The orthographic form of most languages does not entirely reflect its phonology, or the pronunciation of the words. Translating the written form of a word into its acoustic realisation is essential in all speech-enabled applications, as well as in the systems which enable a user to learn a new language.

Romanian is a rather simple language in terms of letter-to-sound rules. Most of the letters have a direct correspondence with a single sound or phoneme³. The most problematic set of letters are the vowels: *e*, *i*, *o* and *u*, which can also be semi-vowels in diphthong and triphthong letter groups (e.g. *oa-ie* - sheep vs *chi-ri-e* - rent). Another set of problematic letters are *c* and *g* which are pronounced differently when followed by the letter groups *he* and *hi*. The letters *b* and *x* also have a different pronunciation depending on their context. If for the last two categories, rules can solve the issue, for the vowels, machine learning methods are required.

2.4. Algorithms

Given each of the tasks enumerated in the previous sections and the state-of-the-art presented in Section 1, we selected two traditional machine learning algorithms: decision trees and

³ We are not taking into account allophones.

multilayer perceptron as being the most representative and accurate across the cited studies. Their performance on these tasks will be compared to that of the deep neural networks.

The classification and decision trees (Breiman *et al.*, 1984) are a class of learning algorithms which make successive binary splits on the training dataset by using adequate attribute values as decisions in their nodes.

The multilayer perceptron (MLP) is a class of feedforward neural networks with at least 3 hidden layers and a non-linear activation function (Hastie *et al.*, 2009). Their performance was state-of-the-art in the mid 80s, but have been replaced by the more simpler Support Vector Machines. MLPs are the predecessors of the more elaborate Deep Neural Networks.

Deep Neural Networks (DNNs) (Bengio *et al.*, 2015) are a class of machine learning algorithms based on neural networks, where the number of hidden layers and nodes per layer is generally large. DNNs have been at the forefront of the machine learning field for the last decade. Their performance on various high complexity tasks have led to a great increase in their use across most of the supervised learning problems. Their main disadvantage has been the need for large amounts of training data, due to the large number of trainable parameters.

3. Results

3.1. Datasets

The evaluation of supervised algorithms requires large quantities of high-quality training data. And the tasks evaluated in this paper are no exception. Therefore, we selected large purposely built lexicons for each of the selected text processing tasks. The lexicons were manually created or checked, and have been previously used in similar evaluations.

The syllabification of approximately 507,000 words was extracted from the RoSyllabiDict lexicon (Barbu, 2008). For stress assignment, we used the DEX Online Database (DEX, 2000) which includes over 1,600,000 words and their inflected forms along with the stress labels. This large list of words was reduced to the subset of words also available in the syllabification corpus (507,000 words), for compute efficiency reasons. The MaRePhor phonetic dictionary (Toma *et al.*, 2017) and its approximately 72,000 words represented the training data for our phonetic transcription experiments.

3.2. Features

As creating highly specialised input features can be computationally expensive, we resumed the use of integer encodings of the graphemes, augmented with simple positional and categorical features. The categorical features refer to the use of the vowel-consonant classification of the graphemes, while the positional features refer to the length of the word, as well as to the location of the current letter within the word as a forward percentage of the total number of the letters.

We used 2 strategies for defining the grapheme input features: left-right N -letter context, and entire word. The first strategy refers to extracting groups of letters of length $2N+1$ from the current word. The letter situated in the centre of the group is the one for which the prediction is made. For consistency across the words, 0-padding was used. The length of the contexts or

window was set to 3, 5, or 7 letters. In the entire-word prediction strategy, we created a 0-padded sequence of length equal to the length of the longest word in our lexicons (25 letters).

In lexical stress assignment, we did not construct input features which had a central consonant, as the stress is only assigned to vowels. The prediction for each input feature set stated if the current central vowel is stressed or not—for window contexts—, or which letter is stressed in the case of entire word prediction.

For syllabification, the output label marked if the current central letter is a syllable boundary or not. In the case of entire word prediction, the syllable boundaries were encoded as an integer obtained from the conversion of a binary number of 25 bits, where each binary digit encoded the syllable boundary presence at the respective index in the word.

The case of phonetic transcription is slightly different, as the entire-word strategy is hard to encode using simple classification methods. As a consequence, we do not report results for this strategy. Also, as only some of the letters in Romanian pose pronunciation problems, we evaluated the algorithms for the case of predicting all the letters, as well as only the problematic ones.

In all three text processing tasks' the use of letter types (consonant/vowel) was also evaluated. This means that along with the graphemes, the input features included the sequence of vowel-consonant types for each of the graphemes within the context window or word, respectively.

3.3. Implementation

All the evaluations were carried out using Python implementations of the algorithms. For the decision trees, we used the Python Scikit⁴ DecisionTreeClassifier⁵, with default parameters. The MLPs were implemented with the Python Scikit MLPClassifier, using three hidden layer of sizes equal to a halving progression: layer 1 - the length of the input features, layer 2 - half the length of the input features, layer 3 - a quarter of the length of the input features, and a ReLU activation function.

The Deep Neural Networks were implemented in Keras⁶ with a fully connected forward architecture of 3 hidden layers, similar to the MLP: layer 1 – half the size of the input features, layer 2 – a quarter of the size of the input feature, layer 3 – 10 nodes. In between each layer there was a dropout layer of 20% of the neurones. The output layer was a softmax. The training was performed on an nVIDIA 970 GPU. The training stopped after no increase in accuracy for 2 consecutive epochs. The network used an Adam optimiser (Kingma and Ba, 2014) and the loss was set to categorical cross-entropy.

3.4. Algorithm evaluation

The tables below present the accuracy of the 3 algorithms with respect to each of the 3 text processing tasks. We evaluated their accuracy and F-measure on a held-out randomly selected subset of 20% of the data. In the cases where the algorithms showed very similar accuracy across the evaluation sets, we report a 4 digit precision result. WIN3, WIN5 and WIN7 refer to the length of the context window.

⁴ <http://scikit-learn.org/stable/>

⁵ <http://scikit-learn.org/stable/modules/tree.html>

⁶ <https://keras.io/>

3.5. Discussion

The results presented in the previous section show that the MLP is the worst performing algorithm. It might be true, however, that the tuning of its hyper-parameters could lead to a better performance. However, this was not the purpose of this study.

When comparing the decision trees with the DNNs it seems that their performance is quite similar, and in some limited cases, the decision trees marginally outperform DNNs.

However, the DNNs have proven their efficiency in the entire-word prediction task, where their accuracy is clearly above the one of the decision trees. It is also encouraging that the DNNs had an accuracy of at least 96% on each of the tasks, and in some cases, this accuracy almost reached 100%.

Table 1: Stress assignment evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type		Without letter type	
		Accuracy	F-measure	Accuracy	F-measure
Decision Trees	WIN3	0.97	0.92	0.96	0.92
	WIN5	0.97	0.93	0.97	0.93
	WIN7	0.97	0.93	0.97	0.93
	Entire word	0.92	0.92	0.91	0.91
MLP	WIN3	0.88	0.72	0.84	0.56
	WIN5	0.92	0.79	0.87	0.65
	WIN7	0.92	0.82	0.87	0.66
	Entire word	0.79	0.74	0.52	0.46
DNNs	WIN3	0.96	0.92	0.96	0.90
	WIN5	0.98	0.95	0.97	0.95
	WIN7	0.98	0.95	0.98	0.95
	Entire word	0.96	0.96	0.95	0.95

Table 2: Syllabification evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type		Without letter type	
		Accuracy	F-measure	Accuracy	F-measure
Decision Trees	WIN3	0.99	0.99	0.99	0.99
	WIN5	0.99	0.99	0.99	0.98
	WIN7	0.99	0.99	0.99	0.98
	Entire word	0.86	0.86	0.67	0.67
MLP	WIN3	0.96	0.93	0.78	0.67
	WIN5	0.96	0.95	0.81	0.70
	WIN7	0.97	0.96	0.83	0.74
	Entire word	0.65	0.63	0.30	0.21
DNNs	WIN3	0.99	0.99	0.99	0.99
	WIN5	0.99	0.99	0.99	0.99
	WIN7	0.99	0.99	0.99	0.99
	Entire word	0.98	0.97	0.97	0.97

A COMPARISON BETWEEN TRADITIONAL MACHINE LEARNING APPROACHES AND DEEP NEURAL NETWORKS FOR TEXT PROCESSING IN ROMANIAN

Table 3: Phonetic transcription - **problematic letters** - evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type		Without letter type	
		Accuracy	F-measure	Accuracy	F-measure
Decision Trees	WIN3	0.9882	0.9881	0.9859	0.9858
	WIN5	0.9877	0.9875	0.9837	0.9837
	WIN7	0.9868	0.9868	0.9825	0.9824
MLP	WIN3	0.86	0.84	0.80	0.73
	WIN5	0.90	0.88	0.86	0.80
	WIN7	0.93	0.92	0.87	0.83
DNNs	WIN3	0.99	0.99	0.98	0.98
	WIN5	0.99	0.99	0.99	0.99
	WIN7	0.99	0.99	0.99	0.99

Table 4: Phonetic transcription - **all letters** - evaluation in terms of accuracy and F-measure

Algorithm	Features	With letter type		Without letter type	
		Accuracy	F-measure	Accuracy	F-measure
Decision Trees	WIN3	0.9944	0.9944	0.9947	0.9947
	WIN5	0.9940	0.9939	0.9924	0.9924
	WIN7	0.9936	0.9937	0.9927	0.9926
MLP	WIN3	0.62	0.58	0.63	0.51
	WIN5	0.93	0.92	0.68	0.58
	WIN7	0.96	0.96	0.84	0.80
DNNs	WIN3	0.9958	0.9956	0.9957	0.9957
	WIN5	0.9963	0.9961	0.9963	0.9963
	WIN7	0.9960	0.9960	0.9950	0.9947

The letter type input features did not influence the results of the tasks or algorithms overall, but they did have a major impact in the syllabification task, especially for the MLP. This was to be expected, as most of the syllabification rules in Romanian use the vowel-consonant structure of the word.

In phonetic transcription, the accuracy of predicting only the problematic letters is comparable to the one where all the letters are predicted, even though these are only a small subset of the entire data. Predicting the entire set of letters would also impose additional computation expenses.

4. Conclusions

This paper presented a brief comparison between traditional machine learning algorithms: decision trees and multilayer perceptron, and the deep neural networks applied to three different text processing tasks in Romanian. The tasks included stress assignment,

syllabification and phonetic transcription. The algorithms were compared in terms of accuracy and F-measure on several different input feature sets. The use of context windows versus entire word prediction strategy was evaluated, along with augmenting the input with positional and categorical features.

Results showed that the decision trees can achieve comparable accuracy with the DNNs, while the MLP is the least well performing algorithm out of the 3. Also, using additional features led to better results. The window length did not play such an important role in the accuracy, but window contexts performed better than the entire-word strategy.

As future work, in terms of traditional machine learning mechanisms, other algorithms could be evaluated, such as support vector machines. However, previous work showed that decision trees and multilayer perceptrons are best suited to these classification tasks.

Also, combining the features could lead to better accuracies of individual tasks. For example, the syllabification and stress assignment are important for phonetic transcription. The part-of-speech tagging is essential for diacritic restoration, and so on.

There is still a large amount of analysis to be performed with respect to the architecture of the MLP and DNNs and the tuning of their hyper-parameters. One other important issue is the use of sequence-to-sequence learning for all the tasks concurrently, as well as evaluating the speed of the algorithms when used in a complete processing system, and the overhead that they introduce in it.

Acknowledgement

This work was partially supported by a grant of the Romanian Ministry of Research and Innovation, PCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73PCCDI, within PNCDI III.

References

- Barbu, A.M. (2008). Romanian Lexical Data Bases: Inflected and Syllabic Forms Dictionaries, In *Proc of Language Resources and Evaluation Conference*, Marrakech, Morocco.
- Bengio, Y., LeCun, Y., Hinton, G. (2015). Deep Learning, *Nature* 521(7553): pp. 436–444.
- Breiman, L., Friedman, J., Stone, C.J., Olshen, R.A. (1984). Classification and Regression Trees. *Taylor & Francis*.
- Boroș, T., Ștefănescu, D. and Ion, R. (2012). Bermuda, a data-driven tool for phonetic transcription of words, In *Proc. of the Workshop on Natural Language Processing for Improving Textual Accessibility*, Barcelona, Spain.
- Boros, T., Dumitrescu, S.D. and Pipa, S. (2017). Fast and Accurate Decision Trees for Natural Language Processing Tasks. In *Proc. of Recent Advances in Natural Language Processing*, pp 103-110, Varna, Bulgaria.
- Boroș, T., Dumitrescu, S.D. and Pais, V. (2018). Tools and resources for Romanian text-to-speech and speech-to-test applications, *arXiv*, 1802.05583v1
- Burileanu, D. (2002). Basic Research and Implementation Decisions for a Text-to-speech Synthesis System in Romanian. *International Journal of Speech Technology*, no 5, pp. 211-225

A COMPARISON BETWEEN TRADITIONAL MACHINE LEARNING APPROACHES AND DEEP
NEURAL NETWORKS FOR TEXT PROCESSING IN ROMANIAN

- Chitoran, I., Ciobanu, A.M., Dinu, L. and Niculae, V. (2014). Using a Machine Learning Model to Assess the Complexity of Stress Systems. In *Proc. of the Language Resources and Evaluation Conference*, Reykjavik, Iceland.
- Ciobanu, A., Dinu, A., Dinu, L. (2014). Predicting Romanian Stress Assignment. In *Proc. of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pp 64-68, Gothenburg, Sweden.
- Cucu, H., Buzo, A., Besacier, L. and Burileanu, C. (2014). SMT-based ASR domain adaptation methods for under-resourced languages: *Application to Romanian*. *Speech Communication*, no 56, pp 195-212.
- DEX online (2018), <http://www.dexonline.ro>
- Dinu, L.P., Niculae, V. and Sulea, O.-M. (2013). Romanian Syllabication using Machine Learning. In *Proc. of the International Conference on Text, Speech and Dialogue*, pp. 450-456.
- Domokos, J., Buza, O. and Todorean, G. (2011). Automated grapheme-to-phoneme conversion system for the Romanian Language, In *Proc. of the 6th Conference on Speech Technology and Human-Computer Dialogue*, Brasov, Romania.
- Domokos, J., Buza, O. and Todorean, G. (2012). 100k+ Words, Machine-Readable, Pronunciation Dictionary for the Romanian Language, In *Proc of the 20th European Signal Processing Conference*, Bucharest, Romania.
- Franzen, V. and Horne, M. (1997). Word stress in Romanian, *Working Papers* 46, pp 75-91.
- Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. *Springer, New York, NY*, 2009.
- Jitca, D., Teodorescu, H.N., Apopei, V. and Grigoras, F. (2003). An ANN-based method to improve the phonetic transcription and prosody modules of a TTS system for the Romanian language. In *Proc. of the 2nd Conference on Speech Technology and Human-Computer Dialogue*.
- Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv* 1412.6980.
- Oancea, E., Badulescu, A. (2002). Stressed Syllable Determination for Romanian Words within Speech Synthesis Applications. *International Journal of Speech Technology*, no 5, pp 237-246.
- Ordean, M., Saupe, A., Ordean, M., Duma, M. and Silaghi, Gh. (2009). Enhanced Rule-Based Phonetic Transcription for the Romanian Language. In *Proc. of the 11th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, Timisoara, Romania.
- Stan, A., Yamagishi, J., King, S., Aylett, M. (2011). The Romanian speech synthesis (RSS) corpus: Building a high quality HMM-based speech synthesis system using a high sampling rate, In *Speech Communication*, vol. 53, no. 3, pp. 442-450.
- Stan, A., Dinescu, F., Tiple, C., Meza, S., Orza, B., Chirila, M., Giurgiu, M. (2017). The SWARA Speech Corpus: A Large Parallel Romanian Read Speech Dataset, In *Proceedings of the 9th Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, Bucharest, Romania.
- Toma, S.-A., Oancea, E. and Munteanu, D.-P. (2009). Automatic rule-based syllabication for Romanian. In *Proc of the 5th Conference on Speech Technology and Human-Computer Dialogue*, Constanta, Romania

- Toma, S.T., Bârsan, T., Totir, F., Oancea, E. (2013). On letter to sound conversion for Romanian: A comparison of five algorithms. In *Proc. of the 7th Conference on Speech Technology and Human-Computer Dialogue*, Cluj-Napoca, Romania.
- Trandabăţ, D., Irimia, E., Barbu Mititelu, V., Cristea, D. and Tufiş, D. (2012). The Romanian Language in the Digital Era. *Springer, Metanet White Paper Series*.
- Ungurean, C. and Burileanu, D. (2011). An Advanced NLP Framework for High-Quality Text-to-Speech Synthesis. In *Proc. of the 6th Conference on Speech Technology and Human-Computer Dialogue*, Brasov, Romania.
- Zhang, R., Li, W., Mo, T. (2018). Review of Deep Learning. *arXiv* 1804.01653.

PROSODIC PHRASING BETWEEN SUBJECTIVE PERCEPTION AND OBJECTIVITY

VASILE APOPEI, OTILIA PADURARU

*Romanian Academy - Iasi Branch, Institute of Computer Science
{vasile.apopei, otilia .paduraru}@iit.academiaromana-is.ro*

Abstract

Prosodic phrasing of the utterances can be interpreted as a level of suprasegmental linguistic representation that indicates how the words are grouped in speech and is influenced by syntactic, semantic and pragmatic/discourse factors. This representation of speech facilitates the transmission and reception of information between people and helps to structure the ideas in the communication and their compression at a mental level. The goal of this paper is to present a new perspective on the relationship between speech rhythm and prosodic phrasing based on the Spectral Amplitude Modulation Phase Hierarchy (SAMPH) model developed in the Centre for Neuroscience of the University of Cambridge.

Key words — prosodic phrasing, subjective perception, SAMPH model, speech rhythm

1. Introduction

This study is an analysis of transcription objectivity of the prosodic elements in theoretical frameworks of autosegmental-metrical phonology (Goldsmith, 1990) and of intonational phonology proposed by Pierrehumbert and colleagues (Beckman and Pierrehumbert, 1986), (Pierrehumbert, 1980), (Pierrehumbert and Beckman, 1988). Prosodic phrasing analyzed in this way use a hierarchy of prosodic domains, domains subordinated through relative prosodic prominences (Selkirk, 2005).

Most research about prosodic annotation are based on auditory perception (Chomsky and Halle, 1968), (Frazier, 2006), (Romportl, 2010), (Cole and Shattuck-Hufnagel, 2016) and on the notion of "*nuclear / prominent accent*" introduced by the British School of Phonetics and Phonology (Chomsky and Halle, 1968), (Fox, 2000), (Ladd, 2008). Many authors recognize that the nuclear focus/stress is difficult to identify and that should not be confused with the prominences on the F0 frequency contour, but it must be associated with the last accent before the boundary tone of an Intonational Phrase (Jun, 2003), (Ladd, 2008), (Gut, 2009).

In recent years, researchers give more and more importance to prosody with applications into human speech communication and for the development of automatic spoken language systems. An essential component of any investigation of prosody is the consistency of prosodic transcriptions of the speech corpora. Inconsistency of prosodic transcriptions/annotations can be generated by the set of labels of the prosodic annotation system and by differences between annotators in interpretation and labelling of the prosodic elements. Roy *et al.* (2017) asserts that "A prosodic annotation represents a subjective interpretation of the prosodic substance of an utterance, typically that of a trained annotator, that transduces the acoustic speech signal onto a symbolic representation in terms of a small set of prosodic features". A prosodic annotation systems aims to marking the prosodic structure (called in the literature and prosodic form) of the utterances, in terms of the labels

that encode prosody, and the associated function of that prosodic form in conveying linguistic meaning (Cole *et al.*, 2016).

In the last years, more and more researchers are seeking to identify the acoustic, lexical, syntactic and semantic features, in order to facilitate to objectify labelling of the prosodic elements and prosodic phrase (Romportl, 2010), (Rosenberg, 2012), (Mertens and Simon, 2013), (Apopei, 2015), (Christodoulides *et al.*, 2017), (Christodoulides, 2016). The identification of prosodic phrasing models is of interest for vocal recognition and synthesis, as well as for the identification of emotions transmitted by voice.

In our opinion, prosodic phrasing, can be interpreted as a level of suprasegmentally linguistic representation that indicates how the words are grouped in speech production and perception, facilitates the transmission and reception of information between people and helping to structure the ideas in the communication and their comprehension at the level mental (Frazier *et al.*, 2006), (Paige *et al.*, 2017). Prosodic phrasing of the utterances, as a part of the mental representation of speech (Lambrecht, 1994) and intonation (Kimball and Cole, 2016) is learned throughout life based on auditory perception (Gut, 2009) and is influenced by syntactic, semantic and pragmatic/discourse factors (Jun, 2003).

The goal of this paper is to present an original method for detection of prosodic boundary based on the speech rhythm. The researches represents a continuation of those presented in Apopei and Paduraru (2015) and is based on the Strong–weak rhythm patterns of the Stress tier computed using the SAMPH model (Leong, 2012).

2. *The current state of the art in automatic prosodic phrasing and speech rhythm*

The current systems of automatic transcription of the prosodic elements are based on automatic learning systems and prosodically annotated corpora with transcripts made by expert annotators (Braunschweiler, 2006), (Romportl, 2010), (Rosenberg, 2012), (Mertens and Simon, 2013), (Christodoulides *et al.*, 2017). Recent analysis of the various prosodically annotated corpora highlight difficulties in learning of the transcription systems of prosody due to inconsistency in attribution of the prosodic labels and in prosodic structure of utterances. In addition, most of the prosodically annotated corpora are composed from sentences spoken separately, which creates difficulties in the automatic annotation of the prosodic elements on the spontaneous speech taken from TV or radio broadcasts, theater plays etc.

Automatic prosodic transcription systems use for training a gold prosodically annotated corpora, acoustic features and linguistic information (Christodoulides *et al.*, 2017). The most used acoustic features estimated from speech are based on the contour of the fundamental frequency (F0), speech intensity and duration at the syllable and word level. The use of linguistic information (lexical, syntactic and semantic features) in the training of automatic prosodic transcription systems is justified by researchers in the fact that expert annotators integrate all this information into their perception of segmentation and annotation of prosodic events.

If until year 2010 the term "rhythm" was used fairly little in speech analysis (Cumming, 2010), after this year, the term began to appear in syntagms such as: „*speech rhythm*” refer as the alternating pattern of 'Strong' (stressed) and 'weak' (unstressed) of the syllables in speech

(Leong, 2012); “*rhythmic phrase*” with “prosodic phrase” sense (Lai *et al.*, 2016); „*rhythmic group*” with “prosodic word” sense (Cumming, 2010).

Researches about link between rhythm and prosody have been made and in the European AMPER project (Contini *et al.*, 2002). The papers published by the participants in this project focused mainly on speech rhythm measurement and the study of rhythm variability in various Romance languages.

3. Speech rhythm in prosodic phrasing

In this section we present a new perspective on the relationship between speech rhythm and prosodic phrasing based on the Spectral Amplitude Modulation Phase Hierarchy (SAMPH) model developed in the Centre for Neuroscience of the University of Cambridge, by a researchers group coordinated by Usha Goswami (Leong, 2012), (Leong *et al.*, 2014). The model is based on correlation between the delta, theta, alpha and beta cortical oscillations (Cervenka *et al.*, 2011) and speech modulation rates at three levels: 'Stress' (0.9 - 2.5 Hz), 'Syllable' (2.5 – 12 Hz) and 'Phoneme' (12 - 40 Hz) rates.

Analysis of the Strong–weak rhythm patterns from the Stress tier computed using the SAMPH model, highlights that the envelope contour shows variations in the range for strong elements and trend of minimum points on the envelope. Based on these observations on the Stress tier envelope of utterances, we tried to put the points of change of the trend in correlation with audio segments corresponding to words. We have found that this point of change of the trend correspond to boundary of prosodic phrases.

We exemplify our observation by marking the trend of the maximum and, minimum points respectively on the envelope contour of the Stress tier on an utterance of the text (1) and on a segment of the audio recording from the Speech Archives at the Sorbonne University in Paris made in 1913 by Elena Vacarescu (4).

Odată ce stadiul cunoscut al cercetării este bine analizat și argumentat, este posibilă introducerea celorlalte componente similare planului de afaceri iar în fază finală face sincronizarea resurselor și rezultatelor exprimate prin obiectivele proiectului.

(1)

Once the known stage of the research is well analyzed and argued, it is possible to introduce the other components similar to the business plan and at the final stage it makes the synchronization of the resources and the results expressed through the objectives of the project.

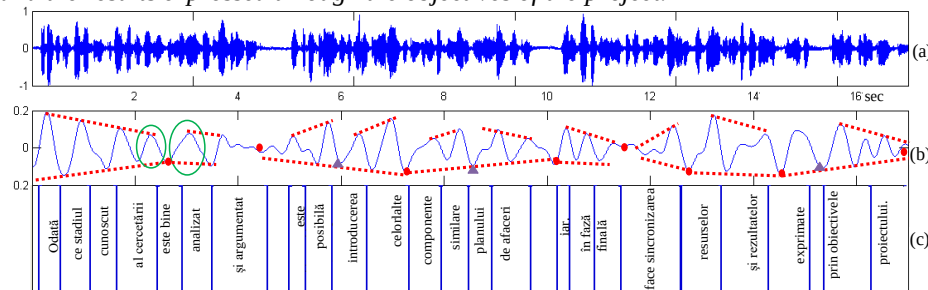


Figure 1: (a) the waveform of an utterance of the text (1); (b) the envelope of the Stress tier on the utterance of the text (1); (c) delimiting the speaking segments corresponding to the words of the text (1).

In Figure 1 we present: (a) the waveform of an utterance of the text (1); (b) the envelope of the Stress tier on the utterance of the text (1) that we marked: with the red dashed line the trend of the minimum points on the envelope and the maximum points respectively; with red

circle we have marked the minimum points, corresponding to the weak accents, where the trend of the minimum points on the outline of the tire is changed; with magenta triangle we have marked the minimum points in which the trend of the maximum points on the outline of the tire is changed; (c) delimiting the speaking segments corresponding to the words of the text (4).

By matching the points of where the trend of the minimum points on the outline of the tire is changed, marked with a red circle on the tire (Figure 1b), with the speech segments outlined in Figure 1c, we obtain the segmentation of the utterance marked on the text with the curly braces (2).

{Odată ce stadiul cunoscut al cercetării} {este bine analizat și argumentat,}
 {este posibilă introducerea celorlalte} {componente similare planului de afaceri} {iar în fază finală} {face (2)
 sincronizarea} {resurselor și rezultatelor} {exprimate prin obiectivele proiectului.}

We continue the segmentation process of the utterance by taking into account the boundaries marked by the points where the trend of the maximum points on the outline of the tire is changed, marked with a magenta triangle on the tire (Figure 1b). By marking on the segmentation (2) of these boundaries with square brackets we obtain segmentation (3).

{[Odată ce stadiul cunoscut al cercetării]} {[este bine analizat și argumentat,]}
 {[este posibilă][introducerea celorlalte]} {[componente similare] [planului de afaceri]} (3)
 {[iar în fază finală]} {[face sincronizarea]} {[resurselor și rezultatelor]}
 {exprimate [prin obiectivele proiectului.]}

At this stage of the prosodic phrasing processing we consider that curly braces mark the beginning and end of the major phonological/prosodic phrases and square brackets delimit the minor phonological/prosodic phrases, according to the prosodic domains hierarchy for English proposed by Selkirk (2005). The boundaries of the minor and major prosodic phrases determined by using this algorithm are the same with those determined by using features extracted from F0 and energy contours (Apopoi and Paduraru, 2015). The minor prosodic phrases can be put in correlation with syntactic phrases (NP, VP etc.). The major prosodic phrases are groups of minor prosodic phrases, perceived auditory, which correspond to the utterance of more complex syntactic structures (Cole, 2015, Apopei, 2014).

In the following, we will present results obtained in prosodic phrasing on a speech with noise from an audio recording from the Speech Archives at the Sorbonne University in Paris made in 1913 by Elena Vacarescu. We must mention that on this record the algorithms for pitch extraction fail to correctly estimate the fundamental frequency. We present in this paper a segment of this record corresponding to the text (4).

In Figure 2 we present: (a) the waveform corresponding to the text (4); (b) the envelope of the Stress tier on the utterance of the text (4) that we marked: with the red dashed line the trend of the minimum points on the envelope and the maximum points respectively; with red circle we have marked the minimum points, corresponding to the weak accents, where the trend of the minimum points on the outline of the tire is changed; with magenta triangle we have marked the minimum points in which the trend of the maximum points on the outline of the tire is changed; (c) delimiting the speaking segments corresponding to the words of the text (4).

Vechea și strălucita familie boierească Văcărescu, din care cu mândrie mă trag, a dat țării trei poeți
 Alecu, Ienăchiță și Iancu Văcărescu. De Ienăchiță Văcărescu voi spune mult cunoscutul lui Testament. (4)

The old and brilliant boyar family Vacarescu, from which I proudly come, gave the country three poets
 Alecu, Ienachita and Iancu Vacarescu. By Ienachita Văcărescu I will tell the much-known his

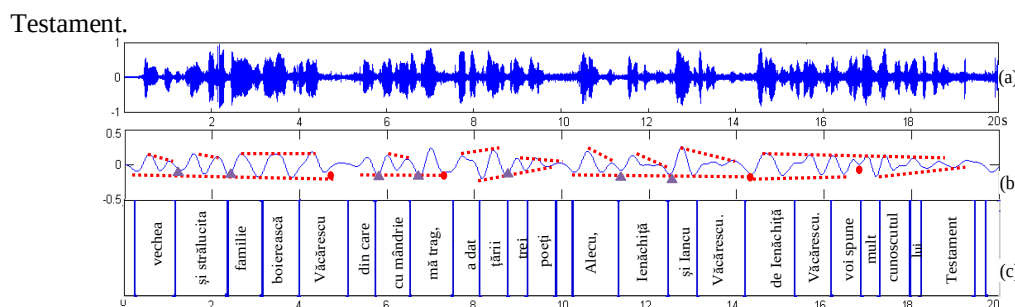


Figure 2: (a) the waveform of an utterance of the text (4); (b) the envelope of the Stress tier on the utterance of the text (4); (c) delimiting the speaking segments corresponding to the words of the text (4).

By matching the points of where the trend of the minimum and maximum points on the outline of the tire is changed, marked with a red circle and magenta triangle respectively on the tire (Figure 2b), with the speech segments outlined in Figure 2c, we obtain the segmentation of the utterance marked on the text with the curly braces and curly braces respectively (5).

$$\begin{aligned}
 & \{[Vechea][\text{și strălucita}][\text{ familie boierească Văcărescu,}]\} \{ \text{din care} [\text{cu mândrie}] \text{ mă trag,} \} \\
 & \{[a dat \text{ țării}][\text{ trei poet}]\} \{[Alecu], [Ienăchiță] [\text{și Iancu Văcărescu.}]\} \\
 & \{[De Ienăchiță Văcărescu \text{ voi spune}]\} \{[mult \text{ cunoscutul lui Testament}]\}.
 \end{aligned} \tag{5}$$

The results obtained in prosodic phrasing, using the new method based on the speech rhythm, on this record confirm that the method is robust and give good results even for the speech with noise.

We have tested the new method of prosodic phrasing of the speech on the following categories of audio recordings:

- The audio recordings taken from the internet network, the recordings of some personalities' speeches that marked the events of year 1918:
- The audio recording about the Resolution of the National Assembly in Alba Iulia on Nov 18/ Dec 1, 1918, made by (re) reading by the Iuliu Hossu, Romanian Greek-Catholic cardinal, in the 1970s, when he was isolated at Căldărușani Monastery.
- The audio recording made by Elena Văcărescu in May 1913, to the Archives of Speech from the Sorbonne University of Paris.
- The audio recordings made by Nicolae Iorga in January 1927 - "Future Generations Have to Capitalize the National Cultural Fund" and in 1929 - "Civilization and Tradition" to the Archives of Speech from the Sorbonne University of Paris.
- The voice from the Nicolae Iorga video recording from the '30s with the discourse, "Our country's duty is to work".
- The audio recording with the end speech of the Minister of Finance Nicolae Titulescu in front of the Deputies' Assembly in the year of 1921, speech on the disk "Idealul creator".
- Audio recordings taken from the COROLA corpus (<http://corola.racai.ro/>). Speech from the recording of the TV shows „Without Replica" and "Labyrinth of Success" broadcasts by Roman TV.
- Audio recordings of the isolated utterances, recordings made by several native Romanian speakers.

Auditory analysis of the speech segments delimited, on all audio recordings listed above, on the basis of the information extracted from the contour of the Strong–weak rhythm patterns of the Stress tier, computed using the SAMPH model, have trusted us that the perception of prosodic phrasing is objectively influenced by the modulation of the speech rhythm at the Stress level.

4. Conclusions

Auditory analysis of the speech segments delimited on the basis of the information extracted from the contour of the Strong–weak rhythm patterns of the Stress tier, computed using the SAMPH model, led us to the conclusion that these segments are consistent with prosodic phrases perceptually identified. The obtained results point to the fact that the boundaries of the prosodic phrases perceived auditory have coverage at the level of features extracted from the speech rhythm and that the method gives very good results in prosodic phrasing. This method gives good results in automatic prosodic phrasing even in speech with noise.

The results presented in this paper reinforce us the belief that from the point of view of auditory perception, prosodic phrasing is objectively influenced by the modulation of the speech rhythm at the Stress level and less by the contour of frequency F0. Frequency contour F0 "enriches" expression of phrasing but does not determine it alone (Kochanski *et al.*, 2005). This "enrichment" from our point of view refers to increasing the perception of the prominent words and the prosodic phrases. If for the prosodic phrasing based on the frequency contour F0 it is necessary to use a line-of-mouth alignment, the new method does not require this alignment. Alignment can sometimes be used to precisely delimitate the boundaries of prosodic phrases.

References

- Apopei, V. and Paduraru O. (2015). *Towards Prosodic Phrasing of Spontaneous and Read Speech for Romanian Corpora*. Proc. 8th Conf. Speech Technology and Human - Computer Dialogue (SPED 2015), Bucharest.
- Apopei, V. (2014). *About prosodic phrasing of the Noun Phrases in speech*. Proceedings of the Romanian Academy, Volume 15, Number 2, 200-207.
- Beckman, M. and Pierrehumbert, J. (1986). *Intonational structure in Japanese and English*. Phonology Yearbook, 3, 255–309.
- Braunschweiler, N. (2006). *The Prosodizer - Automatic Prosodic Annotations of Speech Synthesis Databases*. Proceedings Speech Prosody, Dresden.
- Cervenka, M.C., Nagle, S. and Boatman-Reich, D. (2011). *Cortical High-Gamma Responses in Auditory Processing*. American Journal of Audiology, Vol. 20, 171–180.
- Christodoulides, G., Avanzi, M. and Simon, A. C. (2017). *Automatic Labelling of Prosodic Prominence, Phrasing and Disfluencies in French Speech by Simulating the Perception of Naïve and Expert Listeners*. INTERSPEECH 2017, 3936-3940.
- Christodoulides, G. (2016). *Effects of Cognitive Load on Speech Production and Perception*. PhD Thesis, Université catholique de Louvain, Belgium.
- Chomsky, N. and Halle, M. (1968). *The sound pattern of English*. Harper & Row Publishers, New York.

- Cole, J. (2015), *Prosody in Context: A review*, Language, Cognition and Neuroscience, 30 (1-2), 1-31.
- Cole, J. and Shattuck-Hufnagel, S. (2016). *New Methods for Prosodic Transcription: Capturing Variability as a Source of Information*. Laboratory Phonology: Journal of the Association for Laboratory Phonology 7(1): 8, pp. 1–29.
- Contini, M., Lai, J.P. and Romano, A (2002). *La géolinguistique à Grenoble : de l'AliR à AMPER*. In Simoni-Aurembou M. R. (eds.), *Revue Belge de Philologie et d'Histoire*, 80/3, 931-941.
- Cumming, R. (2010), *Speech Rhythm: The language-specific integration of pitch and duration*. Doctoral dissertation, University of Cambridge.
- Frazier, L, Carlson, K and Clifton, C. Jr (2006). *Prosodic phrasing is central to language comprehension*. Trends in Cognitive Science. Vol.10 No.6, pp. 244-249.
- Fox, A. (2000). *Prosodic Features and Prosodic Structure*. Oxford University Press.
- Goldsmith, J. (1990). *Autosegmental and metrical phonology*. Oxford: Basil Blackwell.
- Gut, U. (2009). *Introduction to English Phonetics and Phonology*. Peter Lang Publisher.
- Iversen, J. R., Patel, A. D. and Ohgushi K. (2009). *Perception of rhythmic grouping depends on auditory experience*. Journal of the Acoustical Society of America, 124(4), 2263-2271
- Jun, S.-A. (2005), *Prosodic Typology: The Phonology of Intonation and Phrasing*, Oxford University Press.
- Jun, S.-A (2003), *Prosodic phrasing and attachment preferences*. Journal of Psycholinguistic Research, 32(2), 219–249.
- Kimball, A. E. and Cole, J. (2016). *Pitch contour shape matters in memory*. Proceedings of the International Conference on Speech Prosody, 2016-January, pp. 1171-1175.
- Knud, L. (1994). *Information Structure and Sentence Form: Topic, Focus, and the Mental Representation of Discourse Referents*. Cambridge: Cambridge University Press.
- Kochanski, G., Grabe, E., Coleman, J. and Rosner B. (2005), *Loudness predicts prominence: Fundamental frequency adds little*. Journal of the Acoustical Society of America, 118, 1038–1054.
- Ladd, D.R. (2008). *Intonational Phonology*. Cambridge: Cambridge University Press.
- Lai, W., Yuan, J., Li, Y., Xu X. and Liberman M. (2016). *The rhythmic constraint on prosodic boundaries in Mandarin Chinese based on corpora of silent reading and speech perception*, INTERSPEECH 2016, 87-91.
- Leong, V. (2012). *Prosodic rhythm in the speech amplitude envelope: Amplitude modulation phase hierarchies (AMPHs) and AMPH models*. Doctoral dissertation, University of Cambridge .
- Leong V., Stone M. A., Turner R. E., Goswami U. (2014). *A role for amplitude modulation phase relationships in speech rhythm perception*. J. Acoust. Soc. Am., vol. 136, no. 1, 366–381.
- Mertens, P. and Simon, A.C (2013). *Towards automatic detection of prosodic boundaries in spoken French*. In P. Mertens and A.C. Simon [Eds], *Proceedings of the Prosody – Discourse Interface Conference*, pp. 81–87
- Paige, D.D., Rupley, W. H., Smith, G. S., Rasinski, T. V., Nichols, W. and Magpuri-Lavell, T. (2017). *Is prosodic reading a strategy for comprehension?*. Journal of Educational Research Online 9 (2), Volume 9, No. 2, 245-275.
- Pierrehumbert, J. and Beckman M. (1988). *Japanese tone structure*. Cambridge, MA: MIT Press.
- Pierrehumbert, J. (1980). *The phonology and phonetics of English intonation*. Ph.D. Dissertation, MIT.
- Reichel, U. D. (2014). *Linking bottom-up intonation stylization to discourse structure*. Computer Speech and Language, 28, 1340–1365.
- Romportl, J. (2010). *On The Objectivity Of Prosodic Phrases*. The Phonetician, 96, 7-19.

- Roy, J., Cole, J. and Mahrt, T. (2017). *Individual differences and patterns of convergence in prosody perception*. *Laboratory Phonology: Journal of the Association for Laboratory Phonology* 8(1): 22, 1–36.
- Selkirk E. (2005). *Comments on intonational phrasing in English*. In *Prosodies: With Special Reference to Iberian Languages*, S. Frota, M. Vigario, and M.J. Freitas (eds.), Berlin: Mouton de Gruyter, 11-58.

CHAPTER 2

TEXT PROCESSING

ON TEXT PROCESSING AFTER OCR

ALEXANDRU COLESNICOV, SVETLANA COJOCARU, LUDMILA MALAHOV,
LYUDMILA BURTSEVA

*Institute of Mathematics and Computer Science,
5 Academiei str., Chisinau, MD-2028, Moldova*
{alexandru.colesnicov, svetlana.cojocar, ludmila.malahov, lyudmila.burtseva}@math.md

Abstract

Digitization of historical texts is more and more significant but meets many problems. To obtain the reliable and useful electronic version of an old text, we should minimize errors in the resulting digital text. Old texts are degraded, and OCR itself can't achieve necessary recognition quality. The post-OCR text improvement may be performed by many techniques, from manual correction to semi-automated and automated approaches.

The authors review existing experience on text correction after OCR, including their own OCR of old Romanian texts in the Cyrillic script since the 17th century. Possible approaches and use of specific and general-purpose software are discussed. We conclude that promising theoretical research didn't result in implementation of the specialized handy software products, and that at least expandable integrating software platform would be demanded.

Key words — OCR, historical text digitization, old Romanian Cyrillic script, post-OCR text improvement.

1. Introduction

Digitization of old Romanian texts plays an important part in the worldwide intentions of the preservation of cultural heritage. It includes imaging, recognition, and post-processing of the recognized text (S. Cojocar *et al.*, 2017).

Logically, post-processing of OCRed text is divided in two stages. The first stage is post-processing that is immediately related with the OCR. It can be said that it is post-OCR correction, but we will show that these processes aren't so simple. The second stage is the "target" post-processing, namely, processing that uses the recognized text as data with some definite purpose. Examples are: philological research, like corpora completion, or scientific re-printing; electronic or re-printed public presentation; full-text search, etc.

We'll refer to the first stage as "processing after OCR", or we'll use the expression "post-OCR text improvement". This stage makes the subject discussed in this paper.

The paper consists therefore of three main sections. Sec. 2 contains review of approaches to the text processing after OCR. Sec. 3 presents practical approaches to the topic and our experience in this area. Sec. 4 discusses in even more details spell checking that is a perspective approach that shouldn't be ignored and uses interesting techniques.

2. Improving OCR results: approaches, techniques, and tools

The most important and most generalized approaches to improve OCR quality are: voting, or comparison of the results obtained by applying different OCR algorithms; lexical post-improvement based on characters or words; post-improvement based on word context; post-improvement based on document context (A.Dengel *et al.*, 2000).

Depending of planned efforts for implementation, the first three of generalized approaches may be combined in pairs 1-2, 2-3, or in a triple 1-2-3. Characters obtained as a result of voting may be additionally checked against a dictionary, and then the resulting words may be once more checked combinatorically or syntactically. The fourth generalized approach stands separately.

Voting uses results of several OCR attempts. They may be performed by different software, on different devices, even with different images. The improved text is produced by comparison of results and selection of one of them when they mismatch. Voting works mostly in the character level. Details of its implementation depend on the information that the voting algorithm obtains as a result of recognition.

At the “abstract level”, the output of the recognition process is a character string. For each position, the result is a single character that the OCR algorithm (classifier) identified as the most suitable. Let us take letter “S” as the example. The voting may be then performed: by majority between classifiers; by weighted majority when each classifier has a weight derived, for example, from its previous error rate, etc.

At the “rank level”, the voting algorithm gets for each text position a list of possible characters ordered by decrease of suitability against recognition patterns. In our example, the list may be {‘S’, ‘9’, ‘5’, ‘g’}. The voting may take this information into account, as well as more complicated ways.

At the “measurement level”, the voting algorithm gets for each text position a list of possible characters accompanied with numerical values that were used by the classifier to order the list. These numerical values may represent probabilities, or distances between the glyph extracted from the image and the corresponding pattern, etc. In our example, the voting algorithm may get something like {(‘S’, 0.92), (‘9’, 0.37), (‘5’, 0.09), (‘g’, 0.003)}. There are really a lot of tricky ways to deal with this information from several classifiers.

Voting may result in a simple sequence of characters, or produce ambiguity. If several variants are issued for some positions in text, possibly depending on the neighboring characters, then a suitable representation would be a net of characters (a graph). Implementation of voting may be complicated and expensive, starting with OCR engines that should provide additional information at higher levels.

Lexical post-improvement is the most developed and the most used.

We can mention here bottom-up techniques that work with characters (n -grams, Markov models, etc.), and top-down techniques based on dictionary lookup. Both approaches can use error correction models for sequences of characters, for example, probabilistic models, or word distance measuring models, etc.

Bottom-up techniques exploit the natural language redundancy. An example of such technique is scanning the text for n -grams,

Another technique based on n -grams uses tables of positional binary n -grams. For $n = 2$, a table entry for positional binary 2-grams is $D_{i,j}(a,b) = 1$, if letters a and b exist in positions,

correspondingly, i and j for at least one word in the dictionary, and 0 otherwise. For example, $D_{2,4}('c', 'u') = 1$ for Romanian because of “scaun”. Big size of these tables, especially for $n > 2$, is an obvious obstacle. The corresponding algorithms of error correction are based on Markov's theory.

Dictionary lookup techniques met several difficulties. The first presumable and almost language independent difficulty of all techniques working with words is the segmentation of a text into words. To process the sequence of words, their boundaries should be found without flaws. OCR engine can merge two words, or split one word in two parts, and we could foresee even worse situations.

The language specific local reorganization of text is necessary before getting a word from text for lookup. Hyphenation merges the end of a text line with the beginning of the next line. Some particles may be attached to a word by hyphen or apostrophe, and not always an arbitrary combination of a word and such additional particle is acceptable. An obvious solution is to include such particles in the dictionary but they may have forms that are not used as separate words, and we can skip errors. This is the case, for example, we have the Romanian syntagms (*într-o*, *dintr-o*). In the opposite case, many languages permit to freely combine many affixes and even many words in a single word; in such languages, word structure should be analyzed before dictionary lookup.

When we search for a word in the dictionary, the difficulty is in the big size of the latter. One million words is quite reasonable quantity as we are to include all word forms.

If the word is not found in the dictionary, we should propose candidates for error correction. Both probabilistic and distance measurement techniques can be applied.

A small but important problem arises when the process is user-driven, and the user insists that an absent word is a new correct word, may be, for this specific text only. New words should be integrated in the lookup processes. For example, they should be seen if appropriate in the hint list at visual error correction.

Quick dictionary lookup can be achieved by several techniques. Most popular due to their relative simplicity are: tries; hashing; binary search; indexing.

Any of these dictionary lookup techniques may be combined, in the case of a big dictionary, with dictionary partitioning. The partitioning may be hierarchical, or may take into account word frequencies that permits to keep frequently used words into the main memory, etc.

Word context post-improvement uses two approaches. Statistical approach supposes examining of transitional probabilities between words, or between word categories. Syntactic approach uses parsing techniques to reconstruct the linguistic structure of a phrase. This information helps to eliminate erroneous hypotheses on recognized words.

Document context post-improvement is very restrictive but may play decisive role in quick checking of many similar documents. It exploits preliminary knowledge about the document structure. Example: performing recognition of mail addresses, we are waiting for zip-code or country name in specific positions, and process them correspondingly.

After this theoretical discussion on approaches and techniques of post-OCR improvement, practice needs their implementation. Most of the studied approaches and techniques weren't implemented as publicly available product, nor as library of methods. If your task is to OCR texts you wouldn't develop software but would take some ready solution, or take several

available software products and apply them. We have ABBYY Finereader (AFR) in one of leading positions on the market, and free Open Source products like Tesseract or Ocropsus, etc.

3. Practical approaches to post-OCR text improvement

Table 1 presents the general sketch of OCR and affined processes as we organized them at recognition of old Romanian Cyrillic documents.

Table 1. Sequence and interdependence of processes at document recognition

Action	Input data	Software	Output data
Imaging	Hard copy of text	Imaging software	Scanned text in images (A)
Image improvement	Images (A)	Scan Tailor	Images of better quality (B)
OCR	Images (B) Set of trained patterns (C) Word list (D)	ABBYY Finereader	Recognized text (E) Expanded set of trained patterns (C)
Post-OCR text improvement	Recognized text (E)	MS Word Libre Office Writer Notepad++, MS Excel Spell checkers, etc.	Improved text (F) Expanded word list (D)
Further text usage	Improved text (F)	That necessary to reach targets	Data for target usage

1. Letters (A-F) denote data sets and permit to follow their reuse.
2. AFR permits to improve input images and to edit the resulting text.

To facilitate OCR we need good images of the document. We can more or less regulate the scan quality if we obtain them by ourselves, but in many cases we should use ready-made images, coming, for example, from Internet. AFR provides its internal image editor with a rich set of operations like correction of trapezoid distortions. The image correction in AFR is performed manually. We used also Scan Tailor, a separate image improvement program that works in batch mode. Its set of improvements includes most needed ones like speckle deletion, page rotation, extraction text from original columns, etc.

Before OCR, AFR can perform page segmentation automatically. AFR analyzes different areas of image and assigns them their types (text, table, or image). The structure of a table is additionally analyzed. The result of automatic segmentation may not be satisfactory, implying manual intervention. Old books with their rich decorations are especially difficult. In many cases, we were to perform manual segmentation of all pages of a book.

After this pre=processing, OCR may be started. AFR has two modes of OCR, training or automatic recognition.

For old books, it's usual to train AFR thoroughly for each new document. Trained recognition patterns may be downloaded from AFR and uploaded into it. We can collect and expand sets of recognition patterns.

Performing this for old Romanian documents in the Cyrillic uncial script, we found that in each historical epoch the used fonts depend on typography and don't demonstrate big variability. For example, all religious Romanian printings of the 17th century are covered by two sets of recognition patterns (L.Malahov *et al.*, 2017). This variability grows in time but the situation remains under control.

Old uncial Romanian Cyrillic provides many other peculiarities like notation of numbers by letters, abundance of overline signs, etc. For example, there are regular abbreviations consisting of a letter over another. Each such combination should be trained as a new pattern (A.Colesnicov *et al.*, 2017).

With old Romanian and its Cyrillic script, text improvement has some specifics.

The segmentation of text in words meets almost no problems because since the very first printings in Romanian, words were clearly separated in the text.

As to letters, some of them produced more errors than others. A tendency to split wide letters may be noted. Nevertheless, we obtained accuracy greater than 99% (in characters) with training and constant extension of word lists.

Let us take as an example word statistics for an 18th century Romanian text on geography (Amf.Hotiniul, 1795). It contains in total 5,066 words. Comparing with modern orthography, 613 words (12.1%) differ. There were 75 OCR errors (accuracy in words greater than 98.5%, in characters greater than 99.7%).

To compare with modern orthography, we used our transliteration utility (A.Colesnicov *et al.*, 2014) that produced itself 31 erroneous words in the modern script (0.6%). We analyzed these words to refine transliteration; as a quick solution, we included these words in the exception dictionary accompanying the transliteration utility.

188 words (3.7%) differ in writing in one word or separately; it's historical development of orthography.

The remaining 319 words (6.3%) are subject to philological expertizing to understand why their writing differs. For example, the same word "up to" was printed both as "până" (modern norm) and "păna". It may be local or historical variation, or misprint. This approach provides reliable result but it is difficult for non-professionals.

Analyzing approaches to post-OCR improvement, we may classify them as manual, semi-automatic, or automatic. Manual correction is slow, expensive, and error-prone, and should be therefore avoided. Nevertheless, we used it in extremity.

Semi-automated tools are represented mostly by different spelling checkers that we'll discuss in the following Sec. 4. There are other semi-automatic tools, for example, PoCoTo.⁷ PoCoTo consists of two parts written in Java, error profiler (server) and PoCoTo client. Error profiler exploits integration with AFR and is language-aware. It needs historical spelling patterns and lexicon of modern and historical word-forms. PoCoTo client is a desktop application that provides, in particular, a very comfortable feature to correct similar OCR errors over the whole text at once.

Automatic tools work using different approaches, driving principles, and language and errors models. Some of them exist for specific languages only but may potentially be of use.

⁷ <https://github.com/cisocrgroup/PoCoTo>

Take as the first example a pattern driven automatic corrector that use error and language models expressed as regular expressions. Supposing “c” instead of “e” as an OCR error, the replacement of all “nc” at word end by “ne” may be suitable for most European languages.

We can also mention as another example an n -gram driven corrector. Such corrector is language-aware as it can use different n -gram tables for different languages.

An automatic corrector using statistical approach was described by X.Tong and D.A.Evans.⁸

An automatic post-OCR corrector AfterScan⁹ applies to text over 5,000 correction rules and over sixty analysis algorithms. It works for English and Russian texts including historical orthography. There are several products of the kind for different languages.

An auxiliary post-OCR process is extraction from the recognized text information that will refine further OCR. Usually, we prepared list of words from the text, and iteratively repeated OCR with this addition to AFR word list. AFR uses language-aware word list to resolve ambiguities and to delete hyphenation. Therefore, larger and actualized word list can imply better OCR. We prepared word lists for AFR using Notepad++ with TextFX plugin.

AFR needs sorted word list containing one word in UTF-8 per line. To get it, we save the OCR result from MS Word to text in UTF-8, and open this file in Notepad++.

With regular expressions search of Notepad++, we merge all lines ending with a hyphen with their next lines. Then we replace all spaces (and, may be, points, commas, etc.) to end of line combination. The result is sorted uniquely, and we get the necessary word list. To use it, we should check and correct it. With old Romanian, we transliterated the list in the modern Latin script, and compared the result with an available word list of the modern Romanian. Most of words become confirmed. Then we confirm more words using previously obtained words of old Romanian. The remaining words should be analyzed and corrected manually with further expansion of the list. This process is complicated and error-prone, and a need for its programmed implementation in the framework of some software platform is clearly seen.

4. *Spell checking after OCR*

The main problem is to perform the checking with the appropriate dictionary. We don't possess dictionaries for old Romanian from different centuries. Therefore we should use an existing software product and replace the dictionary, or we should implement an own spell checker (A.Colesnicov, 1995).

We can use commercial software (MS Word) or free Open Source software (for example, LibreOffice Writer).

All spell checkers support user dictionaries, so the procedure seems simple: we mark the text as written in a language that has installed proofing tools for it, and harvest the user dictionary with necessary words. It doesn't violate the software license because the user may introduce in the user dictionary any word in any alphabet.

⁸ <https://pdfs.semanticscholar.org/132b/b9dd928749af370a87ecb939bbb0b05d7aff.pdf>

⁹ <http://afterscan.com/>

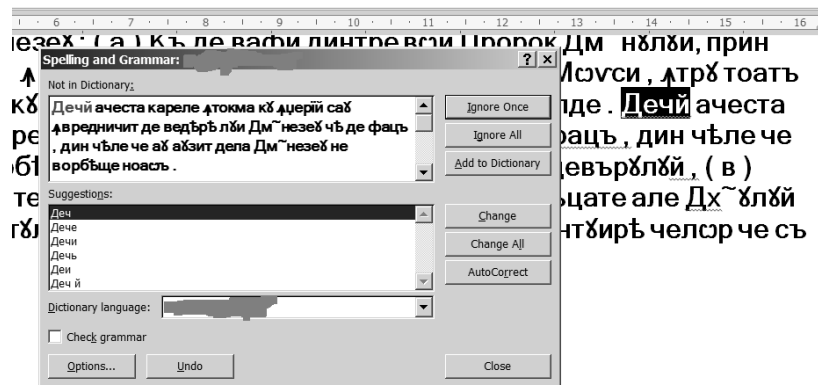


Figure 1: Successful work of MS Word at spellchecking an old Romanian text

In MS Word, this worked partially (Fig. 1). Some old Romanian Cyrillic letters (those like ↑, γ, ω, and the ligature “ia”) are not classified as letters in the spell checker (Fig. 2). The problem may be solved by transliteration of such “no-letters”.

With Libre Office Writer under Linux SuSe, the experiment was successful. We tested the said “no-letters” especially. New word may be introduced directly in the user dictionary, or a file may be copied and configured as such. User dictionaries reside in a dedicated directory. They are simple text files with a standard 4-line header, and contain one word per line. The user can configure one or more files as current user dictionaries.

5. Conclusions

Currently there are no specialized software products that are handy for their user and can improve OCR result, especially for historical texts. The whole technologic chain of old documents digitization is emerging but most stages have no dedicated software products while theoretical solutions are proposed. In practice, we should construct the said technologic chain adapting available general purpose programs to our needs, and adapting ourselves to these programs. This paper demonstrates it starting to reason from a problem of post-OCR text improvement. The need in integrating the whole technological chain in a software platform is also evidentiated.

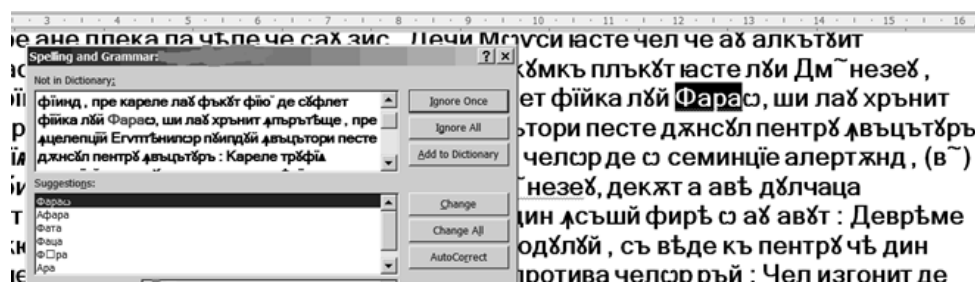


Figure 2: Invalid work of MS Word at spellchecking an old Romanian text

References

- S.Cojocaru, A.Colesnicov, L.Malahov (2017). Digitization of Old Romanian Texts Printed in the Cyrillic Script. *Second International Conference on Digital Access to Textual Cultural Heritage DATECH2017*. Goettingen, June 1-2, 2017, pp.143-148. ISBN: 978-1-4503-5265-9 <http://dl.acm.org/citation.cfm?id=3078093>
- A.Dengel *et al.* (2000) Techniques for Improving OCR Results. Ch. 8 in: H. Bunke, P.S.P. Wang (eds.). *Handbook of Document Recognition and Document Image Analysis*. Singapore, "World Scientific", 2000. ISBN 981-02-2270-X,
- L.Malahov, T.Bumbu, S.Cojocaru, A.Colesnicov, Ș.Ungur (2017). User Interface to Access Old Romanian Documents. *Proceedings of the 4th Conference of Mathematical Society of Moldova, CMSM4'2017*. June 28-July 2, 2017, Chisinau, Republic of Moldova , pp. 479-482, ISBN: 978-9975-71-915-5.
- A.Colesnicov, L.Malahov, T.Bumbu, S.Cojocaru, Ș.Ungur (2017). On Digitization of Romanian Cyrillic Printings of the 17th-18th Centuries. *Computer Science Journal of Moldova*, v.25, n.1 (73), 2017, pp. 217-225.
- Amf.Hotiniul (1795), De obște gheografice, Iași, 1795.
- A.Colesnicov, E.Boian, C.Ciubotaru, S.Cojocaru, L.Malahov (2014). Digitizarea, recunoașterea și conservarea patrimoniului cultural-istoric. In *revista Academos*, Nr. 1(32), 2014, pp. 61-68.
- A.Colesnicov (1995). The Romanian spelling checker ROMSP: the project overview. *Computer Science Journal of Moldova*, vol. 3, nr. 1 (7), 1995. P. 40-54.

ROMANIAN DIACRITICS RESTORATION USING RECURRENT NEURAL NETWORKS

STEFAN RUSETI¹, TEODOR-MIHAI COTET¹, MIHAI DASCALU^{1,2}

¹ *University Politehnica of Bucharest, Computer Science Department, 313 Splaiul
Independentei, 060042, Bucharest, Romania*

² *Academy of Romanian Scientists, 54 Splaiul Independenței, 050094, Bucharest, Romania
stefan.ruseti@cs.pub.ro, teodor_mihai.cotet@stud.acs.upb.ro, mihai.dascalu@cs.pub.ro*

Abstract

Recurrent neural networks have been recently used for most tasks that require the encoding of a text sequence, which makes them suitable for diacritics restoration. We propose a novel model for Romanian language that predicts if a character needs to be replaced by one with diacritics using the surrounding context. Our network models the context with two different encoders, one with character embeddings and the other with pre-trained word embeddings. The model is trained on a corpus of transcriptions from the Romanian Parliament, which contains over 50 million words. Only using character embeddings, the model achieves over 99% accuracy for predicting the correct symbol for the potential diacritics candidates. We show that adding word-level information from the current context further improves accuracy to 99.37% at the character level, and at 98.57% on words.

Key words — diacritics restoration, Recurrent Neural Networks, character and word embeddings.

1. Introduction

Writing texts with Romanian diacritics is in general a cumbersome endeavor as most computer keyboards do not have special keys for specific Romanian letters with diacritics. Hence, people sometimes write digital text in Romanian language with the simplified version of letters with diacritics, namely the ASCII¹ version, thus removing all diacritics. Therefore, diacritics restoration is a mandatory step for adequately processing Romanian texts.

Even a very simple baseline, which substitutes a possible misspelled word with the most frequent candidate, can reach 95.59% accuracy on letters and 93.72% on words. The accuracy on letters is computed by considering all letters which may accept diacritics in the ASCII original version of the text that does not contain diacritics (e.g., letter “t” may accept diacritics, thus becoming “ț”). Similarly, the accuracy on words is computed taking into consideration all words containing at least one letter that may accept diacritics, as described above. Although the accuracy of diacritics restoration systems is quite high, the problem cannot be considered solved. For instance, a page in a novel contains around 400 words. If we consider that 26% of those words contain at least a letter with diacritics (as can be noticed in Table 1 for the Parliament – PAR ° corpus described in the third section) and have an accuracy of 99.00% on words, this will still result in one erroneous word per page. Moreover, although systems can correctly determine the diacritics of most words, there are a few words that tend to be more ambiguous, requiring a deeper understanding of the context.

¹ <http://www.asciitable.com/>

Diacritics change the morphology of a word ('cană' and 'cana' meaning 'cup' and 'the cup', respectively) or even the complete meaning of the word ('fată' and 'față' meaning 'girl' and 'face', respectively); thus, this is a particularly important task for various NLP processes. Without it you do not know the word in case, thus you cannot perform dictionary checks, represent the word in a semantic space, check for stop-words, etc.

The paper continues by describing existing solutions that tackle this task. Next, we present the corpora used for training, the proposed solution in detail, together with performance results of different configurations of our model. The paper concludes with discussions, analyzes the most frequent errors, and considers alternatives for improving the model.

2. Related work

Most previous methods which were experimented for Romanian restoration of diacritics do not use neural networks. Among those that do, there are no solutions specifically optimized for this particular language (i.e., they were generally designed to work on many different languages).

Using a knowledge-based approach using *part-of-speech* tagging for choosing the correct word in ambiguous cases is described by Tufiş & Ceaşu (2008), reporting an 97.75% accuracy on words and 99.40% on characters. Another approach by Ungurean, Burileanu, Popescu, Negrescu, & Dervis (2008) uses sequential filtering with *n-grams* on words and suffixes (last 2-4 characters of each word) for speech synthesis reported an F-measure of 99.34% on characters. As reported in the paper, the use of suffixes was motivated by the fact that they generally capture the morphology of words and incorrect diacritics restorations of ambiguous words are mostly caused by characters at the end of the word.

One of the best results was obtained using an *n-gram* (Brown, Desouza, Mercer, Pietra, & Lai, 1992) language model, precisely solving ambiguity at word level by replacing each word with a version of its which has the highest probability given the *n* previous, already corrected, words. Creating this model requires computing the probabilities for the *n-grams* throughout the training process. The results reported by Petrică, Cucu, Buzo, & Burileanu (2014) are the highest on this task, namely errors of 0.52% for words and 0.116% on letters. Previous methods using *n-grams* have been tried; nonetheless their accuracy on both words and letters are lower than those previously mentioned. Their method removed unreliable online texts (e.g., news articles) in terms of diacritics orthography, filtering those parts which did not contain a high enough percentage of characters with diacritics.

One problem is that all these experiments were done using a different corpus, which makes difficult the comparison of results. However, studying these methods is important, as they provide relevant insights for improving a model.

3. Method

3.1 Corpus

Two corpora have been considered, both described in Table 1. The first corpus (PAR) contains transcriptions of the parliamentary debates in the Romanian Parliament², from 1996 to 2017. The corpus is very diverse in terms of subjects, as it contains the debates of the parliamentary committees in which economic, social, political and judicial issues are discussed exhaustively. Virtually, all texts in this corpus are valid, in the sense that all sequences of characters are meant to be a transcription of speech, thus grammatically valid sentences. In terms of diacritics some mistakes are expected in this corpus.

We have also considered to use the ROWIKI corpus of all Romanian texts from Wikipedia, but, as can be noticed from Table 1, the proportion of letters and words with diacritics are far lower (4.09% compared to 6.39% for letters in the PAR corpus). Moreover, unlike the previous corpus, this one contains some invalid sequences of characters which are not meant to be correctly formed sentences, but some representation of data (e.g. link references, tables with statistics, etc.). Because of these reasons we have decided not to use this corpus, although it was much larger and diverse than the PAR corpus. In Table 1 letters that accept/contain diacritics refer to all letters for which their ASCII version may accept diacritics (e.g., “a”, “ă”, “î”, “s”, “ș”, etc.).

Table 1: Corpora descriptions

		PAR		ROWIKI	
Letters	Letters with diacritics	15M	6.39%	38M	4.09%
	Letters that accept/contain diacritics	84M	35.28%	296M	31.78%
	All letters	239M	100.00%	933M	100.00%
Words	Words with diacritics	13M	26.37%	33M	16.40%
	Words that accept/contain diacritics	35M	70.41%	118M	57.98%
	All words	50M	100.00%	204M	100.00%
Sentences		2.6M		22.0M	
Unique words		0.21M		2.62M	

Different spelling rules existed during different time periods and the most important replacement consisted of the exchange of character „î” with „â” when it occurs in the middle of a word. This does not influence our model because the two cases produce different inputs for the network (different base letters); thus, no contradictory examples appear.

3.2 Network Architecture

The architecture of the proposed model is presented in Figure 1. The neural network is composed on three different paths: characters, current word, and the current sentence. The

² <https://www.senat.ro/>, <http://www.cdep.ro/>

main idea behind this architecture is to combine lexical with semantic information, therefore capturing more complex contexts.

The proposed neural network predicts the correct diacritic for each letter in the input text, only considering letters that can potentially be substituted with the diacritic version. Romanian language contains five different diacritics: “ă”, “â”, “î”, “ș”, and “ț”. We decided to group them into 4 different classes, based on the sign that is added to the letter: a) No sign, b) “ă”, c) “î” and “â”, and d) “ș” and “ț”.

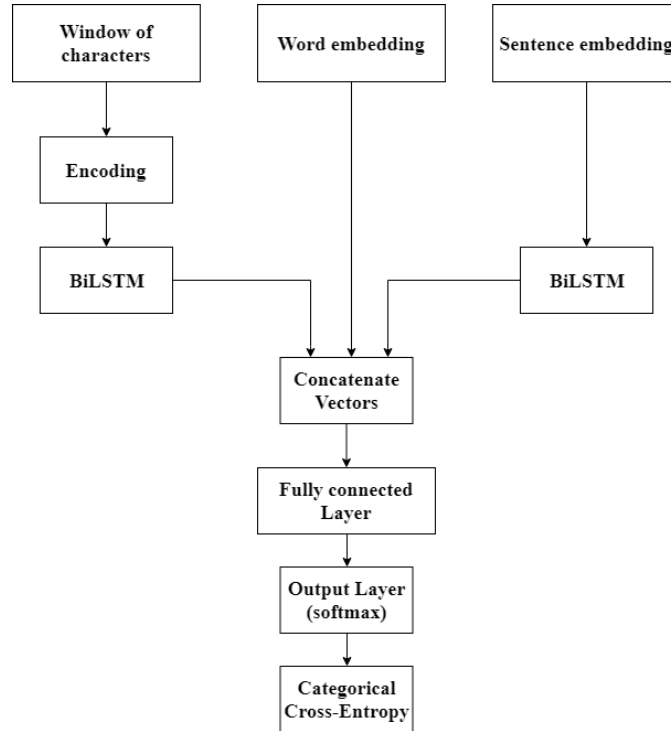


Figure 1: Network design

This grouping of diacritics can take advantage of the similarities between letters. For example, “î” and “â” actually represent the same sound. On the other hand, in the case of “ș” and “ț”, it is not obvious whether they occur in similar contexts. Because of this, experiments using 5 classes, with the two letters being separated, have also been conducted.

The first path in the network is represented by a Bidirectional Long Short-Term Memory (BiLSTM; Graves & Schmidhuber, 2005) encoder using character embeddings. A fixed size window is used to represent the context of the current character. The network can learn different similarities between letters by using character embeddings. Also, applying the encoder at the character level allows the network to generalize different forms of the same word, which will have very similar representations, and also generalize for unseen words, if they look similar to other known concepts. However, the model can benefit from semantic information, as well. In order to achieve this, another path was added to the architecture, represented by a BiLSTM encoder applied on the current sentence. The words in the sentence are represented by pre-trained FastText embeddings, which map words to points in a high-dimensional vector space based on the context in which they occur. The assumption was that

the two encoders are complementary one to another, and the model can learn how to combine the information from both.

The third path is represented by the embedding of the current word. This was added in order to help the network select which part of the encoded context is useful for the current entry. Experiments were also done with models without this word path, which proved to be less accurate.

3.3 Input Processing

The raw text used for input was first cleaned of unusual characters (those that were not in ASCII set were substituted with a fixed character) and all characters were transformed to their lowercase version. The input consists of 3 parts which were differently combined and fed to the neural network.

- The first part is the window of characters, containing a fixed window centered at the character for which the prediction is wanted. The embeddings for characters are learnt, as can be noticed in Figure 1.
- The second part is the embedding of the word which contains the character in question. The word embeddings are pretrained.
- The third part is the embedding of the entire sentence in cause. Each word is embedded independently, thus the sentence is represented as a list of words embeddings. Those word embeddings are identical with the ones from the second part (i.e., each word has the same embedding in both second and third input part).

For the word embeddings, we have used the pre-trained versions from FastText³ which were trained by using the skip-gram model (Bojanowski, Grave, Joulin, & Mikolov, 2016). One advantage of FastText is that it can generate embeddings for words which are not in vocabulary by representing each word as a bag of characters n-grams and associating a vector representation for each n-gram, summing them up thus giving a vector for the word. This feature is very useful for rare words.

Moreover, we have averaged across all words (i.e. words that are in FastText vocabulary) that match the initial word (matching means that stripping the diacritics you get the initial words) because we did not know exactly the word in case without diacritics and thus, neither its corresponding embedding. Consequently, for each letter of the diacritics-free text (i.e. “a”, “e”, “i”, “s”, “t”) which is fed into the network, we create these 3 parts.

All implementation was done in Python. For the machine learning part, we have used TensorFlow⁴, an open source framework which facilitates execution on GPUs. Furthermore, because of the intense computations of the input we have used the Tensorflow Dataset API⁵ which enables parallel usage of multiple CPUs for input preparation, while the GPUs are used for training the model on already prepared input. Keras⁶ was used to implement the neural network we have used, while nltk⁷ was used for text tokenization.

³ <https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md>

⁴ <https://www.tensorflow.org/>

⁵ <https://www.tensorflow.org/guide/datasets>

⁶ <https://keras.io/>

⁷ <https://www.nltk.org/>

4. Results

In order to train and evaluate the model, the PAR corpus was split into three distinct partitions: 60% for training, 20% for validation and 20% for test. The architecture and hyperparameters of the neural network were chosen using the validation partition. Some of the best performing models were then evaluated on the test partition and the results are presented in Table 2.

Some of the hyperparameters of the models were not included in Table 2 because they were not modified during the experiments. The cell size of the word BiLSTM encoder was set to 300, the same as the word embedding size. The character window size was set to 13, while the maximum sentence length was 31. The training was done in batches of 256, using the cross-entropy loss function and the Adam optimizer (Kingma & Ba, 2015).

Table 2: Char/word accuracy

Model	Char Embedding	Char LSTM	Hidden	Epochs	Dev char acc (%)	Test char acc (%)	Test word acc (%)
Chars	16	32	32	5	98.865	98.864	97.413
Chars	20	64	256	5	99.012	99.017	97.750
Chars (5 classes)	16	32	32	5	99.048	99.068	97.867
Chars	24	64	64	4	99.064	99.057	97.856
Chars + sentence	20	64	256	3	99.068	99.065	97.881
Chars + word	20	64	256	4	99.309	99.329	98.453
Chars + word + sentence	20	64	256	5	99.365	99.378	98.573
Chars + word + sentence	20	64	256, 128	5	99.380	99.366	98.553

A detailed analysis of the best models, that includes precision and recall values for each letter, was also performed. In Table 3 we present these results for the All-256-128 model that uses characters, words, and sentences with two hidden layers of sizes 256 and 128.

Table 3: Detailed performance per letter

Model	Letter	Precision (%)	Recall (%)	F-Score (%)
All-256-128	“a”	99.16	98.86	99.01
	“ă”	96.29	97.31	96.80
	“â”	99.17	98.80	98.99
	“i”	99.97	99.96	99.97
	“î”	99.65	99.72	99.69
	“s”	99.84	99.84	99.84
	“ș”	99.44	99.43	99.43
	“t”	99.84	99.77	99.80
	“ț”	98.97	99.29	99.13

5. Discussion

The results presented before show that adding contextual information from the sentence encoded with a BiLSTM can improve the results of a char-based approach significantly. However, only adding the encoded sentence, without the current word is not enough, the accuracy of the char model, and char+sentence being very similar.

Our initial choice of grouping the diacritics into 4 classes turned out to make sense for “â” and “î”, but not for “ș” and “ț”, since a model using only characters and 5 classes achieved an improvement of about 0.2% accuracy compared with the same model with 4 classes. Further experiments should be conducted in order to find the best combination of hyperparameters.

The detailed analysis shows that the model is biased towards choosing the class with no diacritics for each letter. Virtually all measures (precision, recall) are higher for letters with no diacritics (“a”, “i”, “s”, “t”) compared to the corresponding ones with diacritics. This can be explained by the higher number of letters with no diacritics compared to the ones with diacritics, but also by the missing diacritics in the corpus. For instance, the word “și” does not exist without diacritics (i.e., “si”), but the model reports an accuracy on this word of 99.82% and not 100% as expected. Indeed, in the corpus the word “și” is spelled wrongly as “si” a few times, thus confusing the neural network. Other similar words which exist in the Romanian language with just the diacritics version as “in” reported a similar accuracy; therefore, we suspect that missing some diacritics in the training corpus contributes significantly to the final error. Thus, having a perfect training corpus in terms of diacritics will probably increase the final accuracy.

Most of the hardest words to be restored are those in which the diacritics make the word (noun or adjective) indefinite as in “politică”, “importantă” and “prezență”, which mean the model still fails to distinguish between definite and indefinite words.

6. Conclusions

In this paper, we proposed a novel neural network architecture that combines lexical information at the character level, with a semantic representation of the surrounding context computed at the word level, using recurrent neural networks. Our experiments show a significant improvement of the accuracy when adding contextual information.

However, there is room for improvement. Due to time constraints and the low quality of the ROWIKI corpus, we only used the PAR dataset, which is smaller. Better quality data could be obtained using an approach similar to the one proposed by Petrică *et al.* (2014), where the texts were filtered based on diacritics distributions. This way, both datasets could be filtered and used for training, and the model could benefit from the higher number and quality of training examples.

The results show that contextual information helped, but the improvement obtained compared to a neural network that uses the character encoder and the current word is not extremely large. One future improvement could consist in adding an attention mechanism based on the current word, used to better select what is relevant from the context for the current letter. Another observed issue is the imbalance of diacritics in texts (e.g., “a” is much more frequent than “ă” or “â”), which could be solved by using a weighted loss that takes into account these distributions.

Acknowledgments

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 54PCCDI / 2018, INTELLIT – “Prezervarea și valorificarea patrimoniului literar românesc folosind soluții digitale inteligente pentru extragerea și sistematizarea de cunoștințe”.

References

- Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2016). Enriching Word Vectors with Subword Information. *EMNLP* (Vol. 91). <https://doi.org/1511.09249v1>
- Brown, P. F., Desouza, P. V., Mercer, R. L., Pietra, V. J. Della, & Lai, J. C. (1992). Class-based n-gram models of natural language. *Computational Linguistics*, 18(4), 467–479.
- Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM networks. In *Proceedings of the International Joint Conference on Neural Networks* (Vol. 4, pp. 2047–2052). IEEE. <https://doi.org/10.1109/IJCNN.2005.1556215>
- Kingma, D., & Ba, J. (2015). Adam: a method for stochastic optimization. *3rd International Conference for Learning Representations*. <https://doi.org/10.1109/ICCE.2017.7889386>
- Petrică, L., Cucu, H., Buzo, A., & Burileanu, C. (2014). A Robust Diacritics Restoration System Using Unreliable Raw Text Data. In *Spoken Language Technologies for Under-Resourced Languages*, (May), 14–16.
- Tușiș, D., & Ceașu, A. (2008). DIAC+: A Professional Diacritics Recovering System. *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, (May). Retrieved from <http://www.lrec-conf.org/proceedings/lrec2008/>
- Ungurean, C., Burileanu, D., Popescu, V., Negrescu, C., & Dervis, A. (2008). Automatic Diacritic Restoration for a TTS-based E-mail Reader Application. *Bulletin, Series C*, 70, 3–12. Retrieved from http://www.scientificbulletin.upb.ro/rev_docs_arhiva/full24826.pdf

TEPROLIN: AN EXTENSIBLE, ONLINE TEXT PREPROCESSING PLATFORM FOR ROMANIAN

RADU ION

*Research Institute for Artificial Intelligence “Mihai Drăgănescu”
Calea 13 Septembrie nr. 13, Bucharest, 050711
radu@racai.ro*

Abstract

This paper gives an overview of the TEPROLIN text preprocessing platform that we are currently developing within the ReTeRom project. TEPROLIN is designed to be an extensible text preprocessing platform which can add arbitrary annotations to a sequence of tokens. These annotations can be produced by any specialized application, written in any programming language, that implements the TEPROLIN functional interface. The functional interface requires applications to establish a clear separation of resource loading and annotation work and to communicate with the TEPROLIN platform via established and OS-independent Inter-Process Communication (IPC) mechanisms. Using these specifications, TEPROLIN can accept any annotation-producing application and integrate it in the text preprocessing workflow at the appropriate level. To ease its use, TEPROLIN will also be exposed as a REST web service.

Key words — dependency parsing, diacritic restoration, hyphenation, lemmatization, POS tagging, REST web service, text normalization, text preprocessing.

1. Introduction

TEPROLIN is a component-project of the complex ReTeRom (an acronym for “Resources and Technologies for developing Romanian human-computer interfaces”) project that is concerned with the creation¹ of Romanian text processing technologies that can be readily used by the other component-projects of ReTeRom. One such component-project, COBILIRO, aims to develop a Romanian bimodal (speech and text) corpus that is annotated for the benefit of speech recognition and synthesis applications, including e.g. word hyphenation and phonetic transcription. The SINTERO component-project, which develops expressive Romanian speech synthesis, requires further annotations of text prompts of the COBILIRO bimodal corpus, such as dependency parsing and, prospectively, new annotations that will be specified at a later time.

TEPROLIN has to be designed in such a way that a) is easily extensible with plugin-like text preprocessing modules and b) is easily used by all project partners either as a module or online, as a web service (the latter usage scenario will spare the user of having to know about installation details of component applications). In order to satisfy these requirements, TEPROLIN has to:

1. Impose an Application Programming Interface (API) for each of its component applications, that requires the separate handling of *resource loading* (e.g. dictionaries, language models, parsing models, etc.) which is, usually, a time-consuming operation, and

¹ If the required technology is already developed, TEPROLIN will enable its use in the ReTeRom project.

performing annotation work which has to be fast² such that the platform can respond in real time to annotation requests. Since each component text preprocessing application is written in a different programming language, the TEPROLIN API interface has a version in each of these programming languages;

2. Impose a standardized way of communicating with each component application, leveraging IPC mechanisms that are OS-independent and which ensure *fast communication*. We can satisfy the OS independence by implementing TEPROLIN in one of the popular, interpreted programming languages such as Python 3 whose modules will abstract away the IPC working details;
3. Maintain a schema of the possible text preprocessing workflows that can be constructed with the component applications and output the results in a structured format, such as the Text Encoding Initiative³. This format has been agreed by the ReTeRom members;
4. Expose all of its acquired functionality as a REST web service (Fielding, 2000), in addition to implementing it as a Python 3 module.

The TEPROLIN text preprocessing platform will also be featured on the ReTeRom website⁴, alongside the text exploration methods with word embeddings described by Păiș and Tufiș (2018). Specifically, TEPROLIN will supply the text annotations (i.e. lemmatization and POS tagging) that the text exploration interface will then use to extract the relevant word embeddings and construct e.g. word similarity graphs.

In what follows, we will briefly review work done in text preprocessing aggregators (Section 2), we will then present the TEPROLIN architecture and the planned Romanian text preprocessing atomic operations that will be supported (Section 3). We will conclude the paper with Section 4.

2. Related work

2.1. UIMA

UIMA⁵ (Unstructured Information Management Architecture, (Ferrucci and Lally, 2004)) is a platform for Natural Language Processing (NLP) applications written in Java that represents documents as text plus annotations and allows users to define pipelines of *analysis engines* (AEs) that manipulate the document by adding, replacing or removing annotations. Document manipulations are done through *annotators* (i.e. annotation-producing NLP applications).

AEs can be primitive (i.e. containing a single annotator, e.g. POS tagging) or aggregate (i.e. containing a pipeline of annotators, e.g. tokenization and POS tagging) but *both primitive and aggregate AEs implement the same interface* and can be used interchangeably by applications. We have borrowed this design choice from UIMA for our TEPROLIN text preprocessing platform but, as our component annotators are written (or are expected to be written) in

² From the point of view of the platform, i.e. the platform should be designed to add as little as possible to the native computation time of the component applications.

³ Version 3.4.0, online at <http://www.tei-c.org/release/doc/tei-p5-doc/en/Guidelines.pdf>

⁴ <http://dev.racai.ro/ReTeRom/>

⁵ <http://uima.apache.org/>

different programming languages (interpreted or compiled), the common interface has to be implemented for each annotator, in its native programming language.

2.2. GATE

GATE (General Architecture for Text Engineering) (Cunningham, 2002) is another, very well-known text preprocessing and mining platform written in Java. Originally developed at the University of Sheffield in 1995, it “is now used worldwide by a wide community of scientists, companies, teachers and students for many natural language processing tasks, including information extraction in many languages.”⁶

GATE is a fully fledged text preprocessing and mining family of complex applications, featuring an Integrated Development Environment (IDE) in which users can visually compose text processing pipelines or information extraction pipelines for supported languages, a collaborative annotation environment, an embeddable GATE library which can be readily used in any Java application, a highly abstracted and very well defined language processing software composition specification and a process for the creation of robust and maintainable services.

Highlights of GATE’s component model include the Document Format Layer which has “format drivers” that will convert any known format (e.g. PDF, DOCX, PS and HTML) to an internal document format, the Processing Layer which is responsible of producing different annotations of the internal documents and the Language Resources Layer which is responsible of providing access to all other layers to different language resources such as dictionaries or ontologies.

2.3. TextFlows

TextFlows⁷ (Perovšek *et al.*, 2016) is a recent attempt at visually modelling language processing and text mining workflows. Written in Python, it has a nice web interface where the user can compose language processing flows using *widgets*, the TextFlows terminology for UIMA’s annotators. The actual computation is done in the cloud and TextFlows claims that it is easily extendable, offering widgets for well-known packages such as NLTK⁸ and scikit-learn⁹.

The cited paper provides a good overview of the existing NLP text preprocessing platforms, so we invite the reader to supplement our short overview with sub-section 6.2 from Perovšek *et al.* (2016).

2.4. TEPROLIN design notes

All of the NLP text preprocessing platforms that we presented in this section have the following aspects in common:

1. Common interfacing of the component NLP applications (e.g. annotators, processing resources, widgets, etc.)

⁶ Quote from the GATE Wikipedia page: https://en.wikipedia.org/wiki/General_Architecture_for_Text_Engineering

⁷ <http://textflows.org/>

⁸ <https://www.nltk.org/>

⁹ <http://scikit-learn.org/>

2. Common representation of documents within the platform and conversion from different formats to the internal representation;
3. Remote execution (i.e. “in the cloud”) of the actual text processing applications;
4. Visual interfaces for composing language processing workflows.

TEPROLIN is expected to be 1) a fast text preprocessing platform¹⁰ that 2) integrates modules written in different programming languages. From these design specifications, it follows that any native Java or Python existing text preprocessing platforms will incur the penalty of having to adapt TEPROLIN’s applications to the platform. Secondly, the remote execution of the NLP applications is not an option for TEPROLIN as its model (to be detailed in Section 3), requires that all applications run on the same machine such that expensive remote web service calls are avoided. Finally, a visual interface to TEPROLIN was never a design requirement but, if it would become one, the adaptation of TEPROLIN to e.g. TextFlows should be the next item on our to-do list.

In fact, if TEPROLIN would benefit from being included in e.g. UIMA, GATE or TextFlows, its web service design would make that task easier than committing early on to one of these platforms, as all of them favor the “remote execution” model, as mentioned above.

3. *TEPROLIN architecture*

As already mentioned, TEPROLIN is a *fast* text preprocessing platform that allows for the integration of NLP applications written in *any programming language*. To be fast (or as fast as possible, being limited only by the component NLP application computation speed), TEPROLIN requires the following:

1. Each component NLP application has to make sure that its resources (should these be used) are loaded only when the application starts and not with every call to the annotation method. This means that the author of the application has to implement a language-independent interface that has a method named e.g. `loadResources()` which is called when the object representing the NLP application is created.
2. Each component NLP application is a resident process in the memory of the machine hosting TEPROLIN, using native OS process communication channels to communicate with TEPROLIN. We have chosen the *named pipe* IPC method as it is available in both Windows and Linux.

TEPROLIN’s internal document format is CoNLL-X (Buchholz and Marsi, 2006) which is a UTF-8 tabular format of the document, with new annotations being added on new, tab-separated columns. The document can be passed among component NLP applications as a simple string and “parsed” through simple string split operations. TEPROLIN will keep a record of what kind of annotation each columns stores. Table 1 shows the standard CoNLL-X format which will be extended with additional columns by TEPROLIN.

¹⁰ In the sense defined above: adding as little overhead as possible.

Table 1: The tabular CoNLL-X format for an example Romanian parsed sentence

I D	Word	Lemma	CTAG	MSD	Fea ts	Hea d	Dep	PHD	PDEP
1	Scrisul	scris	NSRY	Ncmsry	_	6	nsubj	0	root
2	în	în	S	Spsa		3	case	1	amod
3	sine	sine	PXA	Px3--a- -----s	_	1	amod	6	nsubj
4	era	fi	V3	Vmii3s	_	6	cop	0	root
5	o	un	TSR	Tifsr		6	det	0	root
6	treabă	treabă	NSRN	Ncfsrcn	_	0	root	_	_
7	ușoară	ușor	ASN	Afpfsrcn		6	amod	0	root
8	.	.	PERIOD	PERIOD	_	6	punct	0	root

The columns from Table 1 store the following types of annotations:

1. ID: is the 1-based index of the token in the sentence;
2. Word: is the word form of the token in the sentence;
3. Lemma: is the lemma of the token with the word form from the WORD column;
4. CTAG: is the disambiguated, corpus POS tag (Tufiş, 2000) of the word form from the WORD column;
5. MSD¹¹: is the disambiguated, morphosyntactic descriptor POS tag of the word form from the WORD column;
6. Feats: stores supplementary syntactic and/or semantic features of the corresponding token (e.g. the attribute-value pairs descriptions of the MSD); if these are missing, the “default” value is used which, in CoNLL-X, is the ‘_’ character;
7. Head: is the index of the head of the token in the second (WORD) column in the dependency parsing of the sentence;
8. Dep: is the name of the (Romanian) dependency relation (Barbu Mititelu *et al.*, 2016) that holds between columns HEAD and ID;
9. PHD (“parent head”): it is the ID of the head of the token whose index stored in the HEAD column;
10. PDEP (“parent dependency relation”): it is the name of the dependency relation that holds between the indexes stored the PHD column and the HEAD column.

The ReTeRom project will enable TEPROLIN to support the following text preprocessing “atomic” operations¹²:

1. **Romanian text normalization:** this operation replaces all of the non-standard Romanian diacritics with their standard form (e.g. replace ‘ș’ with ‘ş’ and ‘ț’ with ‘ţ’), ensures that the

¹¹ See <http://nl.ijs.si/ME/V4/msd/html/msd-ro.html> for details.

¹² After the ReTeRom project ends, TEPROLIN can be further extended using the infrastructure that is already created.

compound word/clitic splitting dash is the ‘-’ character and also replaces all of the Unicode versions of whitespace with ‘ ’;

2. **Romanian diacritic restoration:** this operation automatically inserts missing Romanian diacritics in words. For instance, the Romanian word “fata” means “the girl” but the word “fată” means “a girl”;

3. **Word hyphenation:** this operation automatically splits Romanian word forms into their component syllables. For instance, the Romanian word “camion” (English “truck”) is split into “ca-mi-on”;

4. **Word stress identification:** after the word is split into the component syllables, this operation will identify the stressed syllable. In our previous example, the stressed syllable of “camion” is the last one: “ca-mi-on”;

5. **Phonetic transcription:** this operation automatically “translates” the word form from its written form to its phonetical representation. For instance, for “doagă”, the phonetic transcription is [döǎ-gă] in which the stress is underlined and “ö” is the symbol for the semi-vowel “o”;

6. **Numeral rewriting:** this operation will take any digit-based numeral and will output its descriptive form. For instance, “124” will be rewritten as ”o sută douăzeci și patru” (English “one hundred twenty four”);

7. **Abbreviation rewriting:** similar to numeral rewriting, this operation will take known abbreviations and will output their expanded form. For instance, “DNA” or “D.N.A.” will be expanded into “Direcția Națională Anticorupție” (English “National Anticorruption Directorate”), in the contexts in which this abbreviation has this meaning;

8. **Sentence splitting:** this operation will break paragraphs into sentences;

9. **Tokenization:** this operation will break sentences into their constituent word/punctuation tokens;

10. **POS tagging:** this operation will assign a POS tag to each token in the sentence. We will use the MSD POS tag set, introduced in Table 1;

11. **Lemmatization:** this operation will assign each word its appropriate lemma, given the disambiguated MSD;

12. **Chunking (shallow parsing):** this operation will recognize non-recursive noun phrases (e.g. “o floare albastră”, “mașina de cusut”), verb groups (e.g. “au fost făcute”, “se vor fi dus”), prepositional phrases (e.g. “peste dealuri și văi”), adjectival phrases (e.g. “foarte frumoasă”) and adverbial phrases (e.g. “destul de repede”);

13. **Dependency parsing:** this operation will automatically construct the dependency relation tree of the given sentences, using the Romanian UD relations already introduced above, in Table 1.

4. Conclusions

The TEPROLIN text preprocessing platform is under development and, when ready, it will be available as a Python 3 class as well as a REST web service. We use existing NLP applications, provided by the ReTeRom partners or available at our institute.

TEPROLIN can be easily extended with new NLP component applications and, if the ReTeRom project partners will see the usefulness of the enterprise, we can easily adapt TEPROLIN to work inside a major language processing and text mining platform such as UIMA, GATE or TextFlows.

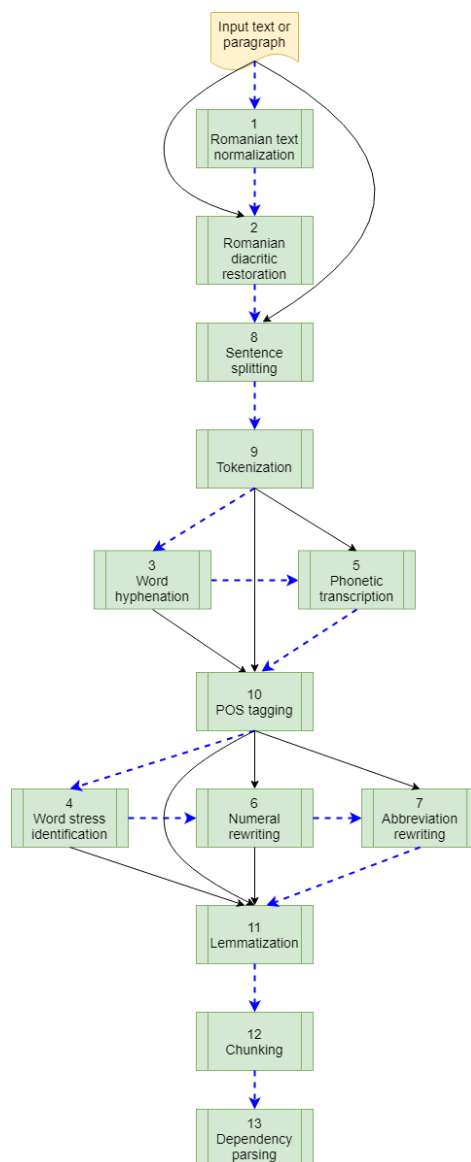


Figure 1: The call dependency graph of the TEPROLIN platform. The dotted line represents the composition chain of the atomic operations that produces all possible annotations.

Acknowledgements

This work was supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73PCCDI, within PNCDI III.

References

Barbu Mititelu, V., Ion, R., Simionescu, R., Irimia, E. and Perez, C.-A. (2016). The Romanian Treebank Annotated According to Universal Dependencies. In *Proceedings of HrTAL2016*, September 29-October 1, 2016, Dubrovnik, Croatia.

- Buchholz, S. and Marsi, E. (2006). CoNLL-X shared task on Multilingual Dependency Parsing. In *Proceedings of the 10th Conference on Computational Natural Language Learning (CoNLL-X)*, June 2006, New York City, 149-164. © 2006 Association for Computational Linguistics.
- Cunningham, H. (2002). GATE, a General Architecture for Text Engineering. *Computers and the Humanities*, 36:2, 223-254.
- Ferrucci, D. and Lally, A. (2004) UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment. *Natural Language Engineering*, 10:3-4, September 2004.
- Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures, *Ph.D. dissertation*, University of California, Irvine.
- Păiș, V. and Tufiș, D. (2018). More Romanian Word Embeddings from the ReTeRom Project. Submitted to *Proceedings of International Conference on Linguistic Resources and Tools for Processing Romanian Language (ConsILR) 2018*, 22-23 November, 2018, Iași, Romania.
- Perovšek, M., Kranjc, J., Erjavec, T., Cestnik, B. and Lavrač, N. (2016). TextFlows: A visual programming platform for text mining and natural language processing. *Science of Computer Programming*, 121, 128-152 (Online at http://kt.ijs.si/nada_lavrac/Publications/2016-TextFlows.pdf).
- Tufiș, D. (2000). Using a Large Set of EAGLES-compliant Morpho-Syntactic Descriptors as a Tagset for Probabilistic Tagging. In *Proceedings of LREC2000*, Athens, 2000, 1105-1112.

HOW TO FIND OUT WHETHER THE ROMANIAN LANGUAGE WAS INFLUENCED BY THE TWO HISTORICAL UNIONS?

DAN CRISTEA^{1,2}, DANIELA GÎFU^{1,2}, SVETLANA COJOCARU³, ALEXANDRU COLESNICOV³, LUDMILA MALAHOV³, MARIUS POPESCU⁴, MIHAELA ONOFREI², CECILIA BOLEA²

¹ “Alexandru Ioan Cuza” University of Iasi, Faculty of Computer Science, 16 Berthelot St., Iasi, Romania

² Institute of Computer Science, Romanian Academy, Iasi Branch, 2 Codrescu St, Iasi, Romania

³ Institute of Mathematics and Computer Science “Vladimir Andrunachievici”, 5, Academiei St., Chişinău, Republic of Moldavia

⁴ Faculty of Mathematics and Computer Science, University of Bucharest, 14 Academiei St., Bucharest, Romania

{dcristea, daniela.gifu}@info.uaic.ro, {svetlana.cojocaru, kae, mal}@math.md, marius.popescu@fmi.unibuc.ro, {mihaela.onofrei, cecilia.bolea}@iit.academiaromana-is.ro

Abstract

We propose a research methodology intended to prove empirically, using quantitative methods, whether the modern Romanian language was influenced by the two historical union events (1859 and 1918). The changes in language suited for being approached by computational means address: lexicon, morphology, grammatical structure and semantics. Out of these manifestations of language use, we will concentrate in this preparatory study only on the lexicon and, partly, on the semantics. The study is restricted to the written language.

Key words — Romanian language, corpus linguistics, diachronic corpora, similarity.

1. Introduction

It is well known that Romanian people living in the historical provinces, although separated for long periods of time, have conserved a unified language, with rather minor variations in the lexicon, the grammatical structure or the accent. Obviously, the population, speaking rather the same language, facilitated the historical acts of union, because people had the sense of belonging to the same social community. This year celebration of a centenary since the historical act of Great Union was a catalyst for many studies on this matter. But an inverse influence might equally have been manifested, namely the unions themselves should have had an influence on the evolution of the Romanian language. Each union allowed easier circulation of a common press, the administration was the same, and the school functioned under a unique ministry, therefore it must have adopted common language teaching rules.

In this paper we propose a research methodology intended to prove empirically, using quantitative methods, whether the modern Romanian language has been influenced by the two historical union events (1859 and 1918). The changes in language suited for being approached by computational means could address: the lexicon (inventory of lemmas), the morphology

(inflectional forms of different part of speeches), the grammatical structure (peculiar syntactic roles that words or expressions could fulfil in language use) and the semantics (senses of words). Out of these manifestations of language use, we will concentrate in this preparatory study only on the lexicon and the semantics, while other aspects were being only vaguely touched. The study is restricted to the written language, since the spoken language cannot be properly sourced.

The second goal of this paper is to initiate a diachronic corpus of the Romanian language (Gifu, 2015), to discuss theoretical means of investigating the distance between different variants of Romanian (Dascalu & Gifu, 2015) in the time proximity of historical union events (this way departing from scholars stating that “no scalar measure can be developed for linguistic distance” (Chiswick and Miller, 2004)), and to support such an enterprise by foreseeing a technology able to deal with the whole way from scanned versions of prints (in both the Cyrillic and the Latin alphabet) up to a corpus containing XML-annotated transcribed texts together with their metadata. In an optimistic perspective, this enterprise will pave the way towards the know-how and a technology able to treat prints and manuscripts in old Cyrillic Romanian. With such an acquisition we will be much closer to a true digitisation of old Romanian writings, meaning by this much more than the mere collecting of scans that could only be turned over page by page, as is the current procedure with most of the old language archives of the languages in Europe or elsewhere (EUROPEANA¹, ITU Library and Archives Service², etc.). Such a perspective would finally allow online access of language researchers into the contents of old prints and manuscripts, which, indexed on multiple criteria, would make possible much better informed forms of research (categorisation of texts, advanced content-based search, comparative studies of syntax and semantics, diachronic and diatopic, diastratic, diaphasic studies on the Romanian language, construction of multilingual parallel corpora, multilingual alignment of Psalters’ verses for Biblical studies, etc.).

2. Methodology of study

The major ingredient of this study should be a representative corpus of extracts of language used in the public area, digitised, brought to an uniform set of UTF-8 (16) characters (in which, for instance, any differences in coding the diachytics would disappear), cleaned of any formatting clues and associated with proper metadata.

The methodology of this study goes through a sequence of operations, starting with the paper sources and ending with statistics that exploit the annotated corpus:

1. establish a balanced (over time and territory) collection of source documents;
2. acquire metadata of primary documents;
3. scan/photocopy the collection of primary documents;
4. deskew and segment the pages down to significant rectangular zones containing grouped chunks of text (titles, columns, lines, cuff notes, interlinear characters, etc.);
5. linearize the 2D display of the chunks in an XML notation;

¹ www.europeana.eu

² <https://www.itu.int/en/history/Documents/AboutDigitization.pdf>

HOW TO FIND OUT WHETHER THE ROMANIAN LANGUAGE WAS INFLUENCED BY THE TWO HISTORICAL UNIONS?

6. using only a manageable subcollection, interpret the text zones in UTF-8 (16) by an initially trained optical character recognition (OCR) program;
7. repeat in a bootstrapping manner a sequence of correct-retrain-enlarge-OCR operations until the whole collection is correctly interpreted in the original alphabet and the evaluation of the trained OCR shows a flattening to a maximal value;
8. if the alphabet of the original Romanian text is Cyrillic, transliterate the text in context (operation called by linguists interpretative transcription), on a manageable subcollection, the original Cyrillic into the Latin alphabet, by applying an initial set of transliteration norms;
9. repeat in a bootstrapping manner a sequence of correct-retrain-enlarge-transcribe operations until the whole document is correctly transcribed and the evaluation of the trained interpretable transcribers shows a flattening to a maximal value;
10. submit to linguists a subcollection of this collection, uniformly covering the periods and territories under study, for a manual annotation process to tokens, part-of-speech (POS) tags and lemmas;
11. iterating a bootstrapping correct-retrain-enlarge-annotate procedure obtain the annotated form of the collection, tokenized, POS-tagged and lemmatized, until the evaluation of the trained tokenizer+POS-tagger+lemmatizer shows a flattening to a maximal value;
12. once all texts covering the intended period and space are processed this way, organize and index the respective collection; one may keep together all levels of information in the original XML segmentation of pages (from coordinates of corners of textual zones to their Cyrillic and Latin transcription and the corresponding POS+lemmas of words) or place in separate files each specific level; one way or the other, links that would make explicit the connection of the decrypted and annotated text into the original image should always be maintained;
13. extract word embeddings out of the annotated collection;
14. for each act of union, use pre- and post-union data to compute measures and draw graphics;
15. compare and interpret the results (better, with the eye of a philologist).

In the following we will discuss some of these aspects.

2.1. Establishing sources

Our intention is to include in this corpus printed publications covering a period of 30 years before and after each historical union event. The time limits of this corpus would therefore extend to $1859 - 30 = 1829$ up to $1859 + 30 = 1889$, for the first union event, and $1918 - 30 = 1888$, up to $1918 + 26 = 1944$, for the second event. These yields the continuous period 1829 - 1944 from which our data should be collected (see Figure 1).

³ By coincidence, the year that is considered (Andriescu, 1979) the beginning of the Romanian press, when the two press “veterans” were born: *Albina românească* (Iași, ed. Gh. Asachi) and *Curierul românesc* (Bucharest, ed. I.E. Rădulescu).

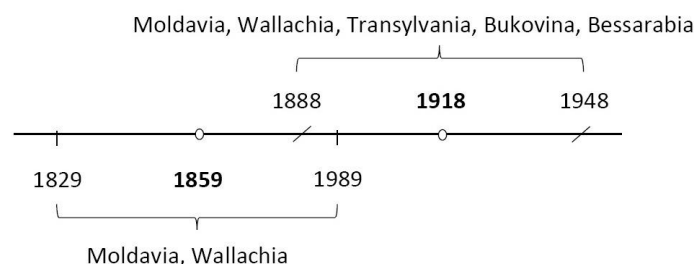


Figure 1: Historical periods around the two union events and the respective provinces involved

To keep the study focused, it is clear that the first 60 years or so of the corpus should collect writings from Moldavia and Wallachia, while for the last approximately 60 years it should contain publications covering all Romanian provinces, including the ex-soviet Moldavia (Bessarabia), which was part of the Big Romania until 1940 and then again between 1941-1944.

2.2. Segmentation and linearization

The zones of printed text on the original scan of a page should be restored for skewing deformations and then segmented at columns, lines, interline characters, markings placed on the cuff of the pages, accents and diacritics, drawings (including capital letters at the beginning of lines), artefacts, etc. The output of this task should be a technology able to produce an XML description of the (rectangular) zones of text that are evidenced on a page, after eliminating the noise, drawings and artefacts. This description will linearize, in an XML specially designed scheme, the zones of text from a 2D page layout (see Figure 2). It should include spatial information about zones/pieces of text in the original image together with their contents. This will contain all links necessary to identify different text spans on the original image. The structure and content of this XML linearization will grow with each of the processing steps applied upon the document. For instance, the Cyrillic decoding of a line of text (after OCR) and its Latin transcription (after transliteration) will be part of this description.

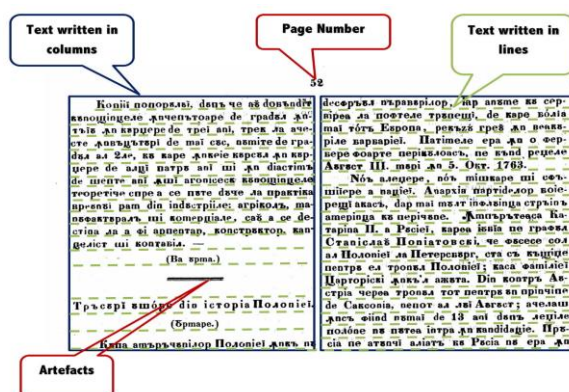


Figure 2: A page from *Foaie pentru minte, inima si literatura*, Braşov, 1847

2.3. OCR and correction

In this section we present our experience of using ABBYY Finereader⁴ (AFR) for historical Romanian texts optical recognition. Certainly, AFR is one of the leading instruments of OCR, especially in industry, services, libraries, newspapers, being used in large-scale OCR actions. It performs good preprocessing, especially in analysis of the document structure. The recommended epoch of documents is the 19th century and later. Single glyphs can be trained, but for complete typesets, it is possible to block default recognition patterns and to train glyphs one by one. AFR is proprietary commercial closed software that can't be modified by users. It is language specific, a "language" consisting of a target alphabet, a word list (lexicon) and a set of recognition patterns. Patterns may be uploaded into AFR, expanded by further training, and finally downloaded in order to be kept. The lexicon is used during recognition to fix ambiguities and to get rid of hyphenation.

The recognition patterns are determined during training and organised in sets of glyphs for specific characters. However, different editions of documents written in the Cyrillic Romanian script impose retraining of the AFR.

The OCR technological process consists of several steps (Malahov *et al.*, 2017). We have used Scan Tailor (in its batch mode) to provide images of the quality suitable for AFR. The program can delete specks, rotate pages, convert images into black-and-white, correct trapezoidal distortions, etc. However, most of AFR corrections should be performed manually, one by one. Moreover, the page structure cannot be restored automatically, especially for historical books that contain many page decorations. A specific flaw arose with some old texts in Romanian Cyrillic that have many overline signs and interline letters. Sometimes AFR splits one line containing overline letters and signs in two. A roundabout solution is to formally increase the image density. This does not imply re-scanning but is equivalent to decreasing the calculated image size. By default, AFR stores the results of OCR as MS Word files.

One of the actions following OCR is replenishing word lists. To work with word lists we have chosen Notepad++, a text editor that can process Unicode and provides search and replace using regular expressions. TextFX extension permits to sort word lists.

Several glyphs belonging to Cyrillic Romanian were not covered by Unicode until the introduction of Cyrillic Extended-B block [A640-A69F] in 2009. Since then, the following fonts appeared that cover this Unicode block. Unifont is a bitmap font and can be used only for preview. The single monospace font for text editors is Everson Mono but it looks ugly. Several other fonts are Kliment STD, Monomakh Unicode TT, Ponomar Unicode TT, Menaion Unicode TT, Segoe UI, and Segoe WP. Calibri and Microsoft Sans Serif cover Cyrillic Extended-B Unicode block only starting with Windows 10. Virtual on-screen keyboard can be used to enter these glyphs, and BabelMap to visualize Unicode fonts.

Most often, OCR software introduces errors. Manual correction is possible even in AFR before storing the OCR result, but exporting it in a more comfortable environment, will ease the correction operations. Improving the OCR result can be done in two ways: by comparing the output against a word list and by using automatic correction rules. Word list creation is a

⁴ <https://finereader.en.softonic.com/>

complicated problem because electronic dictionaries of old word forms do not exist. Possible sources are: scientific editions of old texts with manual conversion in the modern script, or modern dictionaries transliterated to the old Cyrillic Romanian script.

Iterative OCR correction using the recognized text itself could also be an option. Each word can be isolated on a separate line, hyphenations deleted, and the result sorted after deleting repeated words. The resulting word list can be checked manually, and then uploaded into AFR to repeat OCR with supposedly better quality. In such a manner, the word list of the corresponding epoch can be expanded with each new text.

Automatic corrections rules can be deduced by comparing the OCR output (Test) against the same span of text manually revised by experts (Gold). These revisions should be made in close resemblance with the original text and not with the modern language. For instance, the original can miss vocals in writing the name of God (*Dmnul* instead of *Domnul*) and, very often, cratima is not placed between a clitic and an auxiliary (*iau dat* instead of *i-au dat*) or between a negation and a clitic (*nul caută* instead of *nu-l caută*).

2.4. Transliteration and correction

The most needed post-OCR processing is transliteration of the text into the Latin script (Colesnicov *et al.*, 2014). This can be done in correspondence with more scripts covering the period 1812-1989: the Cyrillic Romanian script (CR), the Transitional Romanian script (TR), and the (Soviet) Moldavian Cyrillic script (MC). We will also refer to the Modern Romanian Latin script (since 1904) as to MRL.

There are regular dependences that support conversions $CR \rightarrow MRL$ and $TR \rightarrow MRL$. Most letters are mapped one-to-one, for example, $\phi \rightarrow f$, but several mappings are context dependent. Examples are: $r \rightarrow gh$ (before **e, u, i**); $r \rightarrow g$ (otherwise); $\uparrow \rightarrow \hat{i}$ (before **m, n, m, n**); $\uparrow \rightarrow \hat{im}$ (before **o, u, b, p**); $\uparrow \rightarrow \hat{in}$ (otherwise). In some texts, the letter $\uparrow$ (**in**) is used differently.

These rules are valid both for CR and for TR, because only the font shape differs (Uncial or Antiqua) but Unicode codes are the same. Transliteration $MC \rightarrow MRL$ is more difficult because MC was developed following a political decision without any scientific background. It is an irregular representation of Romanian sounds with Russian letters. One-to-one mapping of most letters and context driven rules exist; examples are previous rules for ϕ and r .

Colesnicov *et al.* (2014) report maximum one error per page, which approximates to 0.15% of erroneous words. In a juridical text, for instance, all errors were of the type $\mathbf{\text{я}} \rightarrow \mathbf{\text{ia}}$ (instead of the correct **ea**), e.g., **примяскэ** $\rightarrow$ (**primiască**) $\rightarrow$ **primească**. One possible solution to this would be to generate both forms of the target word and let a dictionary decide. In our opinion, the issue of transliteration should not hide language differences. For instance, the following forms should be maintained: **Loran** instead of the modern **Laurent** (a foreign proper noun), from **Лоран**; **dizain** instead of **design**, from **дизайн**; **câne** instead of **câine**, from **кыне**; **eşire** instead of **ieşire**, from **ешире**.

2.5. What and how to compare

We will suggest in this section a distance measure between two corpora, which, as we will see, could be easily generalized, and out of which the question put in the title to be answered.

HOW TO FIND OUT WHETHER THE ROMANIAN LANGUAGE WAS INFLUENCED BY THE TWO HISTORICAL UNIONS?

Lexical distance. If we note by: l_0 the lexis of the corpus c_0 and l_1 the lexis of the corpus c_1 , then we can write: $l_1 = l_0 + p - n$, where: p (the positive lexis) is the lexis appearing in l_1 and not appearing in l_0 ; n (the negative lexis) is the lexis appearing in l_0 and not appearing in l_1 . The $+/-$ operations are reunions and differences of sets (Figure 3).

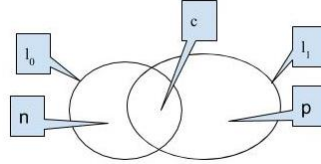


Figure 3: Comparing the lexis of two corpora

The lexical distance (noted here $\setminus_L$) between corpora c_1 and c_0 should therefore be: $l_1 \setminus_L l_0 = p + n$. A normalised distance could be obtained by: $D_{1/0} = (|p| + |n|) / (|p| + |c| + |n|)$, where c is the common corpus (intersection) and $|x|$ is the dimension of the lexis x . This way, the distance is 0 if both p and n are null, and it is 1 if the common corpus c is null.

It can be easily proved that this function is indeed a distance, i.e. it has the properties:

- a). $D(x,y) \geq 0$;
- a') $D(I,y) = 0 \Rightarrow x=y$;
- b). $D(x,y) = D(y,x)$;
- c). $D(x,z) \leq D(x,y) + D(y,z)$, for any x, y, z .

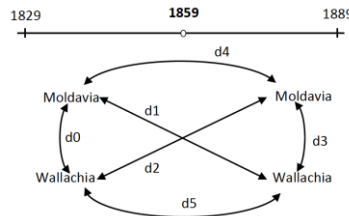


Figure 4: Distances between written languages in provinces Moldavia and Wallachia, in the temporal vicinities of a historical event

Using the notations in Figure 4, the most obvious inequality that should be proven is $d_3 < d_0$, i.e. after a union event it is normal to suppose that the distance between languages in the provinces decreased. If, however, an inequality such as $d_4 < d_5$ is shown, it signifies that the language of Moldavia (M) is more stable than that of Wallachia (W). On the contrary, if $d_5 < d_4$, the interpretation is that, in evolution, the language of W is more stable than the one in M. An inequality of this kind would place the province showing the minimum change at the core of the evolution of Romanian language. A comparison can be made with Modern Italian, which coalesced on the core of the Tuscan dialect (Dante's language).

Other interesting inequalities:

$d_3 < d_1 \Rightarrow$ after union the language in W became more closed to the one in M than it was before the union;

$d_0 > d_2 \Rightarrow$ the language in W was more distant to the one in M before the union than after it;

$d_3 < d_2 \Rightarrow$ after union the language in M became more closed to the one in W than it was before the union;

$d_0 > d_1 \Rightarrow$ the language in M was more distant to the one in W before the union than after it.

Variations of this distance can be suggested by placing frequency filters in the formation of sets l_0 and l_1 . For example, l_i^2 would be sets consisting only of frequency lemma equal to or higher than 2 (no hapax legomena). In notations of this kind, we would have: $D^1, D^2 \dots$ distances between two languages on increasing frequency criteria. We expect that differences between distances will flatten the more frequency filters are applied, because fewer words are taken into account, ie: $D^{k+1} - D^k > D^{m+1} - D^m$, for $m > k$.

2.6. Semantics (senses of words)

Word embeddings show promise as a diachronic tool. In (Hamilton *et al.*, 2016) a robust methodology was developed for quantifying semantic change by evaluating word embeddings against known historical changes (events). We will follow a similar approach:

- Separate word embeddings, for each Romanian province and each historical period (see Figure 1), should be created, by using state of the art techniques, like word2vec (Mikolov *et al.*, 2013) and GloVe (Pennington *et al.*, 2014).
- Having these word embeddings, the semantic distance between written languages in different provinces in the temporal vicinities of a historical event can be measured, and a similar analysis as in the case of the lexicon (see Figure 5) can be performed.

In order to compare word vectors from different embeddings we must ensure that the vectors are aligned to the same coordinate axes because low-dimensional embeddings will not be naturally aligned due to the stochastic nature of word2vec and GloVe. In particular, both these methods may result in arbitrary orthogonal transformations, which do not affect pairwise cosine similarities between words from the same embedding but will preclude comparison of the same word across different embeddings.

Let define $W^{(t,p)} \in \mathbb{R}^{d \times |V|}$ (d being the dimension of embedding and $|V|$ the dimension of vocabulary) as the matrix of word embeddings learned from the corpus of a period t and a province p . We will align two embeddings with the same vocabulary V , while preserving cosine similarities, by optimizing:

$$R = \underset{Q^T Q = I}{\operatorname{argmin}} \|QW^{(t1,p1)} - QW^{(t2,p2)}\|_F$$

with $R \in \mathbb{R}^{d \times d}$. The solution corresponds to the best rotational alignment and the minimum itself $\|RW^{(t1,p1)} - RW^{(t2,p2)}\|_F$ can be viewed as a measure of the semantic distance between the language from $(t1,p1)$ and the language from $(t2,p2)$. This distance can be used to pursue a similar analysis as in the case of the lexicon.

The aligned embeddings can also be used for a more refined analysis at the word level. For a particular word w , the cosine similarities between the two aligned embeddings of w can be used as the semantic similarity of the word w in the language from $(t1,p1)$ and the word w in the language of from $(t2,p2)$. Looking at the nearest neighbours of w in the two aligned embeddings can give an indication about how w was used in $(t1,p1)$ and in $(t2,p1)$. As an example of how this can be visualized see Figure 5, but, apart of time, a province dimension will be added.

HOW TO FIND OUT WHETHER THE ROMANIAN LANGUAGE WAS INFLUENCED BY THE TWO HISTORICAL UNIONS?

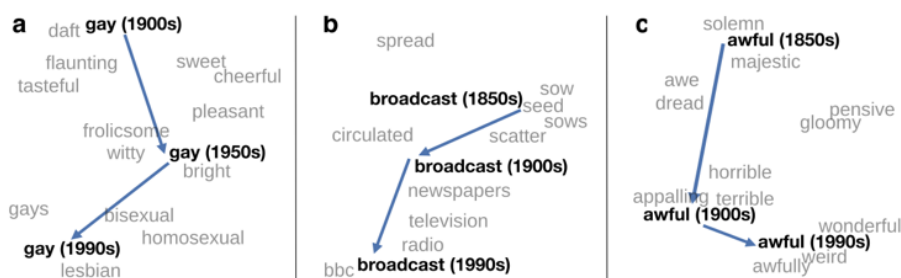


Figure 5: Visualizing changes in word meaning (from (Hamilton *et al.*, 2016))

2.7. Other distances

Etymological distances. If we apply etymological filters to the lexicon of a corpus, we can highlight l^{LAT}_i , l^{SL}_i for the Latin lexis, Slavic lexis etc. of a corpus i and, hence, the corresponding distances: D^{LAT}_{ij} , D^{SL}_{ij} , etc. (see Section 2.5). It would be interesting to reveal and interpret increases or decreases of these distances in time.

eDTLR (Cristea *et al.*, 2011), with its chronological sources of citations and etymology marked, could be a good resource for applying etymological filters.

Morphological distances. Measuring morphological distances between variants of one language means counting differences in word formation (derivation and compounding) and in flexional rules (paradigms, as in conjugation of verbs and declension of nouns and adjectives), for common lexicals. Examples of this kind are: number of modified paradigms, number of differences in endings, migration of words from one paradigm to another, etc.

Syntactic distances. Syntactic distances between languages have been intensively studied in Contrastive Linguistics and Foreign Language Learning. Of interest to our study would be to compare differences in syntactic structures between variants of language, as for instance modification of frequencies of occurrence of certain syntactic roles. In order to do that, the corpus should be annotated by syntactic parsers trained on different regions and periods of time.

3. Discussions and conclusions

Languages have been studied in comparison for a long time and there is a vast literature on the issue, which, for lack of space, we will not browse in this study. One point of comparison is their historical roots: “The main metaphor that is used to explain the historical relationship is that of the *language family* or *family tree*.” (Crystal 1987, p. 292). The main difficulty that has fuelled the pessimistic vision regarding the computation of a linguistic distance (mentioned in the Introduction) comes from “the myriad characteristics that makes up the structure of languages” (Chiswick and Miller, 2004). In this paper we have tried to overpass this barrier in two directions, the lexicon and the semantic content of words and to suggest other possible distances, with the goal of fulfilling the vivid curiosity of how much the two unions in the recent history of Romania have influenced the formation of the modern language.

We are aware of the dangers to misinterpret the results of such a study. Wrong conclusions could come from more directions, but the most pregnant ones are the insufficiency of digitised resources and the unbalanced quantity of textual data in different periods and provinces. Moreover, OCR and lemmatization errors could also induce wrong arguments. The process of manual correction of a corpus of the required dimension being extremely costly, a study of this kind will certainly be feasible in synchronicity with the acquisition of a diachronic corpus of old Romanian language.

The methodology of study for comparing variants of language discussed in this study could suggest more sophisticated solutions applicable to collections of spoken Romanian language and covering modern and contemporary periods. In an enterprise of this kind, the Portal of spoken Romanian and the speech processing technologies that make the core of the ReTeRom project (acknowledged below) will certainly be of great help.

Acknowledgements

This study was realised as part of the research theme of the Natural Language Processing group from the Institute of Computer Science of the Iași branch of the Romanian Academy and was partially supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818 / 73PCCDI (ReTeRom), within PNCDI III and by the README project "Interactive and Innovative application for evaluating the readability of texts in Romanian Language and for improving users' writing styles", contract no. 114/15.09.2017, MySMIS 2014 code 119286. We are grateful to the libraries of the Romanian Academy in Bucharest and Iași and the Central University Library in Cluj, which generously have allowed us the access to many resources in electronic form. Other XIXth century resources have been offered by Gabriela Haja, from the "Alexandru Philippide" Institute of Philology and Literary History in Iași, Cătălina Măranduc from the Faculty of Computer Science of the "Alexandru Ioan Cuza" University of Iași and the Natural Language Processing group from the Institute of Mathematics and Computer Science in Chișinău, to whom we address our thanks.

References

- Andriescu, Al. (1979). The Romanian press language in the XIXth century (Limba presei românești în secolul al XIX-lea). *Junimea Publishing House*, Iași.
- Chiswick, B.R., Miller, P.W. (2004). Linguistic Distance: A Quantitative Measure of the Distance Between English and Other Languages. In *Discussion Paper Series, Institute for the Study of Labor*, August 2004, IZA DP, 1246⁵.
- Colesnicov, A., Boian, E., Ciubotaru, C., Cojocaru, S., Malahov, L. (2014). Digitizarea, recunoașterea și conservarea patrimoniului cultural-istoric. In *Akados*, 32: 1, 61-68.
- Cristea, D., Haja, G., Moruz, A., Răschip, M., Pătrașcu, M. I. (2011). Partial statistics at the end of the eDTLR project – the Thesaurus Dictionary of the Romanian Language in electronic form (Statistici parțiale la încheierea proiectului eDTLR – Dicționarul Tezaur al Limbii Române în format electronic). In R. Zafiu, C. Ușurelu, H. Bogdan Oprea (eds.) *Romanian language. Hypostasis of linguistic variation. Acts of the 10th Colloquium of the Chair of Romanian*

⁵ <http://ftp.iza.org/dp1246.pdf>

HOW TO FIND OUT WHETHER THE ROMANIAN LANGUAGE WAS INFLUENCED BY THE TWO HISTORICAL UNIONS?

- Language* (Limba română. Ipoteze ale variației lingvistice. Actele celui de-al 10-lea Colocviu al Catedrei de limba română), Bucharest, 3-4 December, vol. I: *Grammar and phonology, lexicon, semantics, terminology, history of Romanian language, dialectology and philology* (Gramatică și fonologie, lexic, semantică, terminologie, istoria limbii române, dialectologie și filologie), Bucharest University Editing House, p. 213-224, ISBN 978-606-16-0046-5.
- Cristea, D., Simionescu, R., Haja, G. (2012). Reconstructing the Diachronic Morphology of Romanian from Dictionary Citations. In *Proceedings of LREC2012*, Istanbul, 21-25 May.
- Crystal, D. (1987). *The Cambridge Encyclopaedia of Language*, Cambridge: Cambridge University Press.
- Dascălu, M. and Gifu, D. (2015). Evaluating the complexity of online Romanian press, In *Proceedings of The 11th International Conference "Linguistic Resources and Tools for processing of the Romanian language"*, D. Gifu, D. Trandabăț, D. Cristea, D. Tufiş (eds.), "Alexandru Ioan Cuza" University Publishing House, Iași, pp. 149-162.
- Gifu, D. (2015). Contrastive Diachronic Study on Romanian Language. In *Proceedings FOI-2015*, S. Cojocaru, C. Gaindric (eds.), Institute of Mathematics and Computer Science, Academy of Sciences of Moldova, 2015, pp. 296-310.
- Gifu, D. (2016). Lexical Semantics in Text Processing. Contrastive Diachronic Studies on Romanian Language, *PhD Thesis*, Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iași, Appendix 1, 132-140⁶.
- Hamilton, W., Leskovec, J., Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL2016*, Berlin.
- Malahov, L., Cojocaru, S., Colesnicov, A. (2017). Digitization of Old Romanian Texts Printed in the Cyrillic Script. In *Conference Proceedings of Second International Conference on Digital Access to Textual Cultural Heritage, DATeCH2017*, Goettingen, 143-148, ISBN: 978-1-4503-5265-9⁷.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems*, Proceedings of a meeting held December 5-8, Lake Tahoe, Nevada, United States, 3111-3119.
- Pennington, J., Socher, R., Manning, C. D. (2014). GloVe: Global Vectors for Word Representation. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 1532-1543.

⁶ <https://profs.info.uaic.ro/~daniela.gifu/PhD%20Daniela%20Gifu%202016/PhD%20thesis%20Daniela%20Gifu%20final.pdf>

⁷ <http://dl.acm.org/citation.cfm?id=3078093>

CHAPTER 3

LINGUISTIC RESOURCES

MORE ROMANIAN WORD EMBEDDINGS FROM THE RETEROM PROJECT

VASILE PĂIȘ, DAN TUFIȘ

Research Institute for Artificial Intelligence, Romanian Academy
{pais,tufis}@racai.ro

Abstract

Automatically learned vector representations of words, also known as “word embeddings”, are becoming a basic building block for more and more natural language processing algorithms.

There are different ways and tools for constructing word embeddings. Most of the approaches rely on raw texts, the construction items being the word occurrences and/or letter n-grams. More elaborated research is using additional linguistic features extracted after text preprocessing. Morphology is clearly served by vector representations constructed from raw texts and letter n-grams. Syntax and semantics studies may profit more from the vector representations constructed with additional features such as lemma, part-of-speech, syntactic or semantic dependants associated with each word.

One of the key objectives of the ReTeRom project is the development of advanced technologies for Romanian natural language processing, including morphological, syntactic and semantic analysis of text. As such, we plan to develop an open-access large library of ready-to-use word embeddings sets, each set being characterized by different parameters: used features (wordforms, letter n-grams, lemmas, POSes etc.), vector lengths, window/context size and frequency thresholds.

To this end, the previously created sets of word embeddings (based on word occurrences) on the CoRoLa corpus (Păiș and Tufiș, 2018) are and will be further augmented with new representations learned from the same corpus by using specific features such as lemmas and parts of speech. Furthermore, in order to better understand and explore the vectors, graphical representations will be available by customized interfaces.

Key words – word embeddings, natural language processing, Romanian resources

1. Introduction

Word embeddings represent dense representations of words in a low dimensional vector space. Frequently there is made a contrast between the localist representations (the word themselves) and the distributed representations (the word embeddings). They are vectors of real numbers that can be used in natural language processing tasks instead of the words themselves, in order to improve the accuracy of different algorithms. This is happening because the vectors are capturing in some way the sense of the associated words. Thus, two words having similar meaning will have vectors with a small corresponding distance between them, as opposed to words with completely different meaning which will have distant vectors.

Modern approaches for computing vector representation of words make use of a famous observation made by linguist John Rupert Firth. According to this, “you shall know a word by the company it keeps” (Firth, 1957).

Distance between two vectors associated with words is usually computed using the “cosine distance”. This is also known as the “cosine similarity”, and uses the following formula:

$$\text{CosineDistance}(X,Y) = \frac{X \cdot Y}{||X|| ||Y||} = \frac{\sum_{i=1}^n X_i Y_i}{\sqrt{\sum_{i=1}^n X_i^2} \sqrt{\sum_{i=1}^n Y_i^2}}$$

where X, Y are the two vectors having dimension n , and X_i, Y_i represent the i -th element of the corresponding vectors.

Current state of the art methods for obtaining word embedding representations are using artificial neural networks that are trained to predict one or more words, given for input a sliding window of context words from the training corpus. This approach has the advantage of being completely unsupervised, requiring only a large training corpus in order to produce the representations. This approach was initially introduced by Bengio *et al.* as described in (Bengio *et al.*, 2003).

If the neural network is being trained to predict a middle word given a context window of m words, the model is referred to as CBOW (“Continuous Bag Of Words”). On the other hand, if the neural network is trying to predict the context based on a given central word, the model is referred to as Skip-gram. This is detailed in the paper by Mikolov *et al.* (Mikolov *et al.*, 2013). Both methods can produce comparable representations and these are actually the weights learned by the network. Vector length does not depend on the vocabulary size, being a network hyper-parameter (equal to the number of neurons on the hidden layer of the network, orders of magnitude smaller than the vocabulary). Contrary to other co-occurrence-based representations that depend on vocabulary size, word embeddings are dense representations in much less dimensional spaces, thus less computationally demanding.

In the ReTeRom project we are aiming to develop an integrated and configurable chain of advanced language processing modules, based on automated learning, for morphological, syntactic and semantic analysis of Romanian texts. The underlying data-source is the CoRoLa corpus, on which various algorithms were trained and tested. The new resources that will be created in the ReTeRom project will be added to the CoRoLa bimodal corpus.

The Reference Corpus for Contemporary Romanian Language (CoRoLa) (Barbu Mititelu, 2018) was constructed as a priority project of the Romanian Academy, between 2014 and 2017. It contains both written texts and oral recordings. Its aim was to cover major functional language styles (legal, scientific, journalistic, imaginative, memoirs, administrative), in four domains (arts and culture, nature, society, science) and in 71 sub-domains while taking into account intellectual property rights (IPR). With over 1 billion word tokens (written and spoken), CoRoLa (corola.racai.ro) is one of the largest fully IPR-cleared Reference Corpus in the world. CoRoLa is searchable via three interfaces (two for written part and one for the oral part) and it is supported by the KorAP corpus management platform, developed at Institute for German Language in Mannheim, (Bański *et al.*, 2014), (Diewald *et al.*, 2016).

Apart from the raw texts and sound records, the CoRoLa corpus contains linguistic annotations in the form of: phoneme, syllable, lemma, text-to-sound alignments, part of

speech (POS) tagging, syntactic chunking and dependency parsing. Several of these annotation types were added using the TTL (Tufiş *et al.*, 2008) service.

In a previous work (Păiş and Tufiş, 2018) we focused on computing word embeddings on the CoRoLa corpus, using the complete word forms. In the mentioned paper we provided various quantitative and qualitative details and compared the results with previous state of the art known vectors (Bojanowski *et al.*, 2016), (Pre-trained word vectors using Wikipedia corpus), obtained by Bojanowski and his colleagues using the Romanian Wikipedia corpus. We demonstrated that the CoRoLa based representations are superior to other known ones for the Romanian language. Therefore, it was natural to select these representations for use in the ReTeRom project.

2. Visualizing Romanian word embeddings

One problem we were faced with when thinking of ways to evaluate the quality of the learned representations was the difficulty to interact with the model in its vector format. On the other hand, integrating the vector representations into algorithms requires adequate means for retrieving, selecting, filtering and visualization of the word embeddings of interest. In the following we will describe the existing interfaces by means of which a user can investigate the significance of the numerical encodings of words and their used contexts.

Our previous evaluations indicated that the best representation using word forms from the CoRoLa corpus was based on vectors with 300 dimensions. The interrogation interface constructed previously allowed for asking the model to provide n similar words to a certain given word, or to ask a question in the format “what word is similar to word C in the same way as word B is similar to word A?”. This is also known as analogies. Using vector notation, this question can be formulated as “A-B+C”. Such a question with its associated answer is:

$$\text{vec}(\text{"rege"}) - \text{vec}(\text{"bărbat"}) + \text{vec}(\text{"femeie"}) = \text{vec}(\text{"regină"})$$

This can be formulated as: what word is similar to “femeie” (“woman”) in the same way as “bărbat” (“man”) is similar to “rege” (“king”). In this case, the answer is the expected one “regină” (“queen”).

Figures 1 and 2 shows examples of the two textual query interfaces.

Introduceti un cuvant pentru a obtine cuvinte similare.

Cuvant

anglia

Cuvinte similare

☐ Afisare vectori

Cuvinte similare pentru *anglia*:

franța	0.800742
britanie	0.798742
scoția	0.786095
anglie	0.749791
olanda	0.741721
angliei	0.737236
irlanda	0.726487
britania	0.712126
spania	0.709951
italia	0.707604

Figure 1. Textual interface for querying similar words

In Figure 1, the (10) words most similar to “anglia” are shown with the similarity score (cosine distance). Figure 2 shows a snapshot illustrating the linear algebraic properties of the

vectors, corresponding to the query: $\text{vec}(\text{"anglia"}) - \text{vec}(\text{"londra"}) + \text{vec}(\text{"paris"}) = ?$. The result, $\text{vec}(\text{"frança"})$, signifies that the vector for the word "frança" is the closest (in terms of cosine distance) to the vector resulting from the simple algebraic operations $A-B+C$.

Introduceți 3 cuvinte pentru a obține o analogie de forma $\text{vec}(A) - \text{vec}(B) + \text{vec}(C)$, unde $\text{vec}(A)$ este reprezentarea vectorială asociată cuvântului A.

Exemplu: $\text{vec}(\text{"rege"}) - \text{vec}(\text{"bărbat"}) + \text{vec}(\text{"femeie"}) = \text{vec}(\text{"regină"})$

A	B	C	Analogie
anglia	londra	paris	

☐ Afisare vectori

Introduceți un cuvânt pentru a obține cuvinte similare.

Cuvânt:

☐ Afisare vectori

$\text{anglia} - \text{londra} + \text{paris} = \text{frança}$

anglia, -0.067298, -0.02993, 0.3787, 0.1173, -4.075e-05, 0.2539, 0.21682, 0.042544, -0.25545, 0.25728, -0.22942, -0.40283, -0.18616, -0.092577,
londra, 0.067963, -0.27638, 0.43325, -0.085745, -0.012205, 0.31522, -0.0082534, 0.12693, -0.30797, 0.13642, -0.24564, -0.46198, -0.062385, -0.2
paris, -0.1082, -0.39873, 0.25235, 0.14347, -0.028161, 0.37071, -0.0554, -0.14923, -0.17052, 0.050167, -0.34329, -0.26453, -0.22367, -0.12912,
frança, -0.096536, -0.064186, 0.40725, 0.15805, 0.077997, 0.13999, 0.24847, 0.022921, -0.025512, 0.25634, -0.25958, -0.22056, -0.2103, 0.21607,
A-B, -0.135, 0.246, -0.055, 0.203, 0.012, -0.061, 0.225, -0.084, 0.053, 0.121, 0.016, 0.059, -0.124, 0.171, 0.016, -0.229, -0.118, 0.023, 0
A-B+C, -0.243, -0.152, 0.198, 0.347, -0.016, 0.309, 0.170, -0.234, -0.118, 0.171, -0.327, -0.205, -0.348, 0.041, -0.253, -0.406, 0.332, 0.0
(A-B+C)-R, -0.146, -0.088, -0.209, 0.189, -0.094, 0.169, -0.078, -0.257, -0.092, -0.085, -0.067, 0.016, -0.138, -0.175, -0.127, -0.269, 0.123, 0
cos(A-B+C;R), 0.793

Figure 2. Textual interface for querying analogies

At the bottom of the Figure 2 there are displayed the vectors for the words in the query, the partial results of the query ($A-B$, $A-B+C$), the vector of the word answering the questions ($A-B+C-R$) as well as the distance motivating the answer ($\cos(A-B+C;R)$).

Nevertheless, using such simple textual interfaces makes it hard to understand results comprised of more words, like the most similar 1000 words. For this reason, we decided to construct additional visualization interfaces for the use of ReTeRom project, using graphical approaches. The difficulty of constructing graphical interfaces for visualizing word vectors is given by their size. Thus, dimension reduction techniques must be used to project the vectors from dimension 300, as those used in the ReTeRom project, to the two dimensions that can be rendered on the computer screen.

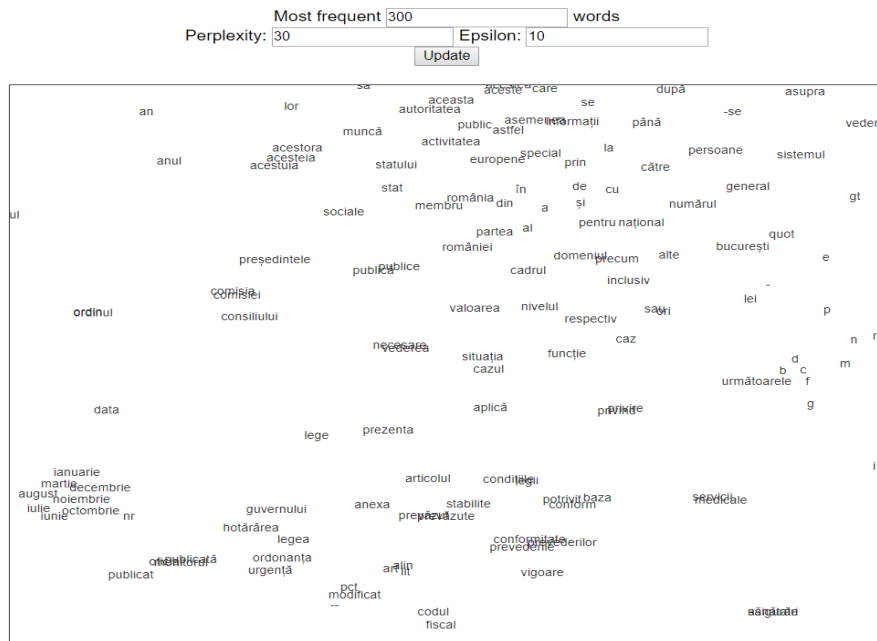


Figure 3. Using t-SNE to obtain a representation for the most frequent 300 words in CoRoLa.

The first interface constructed used the t-Distributed Stochastic Neighbor Embedding (also known as t-SNE) technique for dimensionality reduction, as described in (van der Maaten and Hinton, 2008), (van der Maaten, 2014). Since one of the objectives of the ReTeRom project is to obtain a web-based platform for interacting with the various tools and resources at project's disposal, we decided to use a JavaScript based implementation of the t-SNE algorithm, as described in (tSNE-JS). This was adapted to use the learned representations based on the CoRoLa corpus.

One implementation allows representing for the most frequent n words present in the corpus, with n being a parameter selectable by the user. An example for the most frequent 300 words is given in Fig. 3A second implementation focuses on representing n words most similar to a given word. In this way, the user has better control on what is displayed. An example is shown in Fig 4.

In Fig. 4 we used the same query as with the textual interface. However, this time it's easier to understand what kind of data is returned by the model. In this picture, in the lower right corner are grouped various countries (these were the results returned first in the textual query interface in Fig. 1). In the upper left corner of the picture, there are different cities and regions in England. In other parts of the image, it can be seen other related words, such as names of historical people ("Churchill", "Edward", "Stuart"). The interface based on the t-SNE algorithm is interactive, allowing the user to pan and zoom using the mouse, in order to better explore the generated image.



Figure 4. Using t-SNE based interface to show the most similar 300 words to the word „Anglia”

Another interface was constructed using a graph-based visualization method. In this case, the user starts by entering a word which will become the center of a tree and n most similar words will be brought by the system and connected to the given word. Afterwards the user can bring other similar words either by entering a new word or clicking on one of the words displayed. This will cause the system to load new nodes and connect them to existing ones, thus

constructing a graph with the words as nodes and the similarity between them indicated by edges.

An example representation using this interface, for the word “motocicletă” (“motorcycle”), with 50 similar words, is given in Fig. 5.

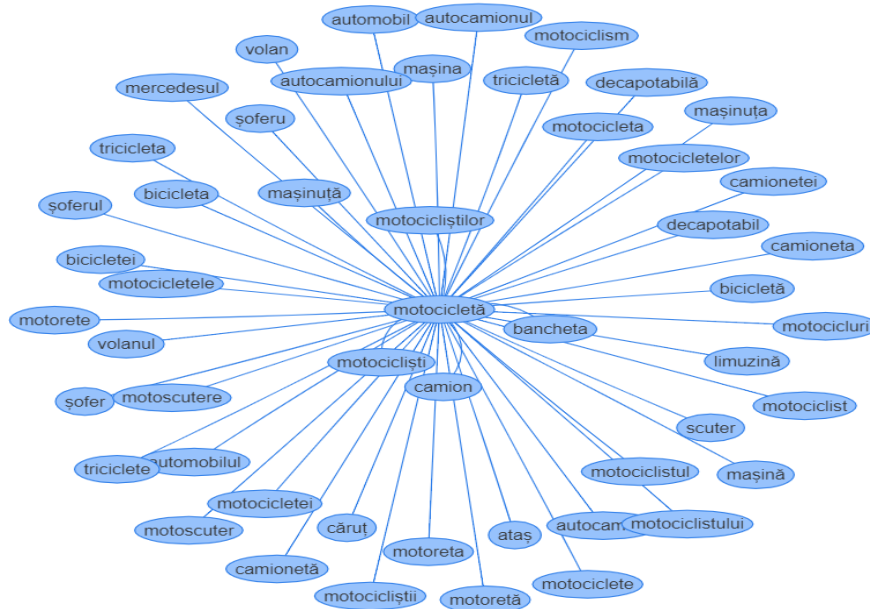


Figure 5. Graph based representation for the word “motocicletă” (“motorcycle”) with 50 similar words

Since the user can manually add multiple words to the visualization it is envisaged its suitability for identifying similarity paths between multiple words. For example, you may wonder if there is a connection between “motorcycle” and “boxing”. Apparently, there isn’t any direct connection. However, by adding on the same diagram the two words “motocicletă” and “box” a connection becomes visible. Both words are similar in a way to the word “motociclist” which is a sport (similar to “box”) which uses “motocicletă”. This can be seen in Fig 6.

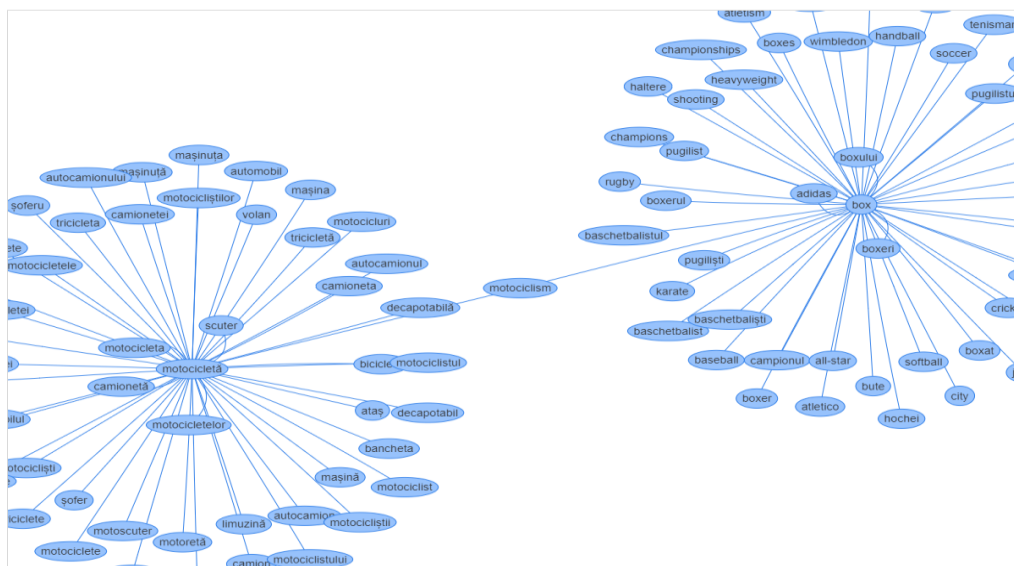


Figure 6. There is an indirect similarity connection between “motocicletă” and “box” illustrated in the graph-based interface

Graphical interfaces do not make obsolete the initial textual interfaces. Instead, they complement them since it's possible to use the textual analogies interface to ask an analogy question and then the graphical interface can be used to better understand why a certain word was chosen as answer. The four words from Fig 2. can be added on the same diagram, thus obtaining additional related words. This can be seen in Fig 7.

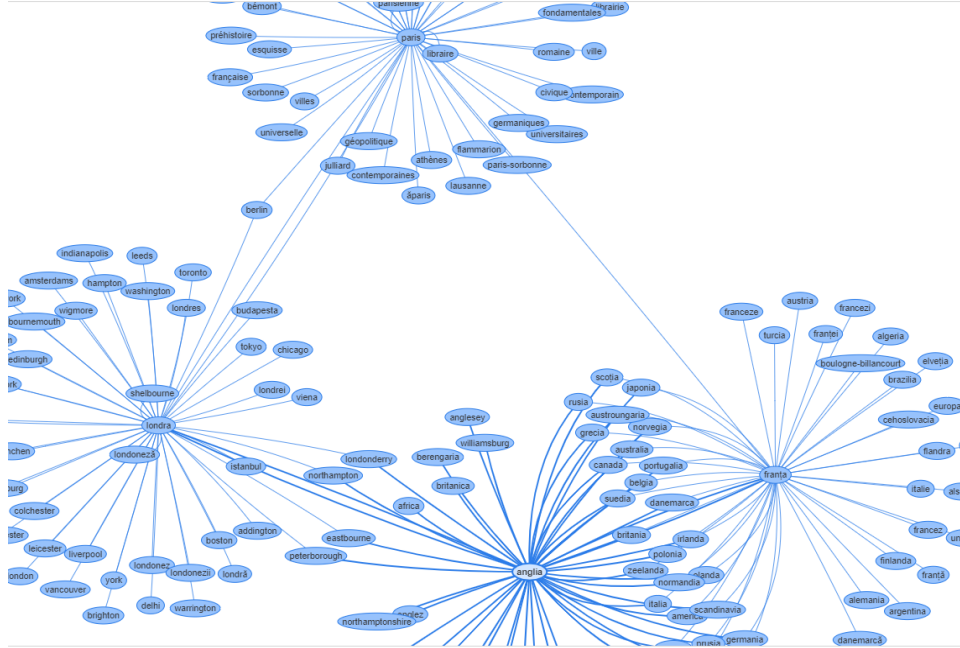


Figure 7. The words used in the textual interface in Fig. 2 shown on the graph-based interface

3. Necessity of other embeddings

Since the embeddings reported in (Păiș and Tufiș, 2018) were produced using the actual word forms available in the CoRoLa corpus, various inflected forms of the words are part of the representation. Analyzing the visualizations, it becomes obvious that amongst the top similar words reported for a given word there are actually several of its own inflected forms.

The presence of inflected forms is not an issue when using the model for computing distances between known words. However, one of the envisaged uses in the ReTeRom project is to obtain similar words and not just computing distances between given words. Therefore, the presence of inflected forms must be addressed.

Since the CoRoLa corpus already contain lemmas as part of its annotations, it becomes an obvious choice to use them for constructing new embedding models.

We decided to keep the model dimensions the same (300) and build two models, one based on the actual case sensitive lemma, as produced by the TTL web service (Tufiș *et al.*, 2008) and present in the corpus annotation, and another based on the lowercased lemma.

In the case of regular word embeddings, constructed based on the actual word form, as we did in (Păiș and Tufiș, 2018), different senses associated with the word are captured in the same vector. This happens because the vector is associated with the word form and not directly with a particular sense. It has been suggested that multiple senses of a word actually reside in linear superposition within the standard word embeddings (Arora *et al.*, 2016). This idea is challenging and will be explored by future research.

programmable means to extract the needed vectors, characterized by used specified features: word, lemma and/or POS-based, vector length, frequency-threshold used for the vector construction, list of nearest similarity neighbors of a vector for a given word, etc.).

5. Conclusions

Word embeddings are becoming a common way of representing the words in various natural language processing tasks. Therefore, for the purposes of the ReTeRom project, this kind of representation will become one of the resources used. Previously computed word embeddings on the CoRoLa corpus will be used, together with new representations learned using lemmas.

Furthermore, in order to visually explore these resources, several web based interfaces were constructed, using simple textual searches, t-SNE dimensionality reduction technique and a graph based approach for connecting visually similar words.

Acknowledgements

This work is supported by a grant (73PCCDI/2018) of the Ministry of Research and Innovation CCCDI-UEFISCDI, project code PN-III-P1-1.2-PCCDI-2017-0818 (ReTeRom) within PNCDI III.

References

- Firth, J.R. (1957). “Papers in Linguistics 1934–1951”, London: Oxford University Press.
- Bengio, Y., Ducharme, R., Vincent, P. (2003). “A neural probabilistic language model”, *Journal of Machine Learning Research*, 3:1137-1155.
- Mikolov, T., Chen, K., Corrado, G., Dean, J. (2013). “Efficient Estimation of Word Representations in Vector Space”, *arXiv:1301.3781*.
- Barbu Mititelu, V., Tufiş, D., Irimia, E. (2018). The Reference Corpus of Contemporary Romanian Language (CoRoLa). In *Proceedings of the 11th Language Resources and Evaluation Conference – LREC’18*, Miyazaki, Japan, European Language Resources Association (ELRA)
- Bański, P., Diewald, N., Hanl, M., Kupietz, M., Witt, A. (2014). Access Control by Query Rewriting. The Case of KorAP. In *Proceedings of the Ninth Conference on International Language Resources and Evaluation (LREC’14)*. Reykjavik, European Language Resources Association (ELRA)
- Diewald, N., Hanl, M., Margaretha, E., Bingel, J., Kupietz, M., Bański, P. and A. Witt (2016). KorAP Architecture – Diving in the Deep Sea of Corpus Data. In: *Calzolari, Nicoletta et al. (eds.): Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC’16)*, Portoroz, European Language Resources Association (ELRA).
- Tufiş, D., Ion, R., Ceaşu, A., Ştefănescu, D. (2008). “RACAI's Linguistic Web Services”, in *Proceedings of the 6th Language Resources and Evaluation Conference – LREC’08*, Marrakech, Morocco.
- Păiş, V., Tufiş, D. (2018). “Computing distributed representations of words using the CoRoLa corpus”, *Proc. Ro. Acad., Series A*, Volume 19, No. 2, pp. 403-409.
- Bojanowski, P., Grave, E., Joulin, A., Mikolov, T. (2016). “Enriching Word Vectors with Subword Information”, *arXiv:1607.04606*.

Pre-trained word vectors using Wikipedia corpus,
<https://github.com/facebookresearch/fastText/blob/master/pretrained-vectors.md>

L.J.P. van der Maaten and G.E. Hinton (2008). Visualizing High-Dimensional Data Using t-SNE. *Journal of Machine Learning Research* 9(Nov):2579-2605.

L.J.P. van der Maaten (2014). Accelerating t-SNE using Tree-Based Algorithms. *Journal of Machine Learning Research* 15(Oct):3221-3245.

tSNE-JS, <http://cs.stanford.edu/people/karpathy/tsnejs/>

Arora, S., Li, Y., Liang, Y., Ma, T., Risteski, A. (2016), "Linear Algebraic Structure of Word Senses, with Applications to Polysemy", *arXiv:1601.03764*.

Ion, R. (2018). "ReTeRom/**TEPROLIN 1.5**: Definirea specificațiilor funcționale și arhitecturale ale platformei integrate și configurabile de prelucrare a textelor". Raport tehnic 1.5, <http://dev.racai.ro/ReTeRom/>

Ion, R. (2018). "ReTeRom/TEPROLIN 1.6: Definirea modulelor software și a serviciilor oferite de proiect; identificarea adaptărilor pentru modulele NLP existente și a modulelor noi necesare", Raport tehnic 1.6, <http://dev.racai.ro/ReTeRom/>

VALENCE DICTIONARY FOR ROMANIAN LANGUAGE IN PRINTED VERSION AND XML FORMAT

ANA-MARIA BARBU

Institute of Linguistics in Bucharest, Romanian Academy, anamaria.barbu@g.unibuc.ro

Abstract

This paper presents the encoding structure of a verb valence dictionary for Romanian, comprising 600 verbs and over 2,000 meanings with one or more valency patterns. This dictionary is devoted both for teaching Romanian to foreign students, by its printed version, and to natural language processing (NLP), by its XML format. The information structure of an entry is presented in detail, together with the corresponding XML description, as it is done in the Document Type Definition of the dictionary, especially for language engineering purposes.

Key words — argument structure, complements, subcategorization, verb pattern, XML elements and attributes.

1. Introduction

This study presents a valence dictionary for Romanian language, which we have conceived as a linguistic resource, in original and annotated format. It has been published under the name *Dicționar de Contexte Verbale* 'Dictionary of Verbal Contexts' (DCV) (Barbu 2017). A valence dictionary is a valuable resource in many respects, but we focus here only on two of them: foreign-language teaching and computational linguistics. Initially, German linguists pointed out the utility of such a dictionary for teaching German language to foreign students. They have highlighted that the set of complements required by a verb and their morphosyntactic characteristics – that is, the *valence* of the verb – is language-specific and it has to be learnt as such. Thus, in the 1970s, the valence dictionaries of Helbig and Schenkel (1968) and Engel and Schumacher (1976) appeared and, together with them, the basis of a *valence theory* was established, even if the valence concept, as it is well known, is rooted in Tesnière's theory of dependency grammar. From the point of view of teaching, a valence dictionary has to be as rich as possible, providing students with relevant information about complements, such as part-of-speech, grammatical cases and specific morpho-syntactic characteristics (i.e. clitics, verbal moods etc.), specific prepositions and conjunctions, semantic information etc. All this information has to be presented in a quite concise way and has to be illustrated by relevant examples. We strive to offer all of these in our dictionary and even more, adding some insights of cognitive linguistics.

From the computational linguistics perspective, a valence dictionary can be especially used in syntactic parsing and word sense disambiguation. In our opinion, it has to be formally consistent, built in a machine-readable format and, as much as possible, based on information directly retrievable from the textual context. In order to reach these desiderata, we have adopted a machine-oriented way of thinking, driven by a rigorous set of conventions, with the risk of overwhelming the human-reader sometimes. The machine-readable format we have chosen is XML. Regarding the information retrievability, we have focused on what is

formally visible, for instance on grammatical cases, instead of grammatical functions (e.g. direct or indirect object) and on semantic features (available, to some extent, in a monolingual lexicon, such as *animate*, *human*, *object* etc.) instead of semantic roles (e.g. Agent, Theme, Beneficiary etc.).

Our dictionary, DCV, is manually built, by only one linguist, but based on corpus information. It comprises 600 verb entries covering about 2000 meanings with one or more valence patterns. The verbs inventory is extracted from a newspaper corpus (ZiareRom), depending on their appearance frequency, and significantly reflects the journalistic language, what means that some verbs used in private life can be still absent (e.g. *a mânca* 'eat' or *a dormi* 'sleep'). Even so, the most used verbs in Romanian are not missing.

The dictionary editing has been done with the Professional Lexicography Software TshwaneLex (tshwanedje.com). It has many facilities, but we only mention the XML editor, the option of RTF or HTML export and the so-called WYSIWYG ("what you see is what you get") view. These facilities help to easily get the dictionary in machine-readable and printed formats. We had to build a particular DTD, appropriate for verb valence description, which is described in the main section of the paper.

Barbu (2018) also presents DCV but in a conceptual manner, by focusing on building procedure and work evolution, and on linguistic aspects, which explain the different choices of valence representation. The present study, instead, details the technical part of the dictionary, comprising the main information elements of the encoding.

The paper includes, in the next section, a short overview of the related work in the valence-dictionary domain, whereas, in the main section, detailed information about the structure of the dictionary is provided. Each of the information types viewed in the printed format is associated with a corresponding XML element by indicating its name, attributes and children, as it is described in the DTD (Document Type Definition) of this dictionary. A section of final remarks concludes this study.

2. Related work

Our dictionary has, as conceptual models, FrameNet (Johnson and Fillmore 2000), VerbNet (Kiper *et al.*, 2000), CPA (Hanks and Pustejovsky, 2005) and VALLEX 2.0 (Žabokrtský and Lopatkova, 2007) described and analysed in (Barbu, 2008). In the meantime, other interesting achievements in valence-lexicon domain have been presented in many studies for several languages. We limit ourselves to mentioning some dictionaries that have electronic (online or downloadable) variants, usable in education or in NLP. For German, there is the valence dictionary VALBU (Schumacher *et al.* 2004), and its online version E-VALBU (Schneider, 2008). This dictionary comprises 638 lemmata and more than 3000 verb variants, together with detailed information about morphology, word formation, phraseology, and stylistics. For Croatian, there are two lexica: one is Crovallex (Mikelić Preradović *et al.*, 2009), which has over 1700 verbs accessible in a browsable version; another is e-Glava (Birtić *et al.*, 2017), which focuses on about 60 psychological verbs, available online as well, but which has been extended to 900 verbs (not yet accessible). For Polish, we can mention Walenty (Przepiórkowski *et al.*, 2014), a large valence dictionary of 8600 verbal lemmata and almost 39000 valence schemata. Besides the impressive number of entries, this dictionary is also distinguished by the richness of encoded information, comparable to that found in VALBU

VALENCE DICTIONARY FOR ROMANIAN LANGUAGE IN PRINTED VERSION AND XML FORMAT

and in our dictionary. Also, it is worth mentioning some Romance languages that enjoy such a resource. For Spanish, there is the valence dictionary ADESSE (García-Miguel *et al.*, 2010). This is an online database, where the grammatical features of verb-arguments are described for 3450 Spanish verbs. For Italian, one can mention two such dictionaries: T-PAS (Jezek *et al.*, 2014) with about 1000 verbs described in CPA-style; and LexIt (Lenci *et al.*, 2012), a computational resource which extracts subcategorization frames and semantic selection properties from a corpus, fully on distributional criteria. Let's also add two French dictionaries: Dicovalence (Département de linguistique, 2017), gathering 3700 verbs and over 8000 valence schemata; and, especially, «Les verbes français» (LVF) (Dubois and Dubois-Charlier, 1997; Hadouche and Lapalme, 2010).

For Romanian, efforts were directed exclusively towards teaching Romanian language by building printed dictionaries like (Ionescu and Streiu, 1999) or (Drăghicescu 2002). These works describe argument structures by minimal information, referring, in principal, to nominal and prepositional complements in a literal manner, for instance „*a conduce pe cineva (până) la...*” (to lead somebody to/until...).

The dictionary we are presenting here is much richer in information than the previous ones, but what distinguishes it the most is its formal description, which we are trying to render in this article. This is the first verb valence dictionary for Romanian meant to be used in computational linguistics or as an online data base, like the dictionaries mentioned above for other languages.

3. Main features encoded in DCV's information structure

An entry has the information structure illustrated, in Fig. 1, on an example in printed format. At the first level, there are three main information fields: the lemma, a set of verb contexts and, optionally, a set of expressions (idioms containing lemma verb). Note that each field can contain textual information (represented by its XML attributes) and/or other structured sub-fields (represented by its XML child elements).

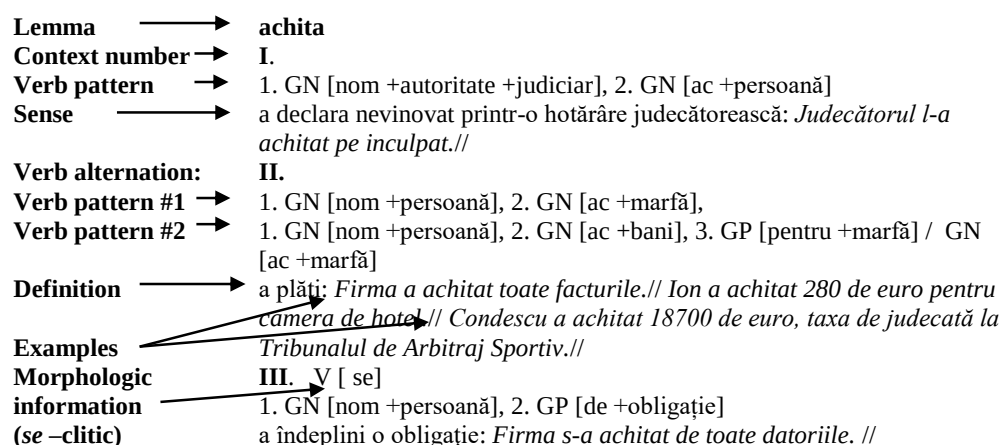


Figure 1: Entry structure in printed version

XML Format. The XML representation of an entry is delimited by the element <Lemma> comprising the attribute LemmaSign=text and the following child elements: <ArgUnit>⁺ for verb contexts, and <ExpressionList>^{*} for expressions.¹

3.1. Lemma

The lemma is, by convention, the bare infinitive form of the verb. For the reflexive verbs, the corresponding mark is not registered in lemma field (as it is usually done, see also Birtić *et al.*, 2017:52) but in that of morphologic characteristics of the verb, as a reflexive clitic. This option offers more encoding flexibility for describing some linguistic facts, such as alternations. Note that in Romanian there are two kinds of reflexive clitics, in accusative: *se* (eg. a *se* căi 'repent') and in dative: *își* (eg. a *își* imagina 'imagine'), that are mentioned as such in our encoding (see Fig. 1, context III).

XML Format. Printed lemma corresponds to the value of the attribute LemmaSign of the Lemma element.

3.2. Verb context

A verb is assigned one or more contexts. By *context* we mean the minimal valence of the verb expanded with typical adjuncts, which speakers frequently associate with that verb, such as adjuncts expressing (parts of) the Path (Origin, Target, Path) associated with directed motion verbs or with verbs implying movement (e.g. to call "demand to come"), the Instrument associated with a lot of human activity verbs etc.

A context comprises the following information:

- the context number, expressed by Roman numbers;
- morphological characteristics of the verb, applicable to the whole context (see Fig. 1, context III), if it is the case;
- a verb pattern (see Fig. 1, context I) or a set of verb patterns forming an *alternation*, defined in Levin (1993) (see Fig. 1, context II);
- one or more senses corresponding to the verb pattern, which are defined as pairs of definition(s) and example(s);
- an optional note (or clue).

XML Format. A verb context is described by the ArgUnit element which has the attribute ArgUnitNr=number indicating the context number, and the following child elements: <LemmaFeature>^{*}, <ArgStrucure>⁺ – for the verb pattern, <Sense>⁺ – for the verb pattern meanings and examples, and <Clue>^{*}.

3.2.1. Morphological characteristics of the verb

This information is applicable to the whole context (including one or more verb patterns). It is enclosed inside the square brackets of the mark V[...] and refers to the following aspects:

- a) the pronominal clitics: *se*, *își*, *o*, *îi* or the negation: *nu* (*mai*);
- b) the third person singular and plural: 3, or the third person singular:3sg

¹ The symbols used as superscripts in XML description are 1 – strictly one, (0,1) – zero or one, * – zero or more, + – at least one, and inside brackets there are lists of possible attribute values. We also use @ for attributes and brackets <...> for elements.

VALENCE DICTIONARY FOR ROMANIAN LANGUAGE IN PRINTED VERSION AND XML FORMAT

c) the verb mood or tense, e.g. ind. imperf. for indicative imperfect.

XML Format. The morphological features of the verb are described within the LemmaFeature element which consists of the following set of attributes: @ReflForm={își, îi, o, se}, @NegForm=text and @FlexForm=text.

3.2.2. Verb pattern

A verb pattern (or verb valence or subcategorization frame) is illustrated in Fig. 2. This is the second verb pattern in the context of the verb *a alerta* 'alert' described in Ex. 1:

Example 1:

alerta

I.

1. GN [nom +cauză +negativ], 2. GN [ac +persoană],

V [se] 1. GN [nom +persoană], 2. (fac.) GP [din cauza / de la +cauză] / GV [când / (pentru) că +cauză]

a (se) alarma: *Gripa porcină a alertat autoritățile județului.// Toată lumea s-a alertat din cauza lipsei ei.// Poliția s-a alertat când a aflat profilul suspectului.// Soțul ei s-a alertat pentru că nu mai răspundea.//*

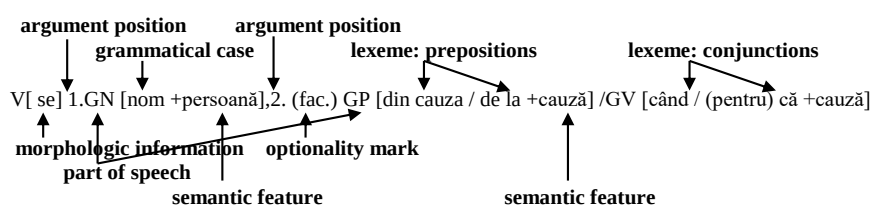


Figure 2: Verb pattern

As depicted in Fig. 2, a verb pattern contains:

a) A field with morphologic information (e.g. the clitic *se*, *își* or *o*), enclosed inside square brackets preceded by V. This field is optional, and, at this level, it only appears in an alternation of verb patterns (see Ex. 1).

b) One or more argument positions. An argument position can be occupied by an argument or by a set of arguments (i.e. variants). Thus, the concept of argument position is important for cases in which the same argument can be realized by different syntactic phrases, like the argument on position 2, which can be expressed by a prepositional phrase (GP) or by a clause (GV) denoting the alert cause. The variants always exclude each other, that is, they never cooccur in the same utterance. Przepiórkowski *et al.* (2014) have also defined the argument-position concept but settled the coordination principle for identifying the variants, instead of the mutual-exclusion one (see (Barbu, 2018) for a discussion). In our encoding, variants are separated by /.

By convention, for the verbs that allow auto-transitivity, when the denoted action affects not the direct object but the subject itself, we stipulate as variants GN[ac] / V[se], like in Ex. 2:

Example 2:

atârna 'hang'

II.

1. GN [nom +uman 'human'], 2. GN [ac +obiect 'object'] / V [se], 3. (fac.) GP [de / pe / în 'of / on / in']

a suspenda: *O haină atârână în cuier. // Copilul se atârână de mama lui.*

'to suspend: *A coat hangs in the hanger. // The baby SE hangs on his mother.*

If an argument position can be omitted (because it may not be relevant in a certain context), it is assigned the optionality mark *fac.* or *opt.* inside parentheses. The mark *fac.* ('*facultative*') means that the marked complement can simply be unexpressed, whereas *opt.* ('*optional*') operates in a set of complements in which any marked complement can be omitted but not all of them. See (Barbu, 2018) for details.

c) One or more arguments occupying the same argument position. The main information of an argument refers to the complement head and indicates the following aspects divided into three fields:

Field 1: *Part of speech.* Our encoding uses the following parts of speech: GN – noun phrase, GAdj – adjective phrase, GAdv – adverb phrase, GP – prepositional phrase, V – verb phrase with the same subject as the lemma, GV – a subordinating clause or a verb phrase possibly having a different subject than the lemma. By convention, a subordinating clause, unlike a verb phrase, is always characterized by a semantic feature, e.g. +cause, +fact, +information etc.

Field 2: One of the following information types may exclusively occupy this field:

i) *Grammatical case.* This information is specific to GN and GAdj and, in this dictionary, is limited to the values nominative (nom), accusative (ac) and dative (dat) because the other Romanian cases, genitive and vocative, are not relevant for verb valence. Note that, by means of the grammatical case, it is indicated the agreement between complements of the same verb pattern. For instance, the predicative element expressed by GAdj displays agreement with the subject in Ex. 3, and with the direct object in Ex. 4.

Example 3:

1. GN [**nom**], 2. GN [nom] / GAdj [**nom**] / GP [de] / V [de sup.]

Președintele.NOM a ajuns și mai puternic.NOM / de rușine / de neînvinș.
The president got even stronger / ashamed / unbeaten.

Example 4:

1. GN [nom +persoană], 2. GN [**ac**], 3. GAdj [**ac**] / GP [drept / ca]

Judecătorul l-a găsit pe Ion.AC vinovat.AC.

The judge found Ion guilty.

ii) *Lexical information.* This can be a set of prepositions for GP, an adverb for GAdv or a set of conjunctions for GV with clause value, as in Fig. 2.

iii) *Verbal morphological information.* This indicates moods, clitics (*se, își, nu (mai), o, îi*), or persons (3sg – third person singular; 3 – third person singular and plural). The encoded moods are: gerundive (ger.), imperative (imper.), indicative (ind.), infinitive (inf.), participle (part.)

VALENCE DICTIONARY FOR ROMANIAN LANGUAGE IN PRINTED VERSION AND XML FORMAT

and supine (sup.). The supine mark can be preceded by a specific preposition (see *de sup.* in Ex. 3).

Field 3: *Semantic information*. The third field is exclusively dedicated to the semantic information, marked with +. We use semantic features (also named semantic restrictions or preferences) instead of semantic roles, which are frequently used in other verb valence descriptions. We consider the semantic roles being in total agreement with Birtic *et al.* (2017: 58) which claim: "Since it is sometimes an intricate task to properly assign a semantic role to a specific participant, we decided to use semantic (conceptual) categories, e.g., person, animal, place, etc.". Besides, semantic features can be easier to manage computationally based on the Wordnet information, for instance, because they can be lexically determined, unlike the semantic roles. Also, the semantic features may be the only ones that distinguish the verb patterns. For instance, in Ex. 5, the semantic features +temă 'theme' and +uman 'human' are the only ones that distinguish the meanings of the two verb contexts. *Otherwise, if semantic roles are used, it results only one verb pattern, namely "1. GN [nom Agent], 2. GN [ac Theme]", which is less relevant for tasks of word sense disambiguation.*

Example 5:

aborda 'approach'

I.

1. GN [nom +persoană], 2. GN [ac +temă]

a pune în discuție; a trata: *Comisia a abordat problema adopției.*

'to question; to address: *The Commission approached the issue of adoption.*'

II.

1. GN [nom +uman], 2. GN [ac +uman]

a se apropia de cineva pentru a-i vorbi: *Dan l-a abordat pe profesor pentru a-i pune întrebări suplimentare.*

'to approach someone to talk to him: *Dan approached the teacher to ask him extra questions.*'

We have defined a set of 90 semantic features, discovered by generalizing over sets of lexical items found in the argument positions in the corpus, such as:

+abilitate is defined as "ability to do something"; it appears, for instance, in valence patterns of verbs like *antrena 'train', cultiva 'cultivate' etc.*;

+animat – "(domestic) animal; it includes human being", – e.g. *abandona 'abandon', accidenta 'hurt' etc.*;

+cauză – "anything can trigger or maintain a certain state of affairs; it includes a circumstance; it includes +stimulus" – e.g. *agrava 'worsen' or alerta 'alert'*;

+fapt – "what has happened or is likely to happen in reality" – e.g. *absolvi 'absolve', califica 'qualify' or prevedea 'stipulate'*;

+informație – "what is being communicated" – e.g. *afla 'find out', aminti 'remember' or arăta 'show'*;

+persoană – "natural or legal person" – e.g. *abandona 'abandon' or abate 'turn'*;

+uman – "human individual. Unlike +persoană 'person', this concept involves sensory and motor manifestations" – e.g. *abandona 'abandon' or abate 'turn' etc. etc.*

Sometimes we have used semantic features in connection with prepositional phrases lacking a lexical information, in order to give a hint about what kind of prepositions characterizes the

complement, like in Ex. 6. The semantic feature +cause stays for all the prepositions able to express a cause: *de, de la, din cauza, din pricina, datorită, la* etc. For computational purposes, this convention raises the need to obtain additional information (about preposition semantics) from sources external to this dictionary.

Example 6:

chinui 'torture'

II. V [se]

1. GN [nom +viu 'alive'], 2. (fac.) GP [+cauză 'cause']

a suferi: *Câinii se chinuie de foame.*

'to suffer: *Dogs suffer from hunger.*'

Note that the presence of several semantic features is allowed in this field either with a cumulative reading (see +*autoritate* +*judiciar* in Fig. 1 for "legal authority") or with a disjunctive one, marked by / (as possible variants).

XML Format. The verb pattern is described in the ArgStructure element, which has two children <LemmaFeature>* and <ArgPosition>+.

<LemmaFeature> has the structure described in section 3.2.1 above.

<ArgPosition> comprises @PositionNr=number, <PositionState>^(0,1) and <Argument>+.

<PositionState> is made up of @ PositionState=text, i.e. fac. or opt.

<Argument> contains @POS={V, GAdj, GAdv, GN, GP, GV} and <AFeature>¹.

<Afeature> only has @Gram=text (i.e. Field 2) and @Sem=text (i.e. Field 3).

3.2.3. Sense

A sense comprises a) a meaning described by one or more definitions and b) one or more examples.

a) Information about meaning is expressed by one or more synonyms or by a short paraphrase. We tried to choose synonyms that, as much as possible, can successfully replace the lemma verb in the examples.

It is worth mentioning that this verb valence dictionary is a meaning-oriented one, that is, for each meaning registered in conventional dictionaries one or more possible verb patterns are provided. This differs from corpus-oriented valence lexica, which register all the verb patterns found in corpus, irrespective of their meaning. At first, we used the Romanian Explanatory Dictionary (DEX) for the meanings inventory of a verb but we should realize that the meaning identification in DEX does not fit with the appropriate valence patterns of the verb. Actually, it has been reconfirmed the claim of Herbs *et al.* (2004: xxxviii) that "Establishing senses according to their valency patterns in some cases results in a rather different identification of senses than in conventional dictionaries".

There are cases in which, for the same verb pattern, more slightly different meanings are appropriate, because there are no specific semantic features that distinguishes them. In Ex. 7, there is such an example with two meanings, marked by a. and b.:

Example 7:

cutremura 'shake'

I.

1. GN [nom +cauză 'cause'], 2. GN [ac +corp 'physical body'],

V [se] 1. GN [nom +corp], 2. (fac.) GP [de la +cauză]

VALENCE DICTIONARY FOR ROMANIAN LANGUAGE IN PRINTED VERSION AND XML FORMAT

a. a produce / suferi un cutremur: *Seismul a cutremurat pământul.// Pământul se cutremură.//*

'to produce /suffer an earthquake: *The earthquake shook the earth.// The earth shakes.*'

b. a (se) zgâlțâi: *Casele s-au cutremurat (de la explozie).//*

'to tremble: *The houses shook (from the explosion).*'

b) Examples, written in italics and separated by '//', make a pair with a meaning. They cover every variant of verb pattern, part of speech, lexeme etc. used in the valence description. They are created by the author for the usual meanings or are adapted corpus examples for special uses or meanings. Facultative arguments are placed inside parentheses (see Ex. 7, meaning b.). Variants described in a verb pattern can be reproduced in the same example, if this is possible. They are also marked by '/', as one can see in Ex. 8 below.

XML Format. The sense information is described by the element *Sense*, formed by @SenseNumber=number, <Definition>⁺ and <Example>⁺.

<Definition> is made up by @Definition=text.

<Example> is made up by @Example=text.

3.2.4. Note

A note assigned to a verb context gives supplementary information about the use of the verb in that context or about other aspects. For instance, in Ex. 8, the note, marked by Obs., indicates that, in the corresponding context, the verb can be used only in reflexive-passive form.

Example 8:

dispune 'order'

II. V [se 3]

1. GN [nom +rezultat 'result'] / GV [(ca) să +rezultat]

a fi luată o hotărâre: *S-a dispus suspendarea / să se suspende legea.// [...]*

'to make a decision': Suspension of the law was ordered.

Obs. Cu sens pasiv; singura formă de pasiv admisă.

'Reflexive with passive meaning; the only passive form allowed.'

XML Format. The element corresponding to a note is *Clue*, made up by @Clue=text.

3.3. Expression

An entry also contains one or more expressions (or idioms) centred on the lemma, if it is the case. Idioms are described in DCV because they can be considered verbal contexts with lexicalized complementation. The verb *a ajunge* 'reach' has the idiom in Ex. 9, where what follows the '=' mark is its gloss. An idiom can have a fixed part, conventionally written with italics (including the verb itself), and a mobile one, written with normal character. The mobile part can take different references (see *cuiva* 'to somebody' in Ex. 9) or can be inflected.

Example 9:

a-i *ajunge* *cuiva* *cuțitul* *la os* = a fi într-o situație disperată
to-clitic.dat reach somebody.dat knife.the to bone = to be in a desperate situation

XML Format. The set of idioms is described in ExpressionList element, which has as unique child <Expression>*. <Expression> is made up by two attributes: @LemmaSign=text and @Definition=text.

Fig. 3 synthesizes the DTD of the dictionary and also shows the XML encoding of the Ex. 8.

4. Final remarks

In our opinion, even if the actual version of the verb valence dictionary for Romanian is rather a small one, it is sufficiently comprehensive for language engineering experiments on word sense disambiguation or syntactic parsing. To our knowledge, some attempts of semantic disambiguation have used common monolingual dictionaries. But if, as an alternative, a valence dictionary has been taken as the reference, we would expect the results to be greatly improved, because such a resource better reflects the formal aspects found in corpus. Actually, this dictionary is the first endeavour at this level of comprehensiveness and consistency for Romanian, covering important linguistic facts, such as part-of-speech variants competing in the same argument position, control and raising verbs, dependencies between the complements of a verb pattern or fine distinctions between verb patterns based on the semantic features of the arguments. All of these are encoded in the consistent way provided and validated by the XML module of a professional lexicographic editor.

DTD schema	XML example for <i>dispune II.</i>
Lemm:LemmaSign = text ArgUnit ⁺ : ArgUnitNr = integer LemmaFeature [*] : ReflForm ^(0,1) = [se, își, o, îi] NegForm ^(0,1) = text FlexForm ^(0,1) = text ArgStructure ⁺ : LemmaFeature [*] : ReflForm ^(0,1) = [se, își, o, îi] NegForm ^(0,1) = text FlexForm ^(0,1) = text ArgPosition ⁺ : PositionNr=integer PositionState ^(0,1) :PositionState=text Argument ⁺ : POS ¹ =[GN, GAdj, GAdv, GP, GV, V] Afeature ¹ : Gram [*] = text Sem [*] =text Sense ⁺ : SenseNumber=integer Definition ⁺ : Definition ⁺ =text Example ⁺ : Example ⁺ =text Clue [*] : Clue=text Expression [*] : LemmaSign=text Definition=text	<Lemma LemmaSign="dispune" <ArgUnit ArgUnitNr="2"> <LemmaFeature ReflForm="se" FlexForm="3"/> <ArgStructure > <ArgPosition PositionNr="1"> <Argument POS="GN"> <Afeature Gram="nom" Sem="+rezultat"/> </Argument> <Argument POS="GV"> <Afeature Gram="(ca) să□" Sem="+rezultat"/> </Argument> </ArgPosition> </ArgStructure> <Sense SenseNumber="1"> <Definition Definition=" a fi luată o hotărâre"/> <Example Example="S-a dispus suspendarea / să se suspende legea."/> <Example Example="S-au dispus măsuri compensatorii."/> <Example Example="S-a dispus ca școlile să se închidă pe perioada viscolului."/> </Sense> <Clue Clue="Cu sens pasiv; singura formă de pasiv admisă."/> </ArgUnit> [...] </Lemma>

Figure 3: XML Encoding

Of course, this enterprise is far from being finished, because further work has to be done for improving the actual encoding and add new types of information, such as semantic roles or

VALENCE DICTIONARY FOR ROMANIAN LANGUAGE IN PRINTED VERSION AND XML FORMAT

diathesis uses, for increasing the number of entries and for getting different kinds of generalizations. We hope to get valuable feedback from the users of this dictionary, both in the educational domain and the computational linguistics, in order to have supplementary guidelines for improvement.

The dictionary in XML format is available for free, by request from the author. We hope that the dictionary will be able to prove its usefulness in various NLP tasks and that it will soon be available as an online data base.

References

- Barbu, A.M. (2008). First steps in building a verb valency lexicon for Romanian. In P. Sojka *et al.* (eds.), *Text, Speech and Dialogue*, Lecture Notes in Artificial Intelligence (LNAI) 5246, Springer, 29-36.
- Barbu, A.M. (2017). *Dicționar de Contexte Verbale* 'Verbal Contexts Dictionary'. Editura Universității din București.
- Barbu, A.M. (2018). Dictionary of Verbal Contexts for the Romanian Language. In *Proceedings of the XVIII EURALEX International Congress*, July 17-21, Ljubljana, Slovenia, 411-422.
- Birtić, M., Brač, I., Runjaić, S. (2017). The main features of the e-Glava online valency dictionary. In *Proceedings of eLex 2017 Conference "Electronic Lexicography in the 21st Century"*, September 19-21, Leiden, the Netherlands, 43-62.
- Département de linguistique (2017). *Dicovalence* [Lexique]. ORTOLANG (Open Resources and TOols for LANGuage) -www.ortolang.fr, <https://hdl.handle.net/11403/dicovalence/v1>.
- Dubois J., Dubois-Charlier, F. (1997). *Les Verbes français*, Paris: Larousse-Bordas.
- Drăghicesu, J. (ed.) (2002). *Dicționar de construcții verbale – română, franceză, italiană, engleză*. Craiova: Editura Universitaria.
- Engel, U., Schumacher, H. (1976). *Kleines Valenzlexikon Deutscher Verben*, Tübingen: Narr.
- García-Miguel, J. M., González Domínguez, F., Vaamonde, G. (2010). ADESSE. A Database with Syntactic and Semantic Annotation of a Corpus of Spanish. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, May 17-23, Valletta, Malta, 1903-1910.
- Hadouche, F., Lapalme, G. (2010). Une version électronique du LVF comparée avec d'autres ressources lexicales. *Langages*, 3-4, 193-220.
- Hanks, P., Pustejovsky J. (2005). A pattern dictionary for natural language processing. *Revue Française de Linguistique Appliquée*, X-2, 63-82.
- Helbig, G., Schenkel, W. (1968). *Wörterbuch zur Valenz und Distribution Deutscher Verben*, Leipzig: Enzyklopädie.
- Herbst, Th., Heath, D., Roe, I. F., Götz, D. (2004). *A Valency Dictionary of English. A Corpus-based Analysis of the Complementation Patterns of English Verbs, Nouns, and Adjectives*. Berlin – New York: Mouton de Gruyter.
- Ionescu, A., Steriu, M. 1999. *Verbul românesc. Dicționar sintactic*. Editura Universității din București.

- Jezek, E., Magnini, B., Feltracco, A., Bianchini, A., Popescu, O. (2014). T-PAS: A resource of Typed Predicate Argument Structures for linguistic analysis and semantic processing. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, May 26-31, Reykjavik, Iceland, 890-895.
- Johnson, C., Fillmore, C. (2000). The FrameNet tagset for frame-semantic and syntactic coding of predicate-argument structure. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics (ANLP-NAACL 2000)*, April 29 - May 04, Seattle, Washington, 56-62.
- Kipper, K., Dang, H. T., Palmer, M. (2000). Class-based construction of a verb lexicon. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, July 30 - August 3, Austin, Texas, 691-696.
- Lenci, A., Lapesa, G., Bonansinga, G. (2012). LexIt: A Computational Resource on Italian Argument Structure. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, May 23-25, Istanbul, 3712-3718.
- Levin, B. (1993). *English Verb Classes and Alternations. A Preliminary Investigation*. Chicago and London: The University of Chicago Press.
- Mikelić Preradović, N., Boras, D., Kisicek, S. (2009). CROVALLEX: Croatian verb valence lexicon. In *Proceedings of the 31st International Conference on Information Technology Interfaces*, June 22-25, Cavtat /Dubrovnik, Croatia, 533-538.
- Przepiórkowski, A., Hajnicz, E., Patejuk, A., Woliński, M., Skwarski, F. & Świdziński M. (2014). Walenty: Towards a comprehensive valence dictionary of Polish. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, May 26-31, Reykjavik, Iceland, 2785–2792.
- Schumacher, H., Kubczak, J., Schmidt, R., de Ruiter, V. (2004). *VALBU – Valenzwörterbuch deutscher Verben*, Tübingen: Narr.
- Schneider, R. (2008). E-VALBU: Advanced SQL/XML processing of dictionary data using an object-relational XML database. In *Sprache und Datenverarbeitung, International Journal for Language Data Processing*, 1, 33-44.
- Žabokrtský, Z., Lopatkova, M. (2007). Valency Information in VALLEX 2.0: Logical structure of the lexicon. In *The Prague Bulletin of Mathematical Linguistics*, 87, 41-60.

NONSTANDARD VS. RRT: COMPARISON OF TWO ROMANIAN CORPORA IN UD

CĂTĂLINA MĂRĂNDUC^{1,2}, VICTORIA BOBICEV³

¹*Faculty of Computer Science, “Alexandru-Ioan Cuza” University, Iași*

²*“Iorgu Iordan, Al. Rosetti” Institute of Linguistics, Bucharest,
catalinamaranduc@gmail.com*

³*Technical University of Moldova, Chișinău, Republic of Moldova,
victoria.bobicev@ia.utm.md*

Abstract

The UD-Romanian treebank was introduced in the Universal Dependencies group in 2015 by the Institute of Artificial Intelligence of Bucharest (RACAI), with the contribution of the Treebank created at the Faculty of Computer Science, Al. I. Cuza University of Iași (UAIC). This Treebank contains 9,524 sentences in Romanian Contemporary Standard language. The UAIC treebank is balanced; after 2015, the annotation of nonstandard language styles (Old, Popular-Regional, Chat) was developed. In order to capitalize on these resources, in November 2017 another treebank, called UD-Romanian-Nonstandard was created. For this purpose, two tools have been created to transpose the treebank from the quite different annotation conventions of the UAIC Treebank into the UD conventions, and the XML format into the CONLLU one. A new working interface that supports multiple conventions and formats has been created. The nonstandard treebank was increased from 1,200 to 10,069 sentences, and the automated statistics produced by the UD site tools displayed very interesting conclusions about the annotation differences, but especially about the differences between standard and nonstandard Romanian, which linguists will study.

Key words — conllu format, dependency grammar, nonstandard language, POS-tagging, standard language, syntactic parser.

1. Introduction

UAIC-RoDiaDepTb is a resource recognized by the International Standard Language Resources site¹. It is a Dependency Treebank for Diachronic Romanian created at “Al. I. Cuza” University, Faculty of Computer Science, Iași, Romania.

It is also a balanced treebank in use, having now 23,534 sentences and 462,502 tokens; punctuation included, automatically annotated both morphologically and syntactically, entirely manually checked. The syntactic annotation conventions, founded in 2007, are quite different from the UD ones. 4,000 sentences were transposed into the UD conventions in 2015, to contribute to the formation of the UD_Romanian treebank (Mititelu *et al.*, 2015), which contains only Contemporary Standard Romanian and is currently called Romanian Reference Treebank (RRT) (Mititelu *et al.*, 2016).

In 2017, because the UD affiliated treebanks were more and more varied and not all contained the contemporary standard style, another treebank has been started, called UD_Romanian-

¹ <http://www.islrn.org/>

Nonstandard and containing seventeenth-century and folk texts, some of which came from the Republic of Moldova (Măranduc and Bobicev, 2017).

Language observing academical orthographic and morphologic standards is used only in publications, public occasions or in education. Neither spoken, familiar, or social media, nor poetry, nor popular language observes the academic rules, being creative styles of the language. Furthermore, the old language did not follow any of the current rules, while the regional language has rules specific to each geographical area. If we do not study these natural language facts, we will not be able to understand the overall dynamics of Romanian. If we do not train the processing tools on such language facts, they will not face the confrontation with natural language creativity.

Participation in UD makes possible comparative studies by aligning the Alba Iulia New Testament, first printed in Romania in 1648, with other Old Testament translations from PROIEL (Haug, 2014), a UD affiliate project. Also, it will be possible to preserve folk poetry, which is another type of cultural heritage (Bobicev *et al.*, 2016), and to compare the spoken language in Romania with the one of the Republic of Moldova.

Romanian treebanks have now access to various very useful tools for validating and correcting errors. But we have mostly benefitted from the output of the automatic statistical comparison between the two Romanian treebanks². This statistic does highlight some errors. Nevertheless, what is most important is that it highlights the differences between the current language according to academic norms and the language in old or popular and regional texts.

For more accurate annotation, some specific conventions are used in the nonstandard treebank, other than those used in the RRT treebank. Studies about these differences, that statistics make possible, are just starting out and will have more data to interpret while treebanks grow. The condition for this to happen is that in the first Romanian Treebank will continue the uploading of only contemporary language (The Institute of Artificial Intelligence in Bucharest being interested in scientific texts and Romanian laws) while, in the second treebank, only the nonstandard (Old, Folk, Regional, Social Media) texts will be uploaded.

2. Related Work

In the field of dependency treebanks, the reference remains the Pennsylvania Treebank, which is also the basis of the conventions adopted by the group of 151 treebanks for 72 languages affiliated to the Universal Dependencies (UD). The Stanford research regarding the tools for the morphologic syntactic annotation, focusing on cross-lingual studies (Marneffe *et al.*, 2014) are well known. The annotation in this type of grammars is based on words, and not on multiword structures (Marneffe and Manning, 2008).

The Prague Dependency Treebank is another important member of the UD group (Hajič *et al.*, 2017), and it participates in solving many linguistic or computational problems (Zeman, 2008).

The UD group, whose initial aim was to create a universal syntactic parser trained on all families of languages, is now growing and in addition to developing more and better tools for

² <http://universaldependencies.org/treebanks/ro-comparison.html>

morphological and syntactic annotation and for error detection, is especially interested in linguistic and computational comparisons among affiliated languages (Nivre *et al.*, 2016).

Tools for processing texts in all the 72 languages affiliated continues to be one of the main concerns for the members. Word segmentation (Borschinger and Johnson, 2011), morphological annotation and syntactic or semantic parsing (Rasooli and Tetreault, 2015, Simov and Osenova, 2011) are the most important subjects of the studies that this group generates. As far as morphology is concerned, it starts from universal categories (Petrov *et al.* 2012), and it comes to detailed annotations, e.g., for the rich inflexion East European languages (Erjavec, 2012).

Many affiliated treebanks have developed parallel annotations in several annotation conventions and formats, in order to add semantic or other types of information (Bóhmová *et al.*, 2003, Osenova and Simov, 2011, Măranduc *et al.*, 2018, Sgall *et al.*, 1986).

3. Tools for Old and Regional Romanian processing

Creating tools for processing old and regional languages is a difficult task. These language variants have peculiarities and inconsistencies different from those of standard languages. It is difficult to obtain a large corpus of such texts for the training of these tools. Such studies are recent; they date back to the last decade.

The method which we used was to process old texts with tools for annotating contemporary Romanian language, and then to manually correct the output, which contained many errors. This way a training corpus for Old Romanian was obtained. An instrument was created to automatically add into the POS-tagger lexicon, all the new combinations *word form – lemma – analysis* contained in the manually corrected corpus. This lexicon was also completed with lexical variants extracted from the Thesaurus Dictionary of the Romanian Language. Data extraction from specialized dictionaries and from the indexes of old books issued in critical editions was done using a tool called DEPAR (Dictionaries Parser) (Măranduc *et al.*, 2017b). The improved accuracy of the OldRo-POS-tagger was 91.66% (Măranduc *et al.* 2017c).

Uploading of the Alba Iulia New Testament (1648), printed in Old Romanian Cyrillic letters, required a special optical character recognizer built in Chişinău (Colesnicov *et al.*, 2016, Cojocaru *et al.*, 2017). Letters-digits, overwritten letters, abbreviations, inexistent hyphen, joint words are some of the difficulties encountered (Malahov *et al.*, 2017).

A parser was also trained on the contemporary treebank adding a small number of old texts at the beginning (Bobicev *et al.*, 2017). Experiments showed that quantity was more important than sorting the types of texts to increase the accuracy. The accuracy registered on 1st of July 2017 was 0.8201% (label precision), 0.8403% (head precision) and 0.7503% (overall precision), after the training on 20,784 sentences in various styles.

4. Tools for the Conversion into UD

The conventions of the UAIC treebank (established in 2007 and intended also for didactic use) are quite different from the UD ones. UD annotation convention highlights words with full meaning and the relational words are subordinated to them. In the UAIC convention, the

types of relations are highlighted, and the related words are heads, being placed between the head and the subordinated word in the tree.

In the UAIC format there are 14 kinds of circumstantial modifiers, which is a rich source of semantic information that we cannot overlook. A specific type of annotation, called logic-semantic, has been created to preserve the semantic information, already annotated in the UAIC format, and increase it. In order to use as a pivot, the basic annotation convention, automatically processed and fully supervised, is now transformed into two new conventions, the UD and the semantic ones.

For this, a rule-based transformation tool, called TREEOPS, has been created, with two different sets of such transformation rules for the two conventions. The transformation tool works in the XML format. Rules for the Semantic transformation does not transform all syntactic relations into semantics, but leaves unmodified the semantically ambiguous ones, so that a human annotator or a statistical parser needs to intervene later. Rules for the UD transformation apply to all relationships. The result is supervised. By comparing the output with the corrected version using MaltEval_Viz program, it results that TREEOPS has an accuracy of 78.9% for the UAIC-UD transformation.

Rules are in the form: *if* (condition) *then* (action) . There are three types of rules:

- Rules with a single condition;
- Rules with more conditions (and more actions);
- Rules for structural transformations, which, for example, reverse the direction of the relation between the noun and the preposition case marker.

There is also a list of other rules included in the program for establishing the order of these rules' application. Overall, there are more than 150 rules for the transformation into the UD convention (Colhon *et al.* 2017).

For example, the transformation of a circumstantial modifier is a rule of the first type in the conversion of UAIC into SEMANTIC:

```
//word[@deprel='c.c.l.']/@deprel => changeAttrValue('LOC')
```

For the transformation of a circumstantial in the UD format there are three rules for each circumstance. The rules are of the type with two or more conditions; a dependency relation and a morphological analysis:

```
//word[@deprel='c.c.l.' and starts-with(@postag, 'R')]/@deprel
=> changeAttrValue('advmod')
//word[@deprel='c.c.l.' and starts-with(@postag, 'V')]/@deprel
=> changeAttrValue('advcl')
//word[@deprel='c.c.l.' and (starts-with(@postag, 'N') or
starts-with(@postag, 'P') or
starts-with(@postag, 'M'))]/@deprel
=> changeAttrValue('obl')
```

Errors are often caused by small morphological annotation errors in the input document. Otherwise, errors are caused by unexpected exceptions commonly encountered in natural

language, especially in the nonstandard one. Therefore, the supervision of TREEOPS output is necessary, although it is not as time-consuming as the morphological and syntactic supervision of the POS-tagger and parser output.

Because TREEOPS works only with XML, after the supervision of its output, the UD XML format is automatically transformed into the CONLLU format by the program called "XML2CONLLU".

The program is a Python script that receives as input (1) an XML document where the relationships and structural transformations of the UAIC-UD are already made; (2) a document that contains morphological equivalences between the "postag" feature of this XML with the content that must appear on columns 4-5-6 in the CoNLLU; (3) the name and the place for the resulting CONLLU document.

The program must make the following changes:

- Write the information found in XML features in the form of columns in CoNLLU;
- Change the *postag*="..." with the result found in the document on the second line of the input, to columns 4-5-6;
- Renumber the sentences;
- Write the text of the analysed sentence;
- Write the *citation-part*="MATT 19.28" (from the sentence features) in the form: #*citation-part*=MATT 19.28;
- Expand it in the form: *ref*=MATT19.28 in the 10-th column of each word of the sentence.
- The annotation |SpaceAfter=No is automatically added in the 10-th column.

In order to be able to supervise corpora in a number of conventions and formats, we had to create a working interface that accepts all of them. As usually, these frameworks are specially created for a particular format and convention. Each treebank has its own framework, and the UAIC-RoDiaDep Treebank has one that supports only the basic convention and the XML format.

The new framework is the Treebank Annotator (Măranduc *et al.*, 2017a). This tool has drop-down lists from the morphological and syntactic tags. Because the treebank will be available in three formats, each of them has a Configuration that must be chosen before opening the document, the drop-down lists are different for each format.

Because the UD format is not XML, but CONLLU, the Treebank Annotator was set to open, display, change and save documents in both XML and CONLLU formats.

All the features of each configuration are displayed by the framework and can/cannot be modified (according to the configuration). Also, the tool allows the introduction of words in text or deletion of words, renumbering ids of words and of heads, without changing the relationships in the tree.

For adding a word, a table with features must be completed (i.e. word form, lemma, POS-tag, head and dependency relation), then another table with the word order is displayed and the user must move the new word to the place where it is needed. For example, in UD we must separate the compound prepositions. See Fig. 1.

Name	Value
Id	20
Form	la
Lemma	la
Part of speech tag	Spsa
Head word id	de la
Chunk	
Dependency relation	fixed

Figure 1. The table for the add word function.

After the add-word, we must delete the second part of the compound preposition, in the word-form field. The UD convention does not allow compounds in the form field.

Another function is *Compare sentences*.

It displays a table with two columns. In each column another document can be selected. After the selection of 1 sentence for each column, generally with the same text in two conventions, it displays a linear graph. See Fig. 2.

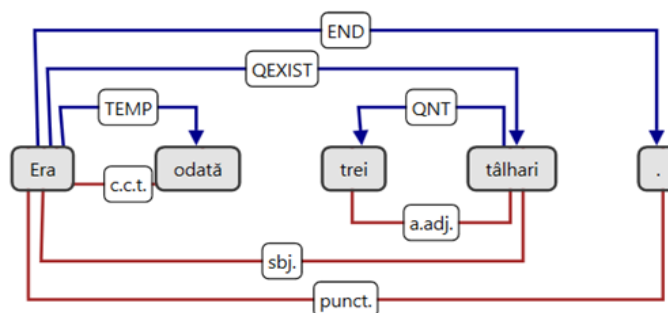


Figure 2: The sentence: *There were three thieves once* in SEMANTIC and UAIC conventions.

Now, we decided to add two new functions: the formation of a sentence from two existing sentences and the formation of two sentences from one existing sentence. These functions are necessary because often the split function of the POS-tagger does not display identical results as the citation part (the number of the chapter and of the verse in each translation of the New Testament in the entire world).

For example, in the sentence „Închină-să voao Aristarh, soțul meu în prinsoare, și Marco.” (*Salute you Aristarh, my fellow prisoner, and Marco.*), the period should be a comma.

The following sentence is: „nepotul Varnavei, despre carele am poruncit voao, să va veni cătră voi, primiți-l.” (*the Varnava’s nephew, whom I have commanded you, if he comes to you, receive it.*). The two sentences must be merged, the result can be seen in Fig. 3.

5. Particularities of the Old Romanian

Romanian linguists and computational linguists can benefit from the fact that the Universal Dependencies site has programs that make possible comparative statistics between treebanks, especially between those for the same language. The statistic displays only the peculiarities of the nonstandard Romanian as they are revealed in the treebank contributed to the UD at this time; perhaps other peculiarities will emerge in the future when the corpus will be increased. This parallel is very important to maintain, the contemporary texts of the UAIC treebank can be uploaded only in the RRT treebank.

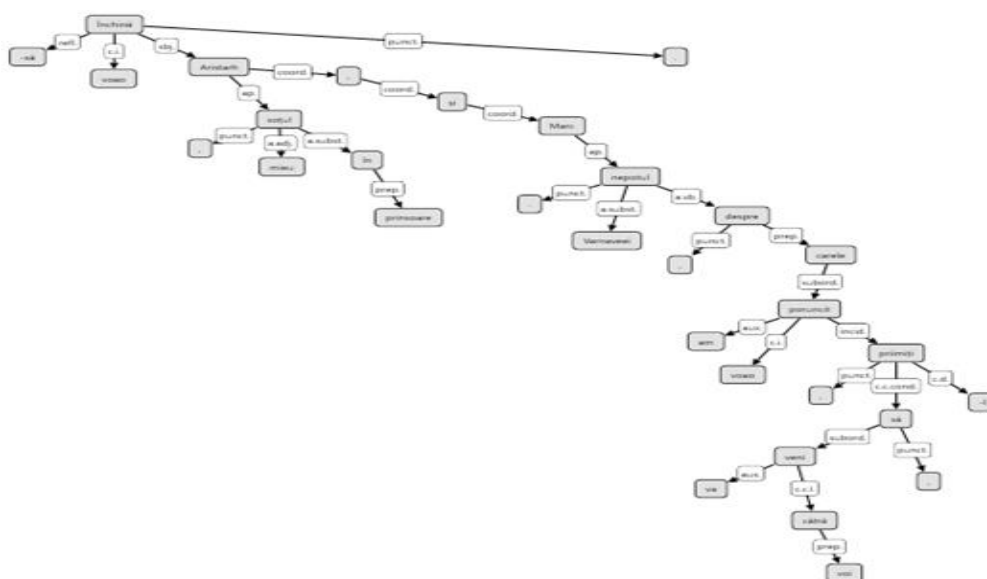


Figure 3: The two sentences in the example above are merged.

At this time, the nonstandard treebank consists of only 1,138 sentences in the folk style and the remaining 8,931 are New Testament (1648) sentences. Popular language is generally more conservative than the standard academic language, so we can say that statistics show predominantly peculiarities of the old language.

First observation: The number of *SpaceAfter=No* annotations, intended for the correct reading of the text by the computer, is also interesting for researchers. It is much higher in nonstandard language occurrences (higher by 5%). Here are many topical inversions that are marked by a dash. Examples:

Example 1:

rogu-te = te rog (I please you), mântui-mă-voi = mă voi mântui (I will save myself), Trimetu-să apostolii = apostolii se trimit (the apostles are sent), da-ți-voi ție = îți voi da ție (I will give you), etc.

In examples of this type, there is no dash in the contemporary language (see the form after the =). Romanian is a language with free word order, and Old Romanian is even more a language with free word order than contemporary Romanian. The hyphen is a mark of the inversion of the word order.

Brackets are frequently used by the translator to give an alternative word or an explanation. The quotes are also used for marking the text that is cited and commented. Examples:

Example 2:

Un om purtând un vas (Urcior)= a man carrying a vessel (jug).

Auzit-ați că Eu am zis voao: "mearge-voiu și voiu veni către voi" You have heard that I have said to you: "I will go and come to you".

Another observation to be made is that the Gospels contain many dialogues, in direct speech. We found in the four Gospels the following number of occurrences for the verbs dicendi: lemma="zice" (say): 1,711, lemma="spune" (say): 78, lemma="întreba" (ask): 101, lemma="răspunde" (answer): 225, lemma="grăi" (speak): 261, lemma="striga" (shout): 62. Folk verses have an oral style too, and are richly dialogical. The topical inversion (marked by a hyphen) is used in the interrogative clauses.

In the morphological section of the statistic, the Ro_nonstandard has 66 lemmas tagged as pronouns, and the ones cited below are old and popular forms never (or rarely) used in the Contemporary Romanian: *cineși, fiecine, fiește, fieștecare, fieștecine, mineși, măria(-sa), neștine, nimicuța, oarece, oarecine, oarecât, sfinția(-sa), sineși, voiși, însul, însumi, însuși, însuți, ălalalt, ista*. These are difficult to translate, (Whoever, Everyone, Myself, our majesty, Somehow, Someone, our Sanctity, You self, Himself, the Other, That). The forms *însumi, însuși, însuți* are used today only as determiners and not as pronouns. If the word form of pronouns is considered, and not only lemmas, the number of special lexical peculiarities of the nonstandard language will be stronger.

Another peculiarity of UD_ro-nonstandard treebank is the existence of the "Definite=Yes" feature of some pronouns. Examples: *carele, carii, carea, caria, careleși, carile, fietecarele*. English: *which*.

Another difference between nonstandard and standard Romanian treebanks is the verbal form in which the auxiliaries are used. The auxiliary 'vrea' is used in the contemporary language, as a future tense formant. But it is not used in the verbal participle form, which has in the old language the function of constructing the past of the conditional mood. Example: "Tu ai vrut ceare de la El, și ți-au vrut da ție apă vie". (You would have asked of Him, and He would have given you living water).

6. Differences of annotation

In the UD_nonstandard treebank, there are six auxiliary verbs: "avea" (have), "fi" (be), "putea" (can), "trebui" (must), "vrea" (want), "ști" (know). Examples:

Example 3:

"Și aduș pre el ucenicilor Tăi și nu-l putură vindeca". (And I brought him to your disciples, and they could not heal him).

"Preuți cărturari și oameni înțelepți, carii să știe izvodi Testamentul cel nou" (Learned priests and wise men, who know how (to) write the New Testament).

"Moșia trebe cosită, Găina trebe-mpărțită". (The field must (be) mown, the hen must (be) divided).

The "Polarity=Neg" is much more richly represented in our annotation system. In the UD_ro-RRT treebank, there are only two negative polarity annotations: "ADV: ne, non, non-" and "PART: nu, n-". In the UD_ro-nonstandard treebank, there are the following "Polarity=Neg" annotations (with the letter "z" contained in the X-POS-tag, the Romanian POS-tag):

- ADV: nu, n-, nu-, nece, neceodată, nice, n, Nececum, Nicecum (Rz);
- AUX-Ger: neputîndu, nefiînd (Vag-----z);
- CCONJ: nece, nici, neci, nice (Cccsz);
- VERB-Fin: face, duce, speria, umbla, gândi, dormi (imperative=Vmm-2s--z);
- VERB-Ger: neavînd, neştiind, neaflînd, necrezînd, nefăcînd (Vmg-----z);
- VERB-Part: nespălate, nestinsu, necunoscut, netăiaţi, neînsurat, neştiut, nebotezat, necunutat, necusută, nedespărţitei (Vmp--sm-z, Vmp--sf-z--r, Vmp--sf-z--o, Vmp--pm-z, Vmp--pf-z).

Another difference between the annotation system of two treebanks is the Variant=Long for the infinitive and plural imperative, specific of Old Romanian. Examples:

Example 4:

AUX-Inf: fire (Van-----l) Ex: ca să înşale şi pre cei aleşi de-a fire putea (to also deceive pre-elected ones, if he could);

VERB-Fin (imperative negative form): teamereţi, grijireţi, giudecareţi, contenireţi, oprireţi, plîngereţi, spămîntareţi, adunareţi, bucurareţi, creadereţi (Vmm--2p---l);

VERB-Inf: cearere, ertare, perire-, slujirea, săvîrşirea, treacere, ziceare (Vmn-----l).

In the other treebank, there are lots of examples annotated as Variant=Short, words with hyphen. It is about the annotation of the hyphen word forms with y, as +clitic. Examples (from the list of tags used by the RACAI's POS-tagger):

Example 5:

"Afpfpoy" = "Adj. q. positive fem. pl. obl. +def." (or without def. and +clitic?)

"Afpfpoyy" = "Adj. q. positive fem. pl. obl. +def. +clitic"

In the UAIC treebank this annotation was removed, because it is confusing with y showing the definite article (Mărănduc *et al.*, 2016), and because, as we pointed out above, in the nonstandard language there is a very large, virtually infinite number of such forms, which we consider variants of the word (A lexical variant may be shorter or longer than the lemma, and may occur for various reasons, including pronunciation with something else).

7. Conclusions and future work

The intention of our team is to develop the Old Romanian section, including quotes and texts illustrating the sixteenth to nineteenth centuries, and to compare the syntactic and morphologic peculiarities of each period. The sections Folk poetry and Chat will be also increased.

We intend to align the entire text of the Alba Iulia New Testament (1648), the first one printed in Romanian, to similar corpora of the PROIEL project. A small part of the Acts of the Apostles is still work in progress. The variant with word form in Cyrillic letters can be uploaded in the UD if there is interest in this work. Aligning the entire text of the New Testament with the ones in the Ancient Greek, Latin and Slavonic ones will be of great importance to etymologists and students of ancient texts who will be able to elucidate the source of ancient sacred texts translations.

Developing the UAIC treebank is an increasingly important resource illustrating all the styles of Romanian, and what we have to do is not only to continue working on it, to develop new instruments on its basis, but also to highlight the resources and tools, to make them known and useful to other research.

References

- Barbu-Mititelu V., Măranduc, C., Irimia, E. (2015). Universal and Language-Specific Dependency Relations for Analysing Romanian. In *Proceedings of the Third International Conference on Dependency Linguistics, Depling*, Uppsala University, 28–37.
- Barbu-Mititelu, V., Ion, R., Simionescu, R., Irimia, E., Perez, C.-A. (2016). The Romanian Treebank Annotated According to Universal Dependencies. In *Proceedings of The Tenth International Conference on Natural Language Processing*, Croația.
- Bobicev, V., Bumbu, T., Lazu, V., Maxim, V., Istrati, D. (2016). Folk Poetry for Computers: Moldovan Codri's Ballads Parsing. In *Proceedings of ConsILR*, Iași, University Press, 39–50.
- Bobicev, V., Măranduc, C., Perez, C.-A. (2017). Tools for Building a Corpus to Study the Historical and Geographical Variation of the Romanian Language. In *Proceedings of Language Technology for Digital Humanities in Central and (South-) Eastern Europe*, Varna, Bulgaria, 10–20.
- Bóhmová, A., Hajič, J., Hajičová, E., Hladká, B. (2003). Prague Dependency Treebank: A Three-Level Annotation Scenario. In *Text, Speech and Language Technology*, Prague, Springer, 103–127.
- Borschinger B., Johnson, M. (2011). A Particle Filter Algorithm for Bayesian Word Segmentation. In *Proceedings of the Australian Language Technology Association Workshop*, Canberra, Australia, 10–18.
- Cojocaru, S., Colesnicov, A., Malahov, L. (2017). Digitization of Old Romanian Texts Printed in the Cyrillic Script. In *Proceedings of Digital Access of Textual Cultural Heritage*, Göttingen, ACL Publisher, 143–148.
- Colesnicov, A., Malahov, L., Bumbu, T. (2016). Digitization of Romanian Printed Texts of the 17th Century. In *Proceedings of ConsILR*, Iași, University Press, 1–11.
- Colhon, M., Măranduc, C., Mititelu, C. (2017). A Multiform Balanced Dependency Treebank for Romanian. In *Proceedings of Knowledge Resources for the Socio-Economic Sciences and Humanities*, Varna, Bulgaria, 9–18.
- Erjavek, T. (2012). Multext-East: Morphosyntactic Resources for Central and Eastern European Languages. *Language Resources and Evaluation*, 46(1):131–142.
- Hajič, J., Hajičová, E., Mikulová, M., Mirovský, J. (2017). Prague Dependency Treebank. In *Handbook of Linguistic Annotation*, Springer, 555–594.
- Haug, D.T.T. (2014). The PROIEL Corpus: Annotation of Morphology, Syntax and Information Structure, Perspective Project Kick-off Meeting. University of Nijmegen.
- Malahov, L., Măranduc, C., Colesnicov, A. (2017). A Diachronic Corpus for Romanian (RoDia). In *Proceeding of Language Technology for Digital Humanities in Central and (South-) Eastern Europe*, Varna, Bulgaria, 1–10.

COMPARISON OF TWO ROMANIAN CORPORA IN UD

- Mărănduc, C., Malahov, L., Perez, C.-A., Colesnicov, A. (2016). Rodia Project of a Regional and Historical Corpus for Romanian. In *Proceedings of MFOI*, Chişinău, 268–284.
- Mărănduc, C., Bobicev, V. (2017). Nonstandard Treebank Romania-Republic of Moldova in the UD. In *Proceedings of MFOI*, Chişinău, 111–116.
- Mărănduc, C., Hociung, F., Bobicev, V. (2017a). Treebank Annotator for Multiple Formats and Conventions. In *Proceedings of The 4th Conference of Mathematical and Computer Science Society*, Chişinău, 529–534.
- Mărănduc, C., Mititelu, C., Bobicev, V. (2018). Syntactic Semantic Correspondence in Dependency Grammar. In *Proceedings of 16th International Workshop on Treebanks and Linguistic Theories*, Prague, 167–180.
- Mărănduc, C. Mititelu, C., Simionescu, R. (2017b). Parsing Romanian Specialized Dictionaries Structured in Nests. In *Proceedings of Digital Access to Textual Cultural Heritage*, Göttingen, ACL Publisher, 35–41.
- Mărănduc, C., Simionescu, R., Cristea, D. (2017c). Hybrid Pos-tagger for Old Romanian. Presented at the International Conference on Computational Linguistics and Intelligent Text Processing (CICLing), Budapest.
- Marneffe, M.-C. de, Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., Manning, C. (2014). Universal Stanford Dependencies: A Cross-Linguistic Typology. In *Proceedings of LREC*.
- Marneffe M.-C. de, Manning, C. (2008). The Stanford Typed Dependencies Representation. In *COLING Workshop on Cross Framework and Cross-Domain Parser Evaluation*.
- Nivre, J., Marneffe M.-C. de, Ginter, F., Goldberg, Y., Hajič, J., Manning, C., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., Zeman, D. (2016). Universal dependencies v1: A multilingual treebank collection. In *Proceedings of LREC*.
- Osenova, P., Simov, K. (2011). Syntactic Semantic Treebank for Domain Ontology Creation. In *Cognitive Studies*, Warsaw, SOW Publishing House, 213–225.
- Petrov, S., Das, D., McDonald D. (2012). A Universal Part-of-Speech Tagset. In *Proceedings of LREC*.
- Rasooli M.-S., Tetreault, J.-R. (2015). Yara parser: A Fast and Accurate Dependency Parser. *Computing Research Repository*, ArXiv:1503.06733.
- Sgall, P., Hajičová, E., Panevová, J. (1986). *The Meaning of the Sentence in its Semantic and Pragmatic Aspects*. Academic Press and Reide, Prague-Dordrecht.
- Simov, K., Osenova, P. (2011). Towards Minimal Recursion Semantics over Bulgarian Dependency Parsing. In *Proceedings of the RANLP Conference*. Hissar, Bulgaria.
- Zeman, D. (2008). Reusable Tagset Conversion Using Tagset Drivers. In *Proceedings of LREC*.

INTERLINKING AND EXTENDING LARGE LEXICAL RESOURCES FOR ROMANIAN

MIHAI ALEX MORUZ^{1,2}, ANDREI SCUTELNICU^{1,2}, DAN CRISTEA^{1,2}

¹*Faculty of Computer Science, “Alexandru Ioan Cuza” University of Iași*

²*Institute for Computer Science, Romanian Academy, Iași branch
{mmoruz, andreis, dcristea}@info.uaic.ro*

Abstract

CoRoLa – the Representative Corpus of the Romanian language – was developed during a period of 4 years, and is available since 2017. At this moment it contains almost one billion Romanian words, covering a wide variety of domains (politics, humanities, theology, sciences, etc.) and all literary genres (fiction, poetry, theatre plays, scientific texts, journalistic texts, etc.). The raw texts are tokenized, lemmatized, and morphologically processed. This paper describes two main research directions for the extension and improvement of CoRoLa. The first issue discussed is the adding of new layers of annotation, such as noun and verb groups, syntactic trees, and some semantic information. Secondly, we describe an interoperability standard for textual and lexical resources for Romanian (such as a wordnet and a dictionary) which opens the way for an efficient coupling of these resources, with the benefit of a much more diverse display about the use of the Romanian language.

Key words — CoRoLa, lexical resources, wordnet, electronic dictionary, textual linked data

1. Introduction

Linguists around the world are deeply preoccupied to build linguistic resources that best describe languages and their structure, starting with simple collections of scanned documents, as in virtual libraries, to text collections organised as corpora, dictionaries and thesauri. Linguistic resources are the primary ingredients fuelling the constant and relentless concern of researchers to study languages as they are or in their evolution, as well as to facilitate the automation of language processing technologies, so much needed today by the web and mobile industries.

Isolated, each linguistic resource was thought to fulfil some specific needs, but, very recently, the benefits of exploiting them interlinked has become evident (McCrae *et al.*, 2011; Navigli and Ponzetto, 2012). For instance, to name only the domain of language learning, lexical resources are very important tools, structured in many ways: word lists, corpora, dictionary entries, wordnet synsets, language models, etc. In this study, we will concentrate on three types of resources, each with a different structure, showing a proposal of interlinking them and some experiments:

- RoWN – The Romanian WordNet (Tufiş and Cristea, 2002; Tufiş *et al.*, 2013) is created following the model of the Princeton WordNet (Fellbaum, 1998), which can be considered a real revolution in computational linguistics through the scientific advances it has generated in recent years. It is a collection of nouns, verbs, adjectives, and adverbs placed in a graph, where the nodes, called synsets, are sets of literals of

the same syntactic category for which there are contexts in which they can be considered synonyms. The literals, members of synsets, are paired with indexes denoting specific senses and one literal-sense pair can be member of only one synset. Thus, each synset expresses a distinct linguistic concept and it can be seen in semantic relations with other synsets of the graph. For instance, synsets of nouns and verbs are placed in a conceptual hierarchy through hypernymy and/or hyponymy relationships. As shown in Figure 1, RoWordNet synsets have unique identifiers, which point to synsets of the English WordNet, considered the backbone of a whole network of different languages wordnets.

```
<SYNSET>
  <ID>ENG30-00027167-n</ID>
  <POS>n</POS>
  <SYNONYM>
    <LITERAL>loc<SENSE>1</SENSE></LITERAL>
  </SYNONYM>
  <DEF>Punct, porțiune determinată în spațiu.</DEF>
  <STAMP>Dan Cristea</STAMP>
  <ILR>ENG30-00002684-n<TYPE>hypernym</TYPE></ILR>
  <ILR>ENG30-00028651-n<TYPE>part_holonym</TYPE></ILR>
  <ILR>ENG30-02694933-v<TYPE>near_eng_derivat</TYPE></ILR>
  <ILR>ENG30-02333689-v<TYPE>near_eng_derivat</TYPE></ILR>
  <ILR>ENG30-02286204-v<TYPE>near_eng_derivat</TYPE></ILR>
  <ILR>ENG30-00413876-v<TYPE>near_eng_derivat</TYPE></ILR>
  <ILR>ENG30-08489497-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08489627-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08489765-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08490199-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08490402-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08500433-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08562067-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08593262-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08620061-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08630039-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08630985-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08683383-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-08795880-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-09386842-n<TYPE>hyponym</TYPE></ILR>
  <ILR>ENG30-13910384-n<TYPE>hyponym</TYPE></ILR>
  <DOMAIN>factotum</DOMAIN>
  <SUMO>Region<TYPE>=</TYPE></SUMO>
  <SENTIWN><P>0</P><N>0</N><O>1</O></SENTIWN>
</SYNSET>
```

Figure 1: A RoWN entry

- CoRoLa (the CoRpus of contemporary ROmanian LAnguage) is a resource completed at the end of 2017, as a result of a team work of approximately 4 years, contributed by two institutes of the Romanian Academy: the Research Institute for Artificial Intelligence in Bucharest and the Institute of Computer Science in Iași. At this moment, CoRoLa includes texts that sum up almost 1 billion Romanian words from a wide range of fields (politics, humanities, theology, sciences, arts, etc.) and covers all genres (literature, poetry, theatre, scientific texts, journalism, blogs) (Bibiri *et al.*, 2015). The texts were obtained, on the basis of written protocols, from their IPR owners (publishing houses, bloggers, independent writers), and were passed through a processing chain (partially manual, partially automatic) along which the UTF-8 cleaned raw texts were extracted, metadata was filled in and an XML annotation that includes tokens, lemmas and part of speeches was added (Tufiş and Ion, 2017).
- eDTLR, the electronic form of the Thesaurus Dictionary of the Romanian Language (Cristea *et al.*, 2011), is the searchable digital form of the largest and most complete

Romanian dictionary, elaborated by the Romanian Academy in about one century and covering the Romanian language in diachronicity, from its first written origins to the contemporary usage. The entries of this resource are partitioned in a hierarchy of senses, from main to very specific. For each sense and sub-sense of the title word, its part of speech (POS), definition and examples of usage are selected from the approx. 4,000 bibliographical sources, in chronological order (see Figure 2).

All three described resources (RoWN, CoRoLa, eDTLR) are open freely for online access, but their XML formats are available on the basis of restricted licenses of use, which address precisely the community of (computational) linguists.

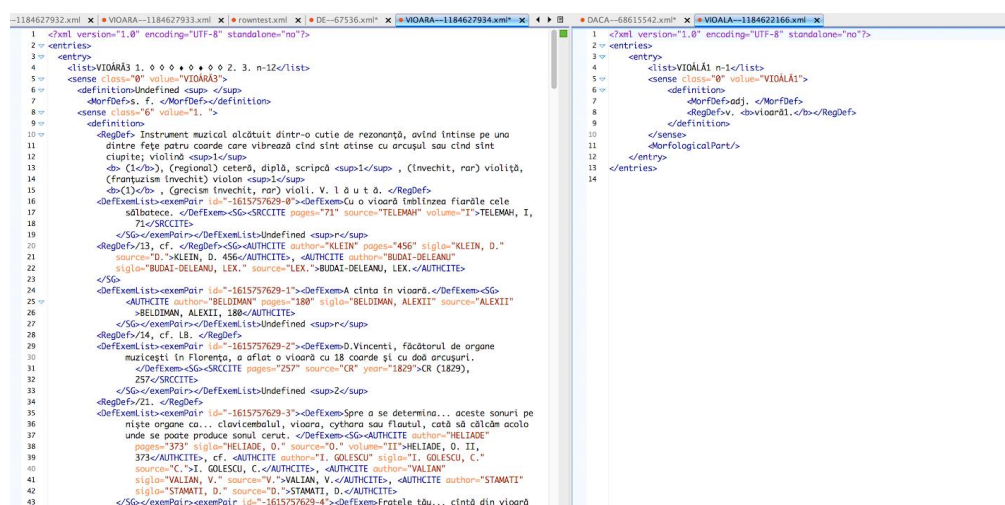


Figure 2: eDTLR dictionary entries

2. Indexing eDTLR and RoWN in CoRoLa

2.1. Pre-processing the resources

In eDTLR, the complexity of the entries vary widely, they do not have a predefined size, i.e. an exact number of definitions and examples, because the number of senses and sub-senses is very different. Fig. 2 displays extracts from the XML representation of two words, one with a very rich tree of senses, with many different definitions and with more citations for each sense, and one with a very simple structure, including only information about POS and synonyms.

Another problem encountered in extracting the indexing information from eDTLR was the presence of accent on exactly one character of each title word. The accents are intended to make explicit the correct pronunciation of the word (e.g. "VIOĂRĂ", with an accented "A"). Since accented characters have different codes than their corresponding un-accented ones, we doubled each original title word with a variant in which the accented letter was automatically replaced by the un-accented one. As such, eDTLR could be easily indexed at the title word level in view of allowing search operations over them, with the retrieved forms always being their accented variants.

Compared to eDTLR, both CoRoLa and RoWN have a compact structure, observing a well-defined template, which makes easier both information extraction and processing.

2.2. Correspondence files

A first step in indexing the resources was to extract the information relevant to their alignment. Thus, a first indexing is done at the input word and their corresponding part of speech levels. In order to index eDTLR and RoWN in CoRoLa, we first extracted all lemmas in the CoRoLa documents, and then we merged them, obtaining a list of unique pairs, containing lemma and part of speech. We have also extracted lists of title words in eDTLR, together with the attached morphological descriptions for the entry and a list containing RoWN literals, their part of speech and the ILI code identifying the synset the literals belong to.

Having these lists available, we have created a correspondence file as follows:

- match the lemma of a CoRoLa word to the entries in eDTLR and the literals in RoWN synsets;
- refine the match on the basis of the identified POS for the corpus word, the morphological definition for the eDTLR entry and the POS of the RoWN synset.

The information is written in a correspondence file that contains POS of the corpus word, the lemma, the eDTLR file name that contains the entry, and the corresponding words in RoWN, as can be seen in Fig. 3. Many lexical tokens in the corpus have correspondences in both eDTLR and RoWN, but there are also words that correspond only to one of the two resources or to none of them. This is because the three lexical resources do not cover identical time periods, showing once more that the language is in a continuous process of change, is a true "living organism" with a spectacular evolution.

```
<entry id="5">
  <CoRoLaPOS>NPN</CoRoLaPOS>
  <CoRoLaWord>aspect</CoRoLaWord>
  <DLRFileName id="1">ASPECT-1940125660.xml</DLRFileName>
  <SynsetId id="1">ENG30-13804669-n</SynsetId>
  <SynsetId id="2">ENG30-05850624-n</SynsetId>
  <SynsetId id="3">ENG30-05820170-n</SynsetId>
  <SynsetId id="4">ENG30-04677514-n</SynsetId>
  <SynsetId id="5">ENG30-04679419-n</SynsetId>
  <SynsetId id="6">ENG30-05064037-n</SynsetId>
  <SynsetId id="7">ENG30-04733118-n</SynsetId>
  <SynsetId id="8">ENG30-04674715-n</SynsetId>
  <SynsetId id="9">ENG30-05940302-n</SynsetId>
</entry>
<entry id="6">
  <CoRoLaPOS>S</CoRoLaPOS>
  <CoRoLaWord>de</CoRoLaWord>
  <DLRFileName id="1">DE--67536.xml</DLRFileName>
  <DLRFileName id="2">DE--67537.xml</DLRFileName>
</entry>
<entry id="7">
  <CoRoLaPOS>NSRN</CoRoLaPOS>
  <CoRoLaWord>alergologie</CoRoLaWord>
</entry>
<entry id="8">
  <CoRoLaPOS>CR</CoRoLaPOS>
  <CoRoLaWord>și</CoRoLaWord>
  <DLRFileName id="1">SI-10923.xml</DLRFileName>
</entry>
<entry id="9">
  <CoRoLaPOS>NSRN</CoRoLaPOS>
  <CoRoLaWord>imunologie</CoRoLaWord>
  <SynsetId id="1">ENG30-06051542-n</SynsetId>
</entry>
```

Figure 3: Correspondence file for the three resources

With this correspondence file available, in order to help researchers of the Romanian language, we have created a web interface in which the user can choose the desired associated

resources and which dynamically displays pairings. The left side window shows a fragment from a CoRoLa document file, while the right side window displays entries on one of the associated resources. When the user is hovering over a word in the left window, the interface brings in the right window an entry of the other two resources. The created links are based on the correspondences mentioned above. As stated already, not every word in the corpus has a connection with elements from the other resources assigned, which makes that only part of the left hand side words be active. Fig. 4 shows the functionality of the interface.

Currently, this interface can only input a CoRoLa file, which is matched against eDTLR and/or RoWN, but we intend to develop the application to allow for multiple file processing and for word searches in the CoRoLa corpus that generate text snippets related to that word (somehow similar to the KoRAP key-word-in-context interface).

A problem encountered in making the correspondence file is that the number of single lemmas is very large and the access time for this file is slow. We have tested the system over a set of one hundred thousand files, for which the total number of words exceeds three hundred million, of which about one million eight hundred thousand of unique lemmas has been identified. As a necessity in future, it would be ideal to make one correspondence file for each entry in the CoRoLa corpus, which would help optimize the search for connectivity between resources.



Figure 4: Web interface for correspondences between CoRoLa – eDTLR and CoRoLa – RoWN. Here it is exemplified a CoRoLa – eDTLR pairing

A second approach to indexing resources would be from the reverse perspective, namely, indexing the eDTLR and RoWN in the CoRoLa corpus.

The approach does not differ much from the previous indexing, namely, we keep the lists described in the previous step, only that now there will be two, one for each resource, as

correspondence files. Thus, for both eDTLR and RoWN we will do a similar verification, first matching head words and then refining with the POS.

The resulting information is written in a correspondence file containing the POS of the input word from eDTLR, the input word and the name of the input file containing the eDTLR entry, and, from the CoRoLa corpus, contexts related to the search word. This can also be seen in Fig. 5, where contexts of 9 words are displayed, i.e. the search element is surrounded by four words before and four words after. The length of the preceding and succeeding windows can be easily altered, although in most situations a 9-words window is sufficient for providing adequate context overview.

```
<?xml version="1.0" encoding="UTF-8"?>
<entry POS="n" content="mare" id="6">
  <eDTLR_FileName>MARE--76941771</eDTLR_FileName>
  <corpus>
    <entry id="1">
      <W Case="direct" LEMMA="munte" MSD="Ncmsry" POS="NOUN" id="52.9">muntele</W>
      <W Case="direct" LEMMA="hem" MSD="Ncmsrn" POS="NOUN" id="52.10">Hem</W>
      <W Case="" LEMMA="de-lângă" MSD="Sp" POS="ADPOSITION" id="52.11">de lângă</W>
      <W Case="direct" LEMMA="tărâm" MSD="Ncmsry" POS="NOUN" id="52.12">tărîmul</W>
      <W Case="oblique" LEMMA="mare" MSD="Ncfsoy" POS="NOUN" id="52.13">Mării</W>
      <W Case="oblique" LEMMA="negru" MSD="Afpfson" POS="ADJECTIVE" id="52.14">Negre</W>
      <W Case="" LEMMA="în-sus" MSD="Rg" POS="ADVERB" id="52.15">în sus</W>
      <W Case="" LEMMA="câtre" MSD="Sp" POS="ADPOSITION" id="52.16">câtre</W>
      <W Case="direct" LEMMA="apă" MSD="Ncfpry" POS="NOUN" id="52.17">apele</W>
    </entry>
    <entry id="2">
      <W Case="direct" LEMMA="etnic" MSD="Afpfprn" POS="ADJECTIVE" id="80.10">etnice</W>
      <W Case="direct" LEMMA="stabilit" MSD="Afpfprn" POS="ADJECTIVE" id="80.11">stabilitate</W>
      <W Case="" LEMMA="la" MSD="Sp" POS="ADPOSITION" id="80.12">la</W>
      <W Case="direct" LEMMA="tărâm" MSD="Ncmsry" POS="NOUN" id="80.13">tărîmul</W>
      <W Case="oblique" LEMMA="mare" MSD="Ncfsoy" POS="NOUN" id="80.14">Mării</W>
      <W Case="oblique" LEMMA="negru" MSD="Afpfson" POS="ADJECTIVE" id="80.15">Negre</W>
      <W Case="" LEMMA="pe" MSD="Sp" POS="ADPOSITION" id="80.16">pe</W>
      <W Case="oblique" LEMMA="Arimaei" MSD="Npfpon" POS="NOUN" id="80.17">Arimaei</W>
      <W Case="" LEMMA="/" MSD="SLASH" POS="" id="80.18">/</W>
    </entry>
    <entry id="3">
      <W Case="" LEMMA="(" MSD="LPAR" POS="" id="84.18">(</W>
      <W Case="" LEMMA="adică" MSD="Rg" POS="ADVERB" id="84.19">adică</W>
      <W Case="" LEMMA="în" MSD="Sp" POS="ADPOSITION" id="84.20">în</W>
      <W Case="direct" LEMMA="apropiere" MSD="Ncfsoy" POS="NOUN" id="84.21">apropierea</W>
      <W Case="oblique" LEMMA="mare" MSD="Ncfsoy" POS="NOUN" id="84.22">Mării</W>
      <W Case="oblique" LEMMA="negru" MSD="Afpfson" POS="ADJECTIVE" id="84.23">Negre</W>
      <W Case="" LEMMA=")" MSD="RPAR" POS="" id="84.24">)</W>
      <W Case="" LEMMA="," MSD="COMMA" POS="" id="84.25">,</W>
      <W Case="" LEMMA="locui" MSD="Vmip3p" POS="VERB" id="84.26">locuiesc</W>
    </entry>
    <entry id="4">
      <W Case="direct" LEMMA="stepă" MSD="Ncfpry" POS="NOUN" id="106.8">stepele</W>
      <W Case="direct" LEMMA="afat" MSD="Afpfprn" POS="ADJECTIVE" id="106.9">afate</W>
      <W Case="" LEMMA="la" MSD="Sp" POS="ADPOSITION" id="106.10">la</W>
      <W Case="direct" LEMMA="nord" MSD="Ncmsry" POS="NOUN" id="106.11">nordul</W>
      <W Case="oblique" LEMMA="mare" MSD="Ncfsoy" POS="NOUN" id="106.12">Mării</W>
      <W Case="oblique" LEMMA="negru" MSD="Afpfson" POS="ADJECTIVE" id="106.13">Negre</W>
      <W Case="" LEMMA="și" MSD="Cc" POS="CONJUNCTION" id="106.14">și</W>
      <W Case="direct" LEMMA="a" MSD="Tsfsr" POS="ARTICLE" id="106.15">a</W>
      <W Case="oblique" LEMMA="mare" MSD="Ncfsoy" POS="NOUN" id="106.16">Mării</W>
    </entry>
    <entry id="5">
```

Figure 5: Correspondence file between eDTLR and CoRoLa

A similar process was carried out for the RoWN to CoRoLa correspondence, but instead of the name of the eDTLR file containing the dictionary entry, RoWN ILI code for the corresponding synset was used. For this approach, we have not yet developed any user interface.

3. Indexing eDTLR in RoWN

In order to extend the coverage of the lexical resources mentioned (eDTLR and RoWN), we also investigated some ways in which to link the senses of words in the thesaurus to synsets in

RoWN. This encumbers a sense disambiguation problem and for that we try to match eDTLR glosses against RoWN senses.

The problem encountered in eDTLR sense glosses is the complexity of entries. Complexity is given by the variable number of senses and sub-senses of the input word, to which general and specific definitions with examples are added. As can be seen in Fig. 2, sense tree complexity varies largely, and it is this complexity which makes it difficult to analyse and extract the necessary information in the process of comparing definitions. We have extracted from eDTLR lists containing: the entry name, a list with all the definitions extracted from the *<RegDef>* tag and the POS. For the RoWN, a similar list was created for each synset. The match function used is as follows:

```
if edtlrList[POS] = rownList[POS] and edtlrList[entry] =
rownList[literal] then return the match between glosses by counting
the number of common words, normalized to [0, 1]
```

We ran this function on the three eDTLR entries for “VIOARĂ”, one of which is an adjective and the others nouns (the name of a plant and the musical instrument). In RoWN, the literal “VIOARĂ” is present twice, with the sense of musical instrument (one modern, and another archaic). In the first phase, applying the above described matching test, we only leave two of the three eDTLR entries, namely the nouns. In the second step, for the entry describing the plant, the algorithm identified, for all sense definitions 1-2% matches, since it referred to common words such as “care”, “de”, “cu”. For the second word, describing the musical instrument, we have three distinct senses, of which the first sense is further partitioned into 14 sub-senses. After analysing the results, we found that the definition in RoWN pertaining to the musical instrument is 100% matched to one of the first-sense definitions in eDTLR, and the one referring to the archaic instrument only has a 55.55% match with the same definition. This is explained by the fact that: some RoWN sense definitions originated in Dex Online; which in turn is based on definitions from DEX, which has as a main resource the DLR thesaurus.

4. Conclusions

In this paper we have described a method for the alignment of three lexical resources for Romanian, eDTLR, RoWN and CoRoLa, at the word level, which allows users to access them in a common interface. We have also investigated a manner in which existing lexical resources can be enriched and linked at the sense level, as an aid for future attempts at word sense disambiguation for the Romanian language. We intend to further investigate ways in which lexical resources such as eDTLR and RoWN can be further linked at the sense level with greater accuracy, and thus attempt to propose new synsets or to extend existing ones.

Acknowledgements

This study was partially supported by a grant of the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818 / 73PCCDI (ReTeRom).

References

- Bibiri, A.D., Bolea, C.D., Scutelniciu, L.A., Moruz, 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 in Informatics*. ACM New York. ISBN 978-1-4503-3335-1.
- Chiarcos C., McCrae J., Cimiano P., Fellbaum C. (2013). Towards Open Data for Linguistics: Linguistic Linked Data. In: Oltramari A., Vossen P., Qin L., Hovy E. (eds.) *New Trends of Research in Ontologies and Lexical Resources. Theory and Applications of Natural Language Processing*. Springer, Berlin, Heidelberg.
- Cristea, D., Haja, G., Moruz, A., Răschip, M., Pătraşcu, M. I. (2011). Partial statistics at the end of the eDTLR project – the Thesaurus Dictionary of the Romanian Language in electronic form (Statistici parţiale la încheierea proiectului eDTLR – Dicţionarul Tezaur al Limbii Române în format electronic). In R. Zafiu, C. Uşurelu, H. Bogdan Oprea (eds.) *Romanian language. Hypostasis of linguistic variation. Acts of the 10th Colloquium of the Chair of Romanian Language* (Limba română. Ipoteze ale variaţiei lingvistice. Actele celui de-al 10-lea Colocviu al Catedrei de limba română), Bucharest, 3-4 December, vol. I: Grammar and phonology, lexicon, semantics, terminology, history of Romanian language, dialectology and philology (Gramatică şi fonologie, lexic, semantică, terminologie, istoria limbii române, dialectologie şi filologie), Bucharest University Editing House, p. 213-224, ISBN 978-606-16-0046-5.
- Fellbaum, C. (ed.) (1998). *WordNet: An Electronic Lexical Database*, Cambridge, MA, Mit Press, 1998
- McCrae J., Spohr D., Cimiano P. (2011). Linking Lexical Resources and Ontologies on the Semantic Web with Lemon. In: Antoniou G. et al. (eds.) *The Semantic Web: Research and Applications. ESWC 2011. Lecture Notes in Computer Science*, vol 6643. Springer, Berlin, Heidelberg.
- Navigli, T. and Ponzetto, S. (2012). BabelNet: The Automatic Construction, Evaluation and Application of a Wide-Coverage Multilingual Semantic Network. *Artificial Intelligence*, 193, Elsevier, 2012, pp. 217-250.
- Tufiş, D., Cristea, D. (2002). Methodological issues in building the Romanian Wordnet and consistency checks in Balkanet, in *Proceedings of the Workshop on Wordnet Structures and Standardization, and how these affect Wordnet Applications and Evaluation, in conjunction with the Third International Conference on Language Resources and Evaluation (LREC)*, 28-31 May, Las Palmas, Spain, p. 35-41.
- Tufiş, D., Barbu Mititelu, V., Ştefănescu, D., Ion, R. (2013). The Romanian Wordnet in a Nutshell. In *Language Resources and Evaluation*, vol. 47, p. 1305-1314.
- Tufiş, D., Ion, R. (2017). Part of Speech Tagging. In R. Mitkov (ed.) *The Oxford Handbook of Computational Linguistics*, Oxford University Press, 2nd edition.

CHAPTER 4

APPLICATIONS

README - IMPROVING WRITING SKILLS IN ROMANIAN LANGUAGE

MARIA-DORINELA SIRBU¹, MIHAI DASCĂLU^{1,2,4}, DANIELA GÎFU^{4,5,6}, TEODOR-MIHAI COTET¹, ADRIAN TOSCA⁴, STEFAN TRAUSAN-MATU^{1,2,3,4}

¹ *University Politehnica of Bucharest, Computer Science Department, 313 Splaiul Independentei, 060042, Bucharest, Romania*

² *Academy of Romanian Scientists, 54 Splaiul Independenței, Sector 6, Bucharest, 050094, Romania*

³ *Research Institute for Artificial Intelligence “Mihai Drăganescu”, Romanian Academy, 13 Calea 13 Septembrie, 050711, Bucharest, Romania*

⁴ *Cognos Business Consulting S.R.L., 35 Blv Marasesti. Sector 4, Bucharest, 040251, Romania*

⁵ *Alexandru Ioan Cuza University of Iași, Faculty of Computer Science,*

⁶ *Romanian Academy – Iasi branch, Institute of Computer Science, 2, Codrescu St., 700481, Romania*

*maria.sirbu@cti.pub.ro, {mihai.dascalu, stefan.trausan}@cs.pub.ro,
daniela.gifu@info.uaic.ro, teodor_mihai.cotet@stud.acs.upb.ro,
adrian.tosca@cognosconsulting.ro*

Abstract

Writing is a central learning activity that requires both practice and experience. However, a problem arises because evaluating the quality of student text productions in order to provide timely and constructive feedback for improving their writing skills is a time-consuming task for tutors. The aim of this paper is to introduce a new version of our Automated Writing Evaluation system, called *ReadMe*, extended to support Romanian language. Based on an extensible rule engine system, *ReadMe* provides personalized feedback at four granularity levels: word, sentence, paragraph, and document. A corpus of approximately 2,900 non-overlapping text segments from 43 books written in Romanian language was used in our experiments. The open-source *ReaderBench* framework generated over 300 textual complexity indices tailored for Romanian language, which were afterwards considered as input for our custom rule-based engine. Personalized feedback is provided using a user-friendly web interface that facilitates evaluation based on a combination of text colour highlights, coupled with comprehensive text suggestions for improvement.

Keywords: Natural Language Processing, Textual Complexity, Feedback Generation and Visualization, Automated Writing Evaluation

1. Introduction

Writing is a central learning activity that requires a lot of practice and experience. Student's capabilities and their performance are frequently evaluated using the quality of their writing. The entire process is a time-consuming task, and many tutors do not have enough time to perform thorough iterative analyses for their students. Thus, computer-based writing systems have been developed in order to help tutors to provide constructive feedback to students. These systems can provide quick feedback and help students to improve their writing quality.

In addition, students need to be engaged in their writing activity, and these computer-based writing systems can support them throughout their endeavours.

The aim of this paper is to introduce a new version of our Automated Writing Evaluation system, called *ReadMe* (Botarleanu, Dascalu, Sirbu, Crossley, & Trausan-Matu, 2018; Sirbu, Botarleanu, Dascalu, Crossley, & Trausan-Matu, 2018), extended to support Romanian language, as well as multiple improvements made on the web interface and the server-side processing. *ReadMe* can provide personalized feedback at four granularity levels based on an extensible rule-based engine system: document, paragraph, sentence, and word. We used over 300 textual complexity indices based on a corpus of 2,877 writing samples in Romanian language and our open-source advanced Natural Language Processing framework *ReaderBench* (Dascalu, 2014; Dascalu, Dessus, Bianco, Trausan-Matu, & Nardy, 2014). These indices are used as input for our extensible rule-based engine system. A user-friendly web interface was created to provide personalized feedback. Feedback is generated using a combination of text colour highlights based on red and blue gradients, together with easy to understand text recommendations for improving writing quality.

The next section is focused on existing approaches of Automated Writing Evaluation systems. The third section presents our corpus and methods. Afterwards, the fourth section is centered on presenting preliminary results, followed by conclusions and future work.

2. State of the Art

Roscoe and McNamara (2013) highlighted two major categories of textual evaluation systems that provide feedback to students with regards to their writing: Automated Essay Scoring (AES) and Automated Writing Evaluation (AWE). AES systems (Attali & Burstein, 2004; Grimes & Warschauer, 2010) provide accurate and summative scores to learners, and their performance can be automatically evaluated and correlated with human judgments. AWE systems include a feedback loop, are built on top of AES systems with the aim to provide objective feedback that goes beyond holistic scoring (i.e., strive to be formative, besides providing summative scoring) (Warschauer & Ware, 2006). Overall, AWE systems are more useful for students to practice writing and to receive feedback (with or without direct teacher interaction in the classroom). These systems are aimed to help students be aware of potential issues, as well as introduce methods for improvement. Even users are sceptical about the potential usage of these systems (Grimes & Warschauer, 2010) because both AES and AWE depend in general on summative feedback that is derived from textual complexity indices computed given an input text. In fact, these indices provide indications regarding a text's quality or traits of writing style, and can measure different aspects ranging from readability, surface-level numerical metrics, etc. (Crossley, Allen, Snow, & McNamara, 2015; McNamara, Crossley, Roscoe, Allen, & Dai, 2015).

Several automated writing evaluation systems, described below, employ similar methods that are meaningfully integrated into the updated version of our tool, *ReadMe*. This work is an extension of our previous studies, with new features and increased performance.

Part of existing systems are freely available (and, in most cases, these are developed by researchers), while systems developed by companies usually require a small to medium fee.

For instance, MyAccess!¹ is a web-based writing guide which helps students to improve the quality of their own text, while also considering a version tailored for teachers. The system offers writing instructions and feedback to students, including grammatical advices, spell checker, as well as an automated scoring system. For teachers, the system offers the possibility to monitor students, to create templates, and essay assignments.

Another AWE system is WriteToLearn² which helps students enhance their comprehension by teaching them summarization techniques and their writing quality by providing automated feedback. We also list PEG³ (Project Essay Grade) or LightSide Revision Assistant⁴, both capable to provide automated feedback to students or any individual who wants to improve his/her writing.

3. Method

Our aim with *ReadMe* is to create a fully Automated Writing Evaluation system, capable to provide personalized feedback in both English and Romanian languages in order to help learners improve their writing. The underlying processing pipeline, as well as the selected corpora, is detailed below.

3.1. Corpus Selection

Our corpus consists of approximately 2,900 non-overlapping text segments from 43 popular Romanian books. Most of them are literature books, ranging from novels (e.g., “Craii de Curtea-Veche”, en. “Philanderer of Old-Court” by Mateiu Caragiale) to a few non-fiction books which were added for diversity (e.g., “Istoria literaturii române”, en. “History of Romanian Literature” by George Călinescu). Most authors are unique in the corpus, with few exceptions; thus, a great variety of writing styles is expected. Moreover, the periods when the books were written are diverse, starting from “Scrisori către Vasile Alecsandri”, en. “Letters to Vasile Alecsandri” by Ion Ghica from 1883. The 43 books were split into 2,877 individual non-overlapping fragments, which were later used in the *ReaderBench* framework as individual texts to be analyzed. The fragmentation was performed by selecting continuous collections of 3 to maximum 10 paragraphs, each fragment containing at least 250 words, and no more than 1000 words. During this splitting phase, some text parts from the original books were discarded due to their violation of the previously mentioned rules.

3.2. The ReaderBench framework

ReaderBench (Dascalu, Crossley, McNamara, Dessus, & Trausan-Matu, 2018) is an open-source advanced Natural Language Processing framework grounded in Cohesion Network Analysis (CNA) (Dascalu, McNamara, Trausan-Matu, & Allen, 2018), which supports multiple languages, including Romanian language (Dascalu & Gifu, 2015), and computes various lexical, syntactic, semantic, and discourse-centered textual complexity indices. Indices derived at surface level are based on lexical components of the text: word, sentence and paragraph lengths, word, sentence and paragraph counts, punctuation marks. Syntax measures are computed by analyzing the text based on words and sentences, part of speech

¹ <https://www.myaccess.com/myaccess/do/log>

² <https://www.writetolearn.net>

³ <http://www.pegwriting.com>

⁴ <https://www.revisionassistant.com/>

tagging (POS) and correspondent syntactic dependencies. Multiple semantic models, such as Latent Semantic Analysis (LSA) (Landauer, Foltz, & Laham, 1998), Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003), and word2vec (Mikolov, Chen, Corrado, & Dean, 2013) are used to measure semantic similarity.

3.3. The Processing Pipeline

Various processing stages were applied on the previously mentioned corpus in order to compute the corresponding textual complexity indices. These steps are presented in Fig. 1. First, we selected an appropriate corpus. Second, we extracted all paragraphs from the books, with titles and authors, if there are any. Third, we cleanup the data by replacing some characters (e.g., replace “`” with “’”). After cleanup, we use Spacy (<https://spacy.io>) for NLP Pre-Processing step which includes: tokenization, part of speech tagging, syntactic dependency parsing, and lemmatization. As the fifth step, we eliminated all words which are not in the dictionary, have not a part of speech tag or doesn’t contain any textual information, such as prepositions, conjunctions, etc. The final step, *Processing*, represents the most complex part of the processing pipeline. In this step we performed the following stages: build cohesion graph, identify topics, perform CNA scoring, build disambiguation graph, build lexical chains, determine semantic chain / voices, determine voice distribution and importance, and compute textual complexity indices.

3.4. Textual Complexity Indices adapted for Romanian language

The *ReaderBench* framework computes a high number of textual complexity indices and it is centered on semantic and discourse analysis by evaluating textual cohesion. A multi-dimensional analysis presented in Fig. 2 is performed.

Surface analysis is applied on lexical level and computes the following indices: fluency, complexity, legibility, etc., based on lexical elements, such as: words, commas, punctuation marks, sentence’ length. Indices computed in this analysis can be divided in three categories: surface indices, entropy at word and character level, and words’ complexity. Examples of surface indices are: average length of characters per paragraph, average number of commas per paragraph/sentence, average number of sentences per paragraph, average number of words per paragraph/sentence, standard deviation of counted words per paragraph/sentence, etc. Regarding the entropy, there are two types of indices that can be computed: used words’ entropy and words’ entropy. Words’ complexity contains indices like: number of syllables, standard deviation of counted characters, word’s polysemy, etc.

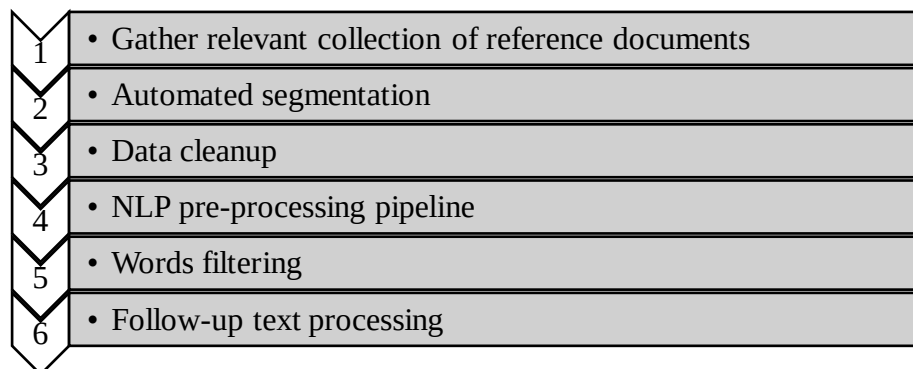


Figure 4. Processing pipeline.

Syntax analysis refers to each sentence's structure, statistics on different parts of speech, and specific syntactic dependency tree. Even if nouns and verbs are the most common parts of speech used in analysis, adjectives, adverbs and prepositions contain also valuable information that they indicate a more complex and deep structure of text. Also, the pronouns illustrate a more complex structure of discourse and the presence of anaphors.

Semantic analysis is based on analyzing the terms. A correct representation of the information is needed to easily understand a text. Besides the text segments correlated between them, a well-structured document must have a clear and logical movement of the highlighted ideas. Previous studies (Dascalu, 2013) shown that the semantic similarity Wu-Palmer (Wu & Palmer, 1994), based on ontologies and combined with semantic models like LSA, LDA and word2vec, can complement each other in order to compute a detailed semantic analysis.

Regarding *discourse structure*, multiple regular expressions and cue phrases can be used to recognize the occurrence of specific discourse's connectors. *Word complexity* considers the number of syllables, word's polysemy count, the depth in hypernym tree, etc. The number of syllables is direct proportional with the difficulty of a word – more syllables means it is harder to pronounce; a multi-sense word is harder to use and to identify its correct sense.

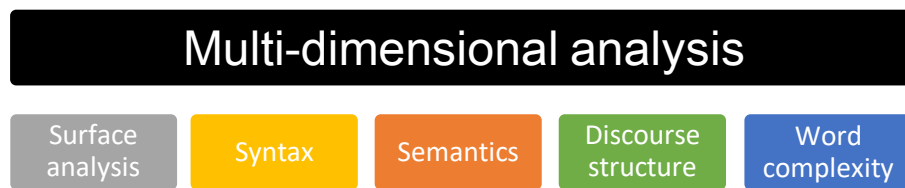


Figure 5. Textual complexity indices available for Romanian language.

3.5. System Overview and Architecture

ReadMe contains two major components: a) the *backend* – responsible for text processing, rule-based engine system configurations, requests processing –, and b) the *frontend* – designed as an interactive and user-friendly web interface which allows users to introduce an essay and receive comprehensive feedback based on text highlighting. Fig. 3 presents the architecture of our *ReadMe* system, as well as the exchanged requests. The *ReadMe Visualization Interface* module represents the interactive views in which users introduce essays and receive feedback. The texts are afterwards sent to the *ReadMe Server* module using a REST web service. The *ReadMe Server* module processes the text using *ReaderBench framework* component which sent back as response many textual complexity indices. After all complexity indices are computed using the *ReaderBench framework*, the *ReadMe Server* processes them, applies the pre-defined rules and sends the feedback back to the *Readme Visualization Interface*.

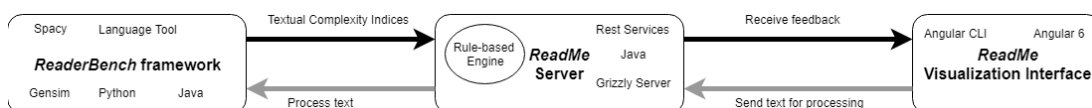


Figure 6. ReadMe – System Architecture.

3.6. *Personalized Feedback Generation using an extensible rule-based engine*

ReadMe contains an extensible rule-based engine system capable to provide comprehensive feedback based on multiple rules manually configured in a JSON file. The engine supports two types of rules for English language: rules based on textual complexity indices, and rules based on components generated using a Principal Component Analysis (PCA) (Jolliffe, 2011). For Romanian language, the current release of the system supports only rules based on individual indices computed by the *ReaderBench* framework. A rule contains the followings: a) a minimum acceptable value, b) a maximum acceptable value, c) one or more feedback messages in the case the document exceeds the minimum threshold for a given index, d) one or more feedback messages for exceeding the maximum threshold, e) as well as the level for which the index is applied: word, sentence, paragraph or entire document. In order to provide a more tailored feedback, besides considering the normal distributions of selected indices, we also introduced some rules based on expert judgment (e.g., “no phrase should have more than 35 words.”; in this case, triggering this rule generates a message to cut the phrase into more coherent sub-phrases).

ReadMe supports four levels of granularity for the performed analysis, namely: document, paragraph, sentence, and word level. For each level, multiple rules are defined. Due to the facile manner in which experts can add, remove or update a rule via the JSON configuration file, more rules can be added every time. In order to avoid monotony, we defined more than two feedback messages for each rule which are selected in a random manner.

3.7. *A user-friendly, interactive and multi-layered visualization interface*

We created an interactive web interface using Angular 6⁵, capable to provide comprehensive and personalized feedback to users. The interface is intuitive and user-friendly – i.e., users must type a text as input and follow a simple sequential wizard. Users can go to the next step or return to the previous one, as they see fit. In the current version, our interactive interface has two steps: a) *checking for grammar* and b) *provide feedback*. In the *checking for grammar* step, the interface highlights the words that are incorrect and provides some recommendations to replace those words, on a mouse-over event. Regarding the second step, the interface generates feedback at four granularity levels, as previously defined.

Users have to select the granularity level for which they want to receive feedback from a dropdown; document level is selected by default. After selecting the desired granularity level, feedback is provided using text highlights and recommendations for improvements. In the right-hand side, the text is displayed as being highlighted with two colour gradients (red and blue) which indicate the severity of problems. On a mouse-over event on a word, sentence, paragraph, or document (depending on the selected granularity level), the rules that were triggered are displayed in the left-hand side, together with corresponding recommendations. The two-colour gradients indicate if the selected text has problem or not. A red gradient means that at least one rule was triggered by the engine, with darker shades suggesting more problems with the text. Blue gradient means that no rules were triggered for that specific text fragment, with darker shades suggesting that the text fragments are similar to the average values across the entire reference collection.

⁵ <https://angular.io/docs>

4. Results

We have introduced a user-friendly web application, capable to provide recommendations regarding spelling, together with personalized feedback, in order to help users improve their writing skills. Using a wizard component, the feedback is provided in steps, helping users throughout the learning process. Step one consists of orthographic analysis. Fig. 4 presents the recommendations obtained for spell-checking a specific input text. In the right-hand side the text introduced by the user is displayed, with incorrectly written words coloured in red. On a mouse-over event over a coloured word, in the left side the recommendations offered to replace that word will appear. The word is underlined on mouse-over event. In Fig. 5, the mouse cursor was put over the coloured word “limbagiul” which is incorrectly written, and the recommendations for this word are: “lampagiul, limbajul, limbarul, limbatul, limonagiul, lumbagoul”. From all of these recommendations, the adequate word is “limbajul” (the second recommendation).

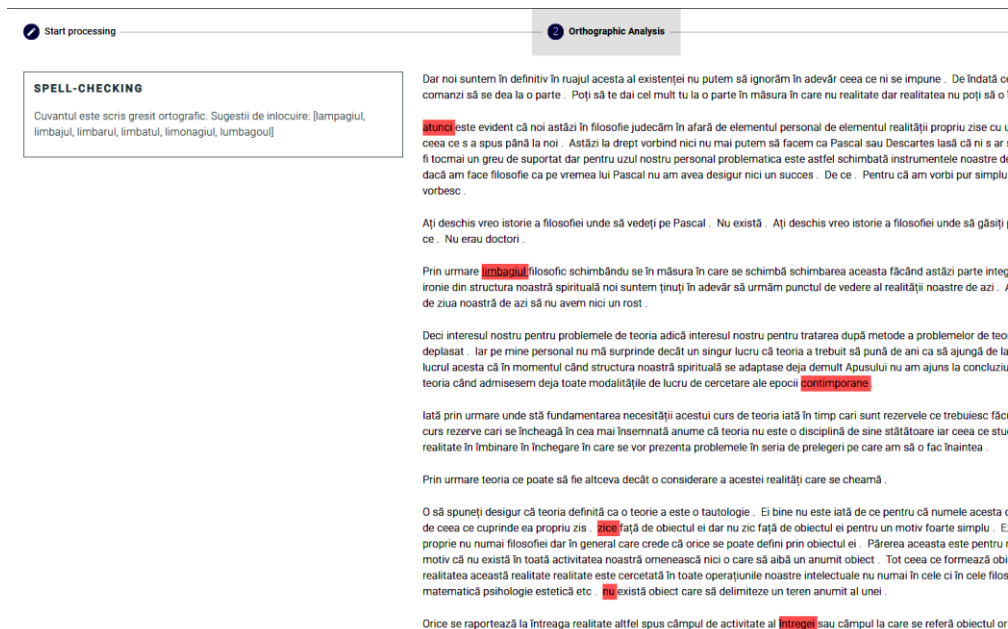


Figure 7. Orthographic Analysis – Recommendations for replacement.

Going further, the second step consists of generating personalized feedback. *ReadMe* provides feedback at four granularity levels, as previously defined, and consistent with the drop-down menu: document, paragraph, sentence, and word. Two colour gradients, blue and red, are used to indicate the severity of issue. Fig. 5 presents a feedback generated at document level, the text is coloured depending on the severity level in the right side, while personalized feedback is provided in the left-hand side. In the specific example presented in Fig. 5, the document presents issues, defined in the left-hand side: the number of paragraphs is too high reported to document's size, the average length of words is too high, while the usage of simpler words is recommended.

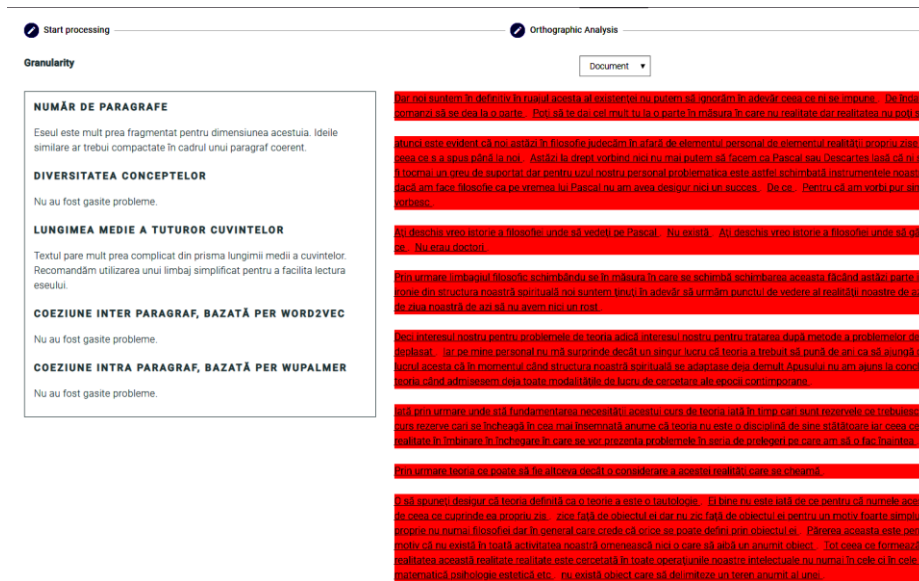


Figure 8. Personalized Feedback – document level.

Going deeper into the analysis by selecting granularity at paragraph level, Fig. 6 shows the feedback for each paragraph on a mouse-over event. Each paragraph is highlighted depending on the identified problems (e.g., paragraphs 2 and 4 to 8 exhibit a more intense red due to the increased number of issues). For example, the sixth paragraph (the one selected in the printscreen) has problems with the number of sentences, inter-paragraph cohesion based on Wu-Palmer similarity (Wu & Palmer, 1994), and intra-paragraph cohesion based on word2vec.

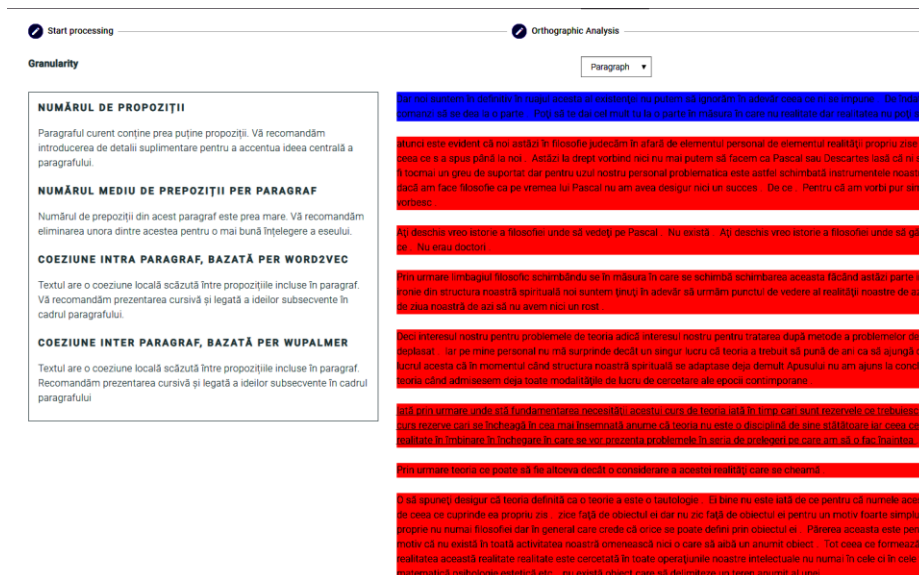


Figure 9. Personalized Feedback – paragraph level.

Sentence level granularity provides feedback at a lower level. Depending on the severity, a darker red shade is used for highlighting the sentences with a higher number of issues. In Fig. 7, the sentence for the second paragraph has most issues: a large number of words, nouns, adjectives, and adverbs, all together. Sentences with blue require the least attention, because no issues were identified.

README - IMPROVING WRITING SKILLS IN ROMANIAN LANGUAGE

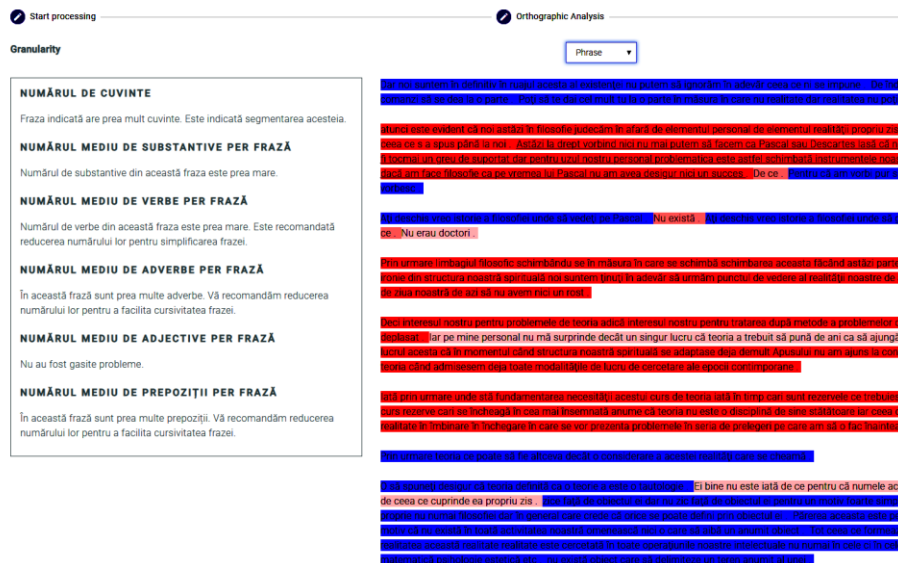


Figure 10. Personalized Feedback – sentence level.

By setting the granularity at word level, feedback regarding individual words can be observed (see Fig. 8). Words coloured in blue suggest that these concepts do not present irregular values among the indices used by the word level rules.



Figure 11. Personalized Feedback – word level.

5. Conclusions and Future Work

Writing quality is a valuable component in evaluating students’ capabilities and performance. In this paper we have introduced the new version of our AWE system –*ReadMe*, with a new visualization interface and improvements to underlying methods, capable to provide tailored feedback for essays written in Romanian language at four granularity levels. The open-source *ReaderBench* framework was extended to support 300 textual complexity indices for Romanian language, which were afterwards considered as input for the *ReadMe* rule-based engine. Personalized feedback is provided using a user-friendly web interface which combines text colour highlights with comprehensive text suggestions for improvement. This

study is targeted to support students, professors, writers, as well as specialists in NLP and linguists, by providing a wide range of complexity indices and personalized feedback to improve writing skills required for proficient essays in Romanian language.

At present we did not perform any classroom experiments; thus, no feedback from students or tutors is currently available. We will continue to develop the *ReadMe* system using different training corpora that reflect students' writing, and we want to test our AWE system with students of different educational levels. In addition, we also strive to enhance the provided visualizations with interactive concept maps of used keywords and student evolution graphs in terms of writing skills.

Acknowledgments

This research was partially funded by the README project "Interactive and Innovative application for evaluating the readability of texts in Romanian Language and for improving users' writing styles", contract no. 114/15.09.2017, MySMIS 2014 code 119286 and by the grant of the Romanian Ministry of Research and Innovation, CCCDI – UEFISCDI, project number PN-III-P1-1.2-PCCDI-2017-0818/73PCCDI (ReTeRom), within PNCDI III.

References

- Attali, Y., & Burstein, J. (2004). Automated essay scoring with e-rater V.2.0. In *Annual Meeting of the International Association for Educational Assessment* (pp. 23). Philadelphia, PA: Association for Educational Assessment.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3(4-5), 993–1022.
- Botarleanu, R.-M., Dascalu, M., Sirbu, M.-D., Crossley, S. A., & Trausan-Matu, S. (2018). ReadME – Generating Personalized Feedback for Essay Writing using the ReaderBench Framework. In H. Knoche, E. Popescu & A. Cartelli (Eds.), *3rd Int. Conf. on Smart Learning Ecosystems and Regional Development (SLERD 2018)* (pp. 133–145). Aalborg, Denmark: Springer.
- Crossley, S. A., Allen, L. K., Snow, E. L., & McNamara, D. S. (2015). Pssst...textual Features... there is more to automatic essay scoring than just you! In *5th Int. Learning Analytics & Knowledge Conf. (LAK'15)* (pp. 203–207). Poughkeepsie, NY: ACM.
- Dascalu, M. (2013). *Analyzing discourse and text complexity for learning and collaborating, Studies in Computational Intelligence*. (Doctoral dissertation), Cotutelle between University Politehnica of Bucharest and University Grenoble Alpes, Bucharest, Romania. Retrieved from <http://tel.archives-ouvertes.fr/tel-00978420>
- Dascalu, M. (2014). *Analyzing discourse and text complexity for learning and collaborating, Studies in Computational Intelligence* (Vol. 534). Switzerland: Springer.
- Dascalu, M., Crossley, S., McNamara, D. S., Dessus, P., & Trausan-Matu, S. (2018). Please ReaderBench this Text: A Multi-Dimensional Textual Complexity Assessment Framework. In S. Craig (Ed.), *Tutoring and Intelligent Tutoring Systems* (pp. 251–271). Hauppauge, NY, USA: Nova Science Publishers, Inc.

- Dascalu, M., Dessus, P., Bianco, M., Trausan-Matu, S., & Nardy, A. (2014). Mining texts, learner productions and strategies with ReaderBench. In A. Peña-Ayala (Ed.), *Educational Data Mining: Applications and Trends* (pp. 345–377). Cham, Switzerland: Springer.
- Dascalu, M., & Gifu, D. (2015). Evaluating the Complexity of Online Romanian Press. In D. Gifu, D. Trandabat, D. Cristea & D. Tufis (Eds.), *11th Int. Conf. "Linguistic Resources and Tools for Processing the Romanian Language"* (pp. 149–162). Iasi, Romania.
- Dascalu, M., McNamara, D. S., Trausan-Matu, S., & Allen, L.K. (2018). Cohesion Network Analysis of CSCL Participation. *Behavior Research Methods*, 50(2), 604–619. doi: 10.3758/s13428-017-0888-4
- Grimes, Douglas, & Warschauer, Mark. (2010). Utility in a fallible tool: A multi-site case study of automated writing evaluation. *The Journal of Technology, Learning and Assessment*, 8(6).
- Jolliffe, Ian. (2011). Principal component analysis *International encyclopedia of statistical science* (pp. 1094-1096): Springer.
- Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to Latent Semantic Analysis. *Discourse Processes*, 25(2/3), 259–284.
- McNamara, D.S., Crossley, S.A., Roscoe, R., Allen, L.K., & Dai, J. (2015). A hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23, 35–59.
- Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representation in Vector Space. In *Workshop at ICLR*. Scottsdale, AZ.
- Roscoe, Rod D, & McNamara, Danielle S. (2013). Writing Pal: Feasibility of an intelligent writing strategy tutor in the high school classroom. *Journal of Educational Psychology*, 105(4), 1010.
- Sirbu, M. D., Botarleanu, R., Dascalu, M., Crossley, S., & Trausan-Matu, S. (2018). ReadME – Enhancing Automated Writing Evaluation. In *18th Int. Conf. on Artificial Intelligence: Methodology, Systems, and Applications (AIMSA 2018)* (pp. 281–285). Varna, Bulgaria: Springer.
- Warschauer, M., & Ware, P. (2006). Automated writing evaluation: defining the classroom research agenda. *Language Teaching Research*, 10, 1–24.
- Wu, Zhibiao, & Palmer, Martha. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics* (pp. 133-138): Association for Computational Linguistics.

NEW GIS BASED APPROACHES FOR THE LINGUISTIC ATLASES

SILVIU-IOAN BEJINARIU¹, VASILE APOPEI¹, MANUELA NEVACI², NICOLAE SARAMANDU²

¹ *Institute of Computer Science, Romanian Academy Iasi Branch, Iași, Romania, {silviu.bejinariu, vasile.apopei}@iit.academiaromana-is.ro*

² *“Iorgu Iordan – Al. Rosetti” Institute of Linguistics, Romanian Academy, Bucharest, Romania, {elanevac, nsaramandu}@yahoo.com*

Abstract

The geographic position is an essential property of the linguistic information which makes possible the spatial analysis of the spoken languages in dialectology. Nowadays, the information systems are increasingly used in all the fields. The usage of Geographic Information Systems (GIS) in dialectology allows the spatial analysis of the evolution of languages and dialects as well as the relationships between them. In this paper, two different GIS based approaches developed for the Linguistic Atlas of the Aromanian Dialect (ALAR) and for the European Linguistic Atlas (ALE) publishing are briefly presented. Apart from an increased quality of the printed maps at any level of details, the usage of relational databases allows linguistic data storage in a standard structure making it available for the automated analysis of language variation. The main difference between the two projects is that in ALAR the GIS support is used for the preparation of the map layout which is then printed using an own application, while in ALE, the map is printed using a specialized GIS software. The interactive consultation of maps and linguistic information is a future step of this research.

Key words — linguistic atlas, geographic information system, linguistic database.

1. Introduction

The Romanian dialectologists have a rich tradition in linguistic cartography. Starting from the first editions in which both data collection and linguistic maps drawing were made manually, the research was continued by using the information technology tools first for map drawing and then for data storage. The geographic position of the linguistic information was not used. More than that, the lack of the geographic information made difficult the development of dialectological studies from the point of view of the spatial analysis. Another important problem for the Romanian linguistic atlases is the great number of possible symbols which can be used for the phonetic transcriptions.

The system developed by researchers from the Institute of Computer Science and “A. Philippide” Institute of Romanian Philology since 2000 (Bejinariu *et al.*, 2006), (Bejinariu *et al.*, 2007), (Bejinariu *et al.*, 2009), (Botoșineanu *et al.*, 2010) allowed the computer-aided publication of two volumes of the *Noul Atlas Lingvistic Român pe Regiuni - Moldova și Bucovina* (*The New Romanian Linguistic Atlas by Regions - Moldavia and Bukovina*, NALR-MB) in 2007 and 2014. In this first version of the software package, entitled ALR (*Atlas Lingvistic Român pe Regiuni*), the most important problems solved are: the linguistic information is structured in a linguistic database, the symbols are drawn using a graphic

image synthesizer, a vectorial description is used to draw maps, synthetic maps (Fig. 1), indexes and uncartographed materials are automatically generated, but the information is not yet related to the real geographic position.

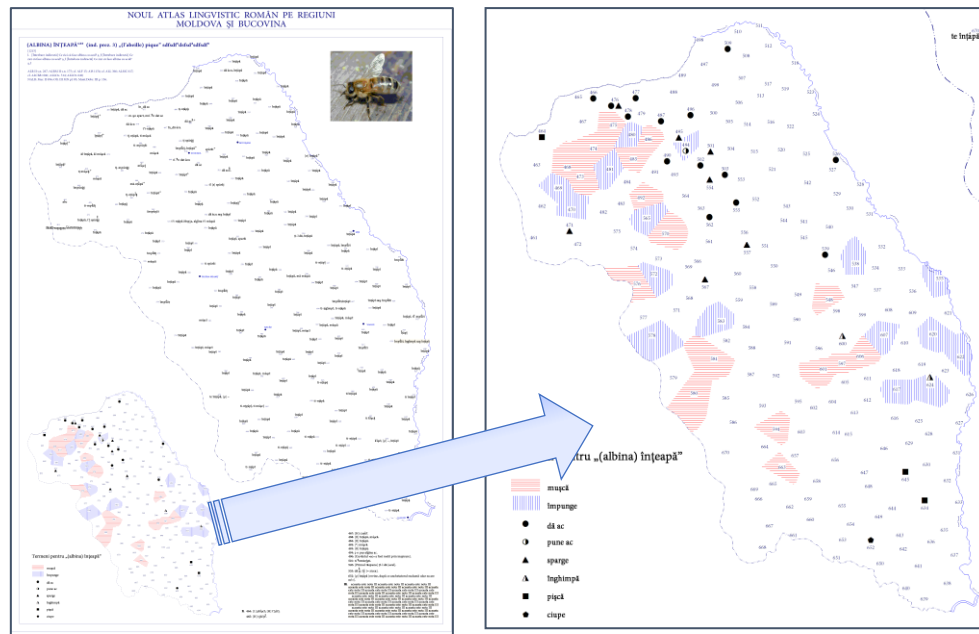


Figure 1: Example of linguistic map from ALR-MB and the corresponding synthetic map in detail

After few years in which the accumulations consisted mainly of the digitalization of linguistic information for the new volumes of *NALR-MB*, new opportunities emerged for the development of the *ALR* application. On this occasion, the whole *ALR* application was redesigned, considering the possibility of correlating the linguistic information with the geographic location. The first challenge was the implementation of the second stage of the *Audio-visual Linguistic Atlas of Bukovina (ALAB)* project developed by the colleagues from the “A. Philippide” Institute of Romanian Philology. The map layout was prepared using QGIS (free and open-source professional GIS software), (QGIS, 2018). The geographic features are described in the geographic coordinate system, and then the maps are drawn using the new version of the *ALR* application, namely *AlrMaps*. An important newly introduced feature is the possibility of displaying several phonetic transcriptions for each investigated point (Fig. 2). The project is detailed in (Bejinariu *et al.*, 2016), (Olariu *et al.*, 2016), (Bejinariu and Olariu, 2017) and (Olariu, 2017). The second challenge was the development of a software application for the *Linguistic Atlas of the Aromanian Dialect (ALAR)* publishing. This task was done by the Institute of Computer Science and the “Iorgu Iordan – Al. Rosetti” Institute of Linguistics. In *ALAR*, the GIS support is used only for the map layout generation, as in the case of *ALAB* project. The cooperation between the two institutes is continuing with the implementation of a system for the *Atlas Linguarum Europae (ALE)* publishing.

The rest of the paper is organized as follows. The second section briefly describes some aspects related to the GIS usage in geolinguistics. The *ALAR* project is described in the third section of the paper. In the fourth section, the *ALE* project, which is being developed using the full support of the QGIS software, is described. The last section concludes the paper.

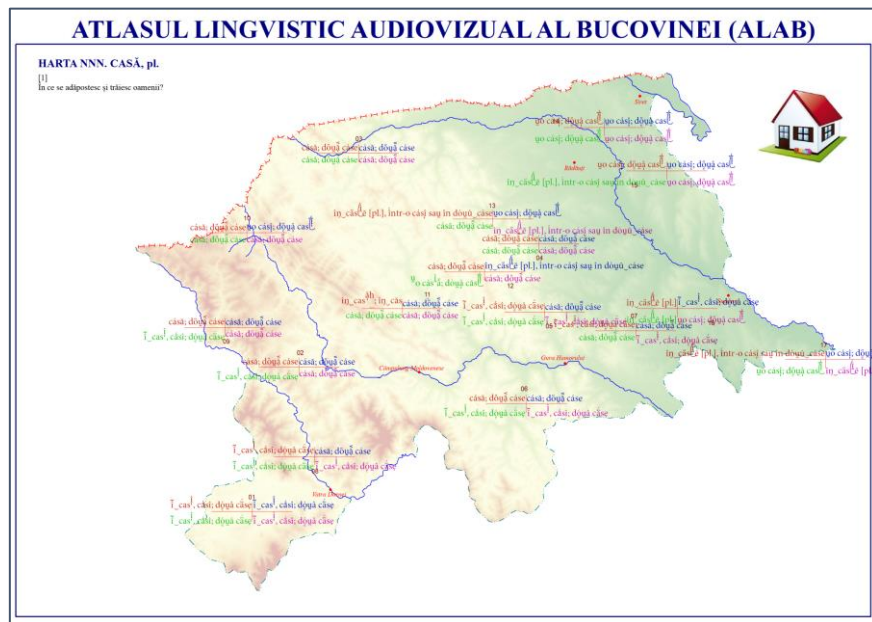


Figure 2: Example of linguistic map from ALAB

2. Geographic Information Systems and linguistic cartography

GIS is defined as an information system which allows the acquisition, storage, validation, integration, manipulation, analysis and visualization of geospatial data (Clarke, 1986). In fact, a GIS integrates all the resources used for geospatial data processing, including equipment, applications, people, rules, information and methods. The geospatial data is usually obtained from different sources and can use different modes of representation and/or encoding. All data are interconnected through the associated geographical position, thus allowing data redundancy to be reduced.

The geospatial database contains the following main components:

- Digital map – describes the features existing on the earth's surface; usually it is organized in layers, one for each type of spatial objects; the description can be either geometrical (shape, position and size) or a raster image (aerial or satellite).
- Internal database – for each layer there is a table with the common attributes of the features.
- External database – other application-specific information related to the objects in the digital map. The information in the external database is not necessarily linked to a specific object or geographic position.

As it was already mentioned in the first section, there are two manners for GIS usage in geolinguistics. In the first case, it is used only for map layout preparation and this manner is applied mostly in case of printed linguistic atlases, when the map contains complex elements which are difficult to manage from a GIS. In *NALR-MB* and *ALAR*, the map contains phonetic transcriptions whose images are synthesized in real time depending on the current size of the map and respectively on the required size of the font and this was the reason for which this solution was adopted. The maps are only generated, edited and printed using an application

implemented for this purpose, even if it contains elements which are specific to a GIS. In the second case, the publishing process is fully managed using the GIS. In some situations an application for the internal and external databases maintenance is required, most often to edit the linguistic information. In the *ALE* project, this solution was adopted for two reasons: the linguistic map contains only symbols, and the map is quite similar to a classical geographic map.

Apart from the recent approaches presented in the first section of this paper, the problem of using GIS in linguistics has begun to be studied about more than 30 years ago. The quantitative spatial analysis has been used in (Lee and Kretzschmar, 1993), to analyse the words usage frequency in the Linguistic Atlas of the Middle and South Atlantic States. A significant Romanian project developed in the last decade by the “Sextil Pușcariu” Institute of Linguistics and Literary History and the York University, is the Romanian Online Dialect Atlas (RODA) application which was used to publish the third volume of *The New Romanian Linguistic Atlas by Regions – Crișana (NALR-C)* and *The Romanian Linguistic Atlas. Spoken Languages between Morava, Danube and Timoc (ALR – MDT)* (Urișescu, 2014). The most recent approaches are focused on the design of interactive Web versions for the linguistic atlases, as that presented in (Olariu *et al.*, 2016). However, many web linguistic atlases show images that contain the printed pages, having a limited interactivity, as (ALF, 2017). On the other hand, an increasing attention is paid to the study of languages in extinction (Junker and Stewart, 2011).

3. *Linguistic Atlas of the Aromanian Dialect (ALAR)*

The Linguistic Atlas of the Aromanian Dialect (Saramandu and Nevaci, 2014) is the most extensive geo-linguistic research of the Aromanian dialect and is the result of a long research conducted in Greece, Albania, R. Macedonia and Bulgaria. The research area is comprised between the Balkan Mountains in the north, Thessaly, Acarnania and Etolia in the south, Adriatic and Ionian seas in the west and the Aegean Sea in the east. *ALAR* highlights the Roman origin of the Aromanians (Nevaci, 2013), (Nevaci, 2016), (Saramandu and Nevaci, 2006).

The management of the linguistic information as well as the maps generation, editing and printing are made using the *AlrMaps* application developed for this purpose. In the next paragraphs some implementation aspects are briefly described (Apopei *et al.*, 2016).

3.1. *ALAR databases*

In *ALAR*, the linguistic database is stored in a two-component dictionary file: the list of survey points and the title word list. In order to enter information into the dictionaries, the *AlrMaps* application provides a user-friendly interface that shows the elements in the dictionary. The user can consult and / or edit the linguistic information linked to the selected item in the dictionary. Each survey point is associated with the following key information: the point code, name, country, language used in that location. Because *AlrMaps* is used also for the Romanian linguistic atlases, other specific information is included: the survey point code in the old linguistic atlases, the county, the region, as well as information on age, gender, ethnicity, education level and occupation of the interviewee. In *ALAR* there are 61 survey points distributed as follows: 32 in Greece, 12 in Albania, 9 in Macedonia, 8 in Bulgaria. An *ALAR* specific information is the speech (“grai”) used in the corresponding location: *pindean*,

The geospatial database which is managed using a GIS application contains mainly the layout of the maps including here not only the description of the earth's features (borders, rivers, major cities) but also the location of the phonetic transcriptions on the printed map. The digital map is organized in layers which are exemplified in Fig. 3 using the *shape* file format. The digital map is used for automatic map generation and the user has then possibility to customize each map in a specific manner by moving, resizing or changing drawing mode for each or for all objects in a layer. The internal GIS database contains mainly the links to the records in the linguistic database, e.g. for the elements in the phonetic transcriptions layer, the internal database contain the survey point code for which the phonetic transcription has to be drawn. Some other common drawing properties as lines thickness, line colour, font size, text colour, and text alignment mode are described in the internal database.

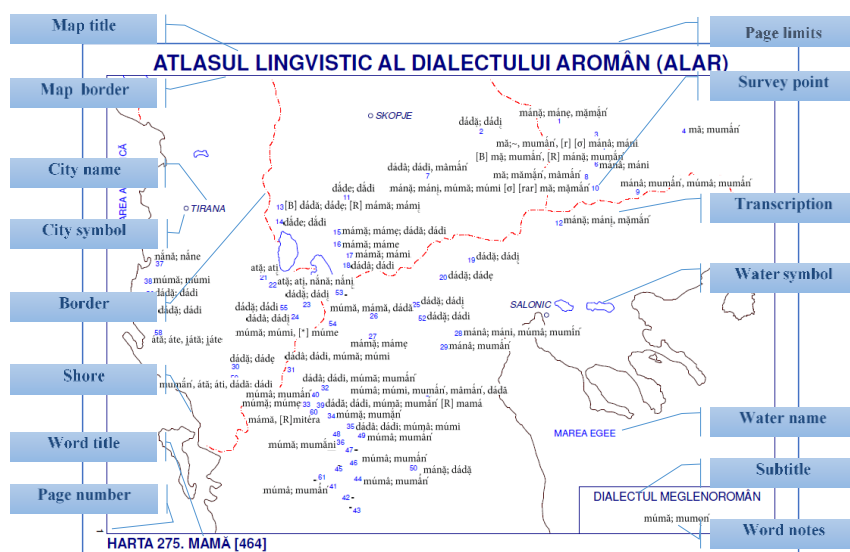


Figure 3: The layers included in the *ALAR* digital map. In background is the linguistic map 464, for title word “Mamă” from *ALAR*.

The *AlrMaps* application allows the generation of phonetic maps (background image in Fig. 3) for each title word in the dictionary as well as the map of the localities and the map of spoken languages distribution. Once generated, the maps can be customized by changing the size, position, drawing properties for a single or simultaneously for all the objects in a specific layer, by using a user friendly interface similar to those used in most graphic editing applications (Fig. 4).

The main advantage of using a dedicated application for linguistic maps generation and printing is that the images of the phonetic transcriptions are generated in real-time considering the specific conditions related to the map size, resolution, magnifying factor, etc. Thus is avoided the situation in which previously generated images of the transcriptions have to be resized, leading to lower quality of the graphics.

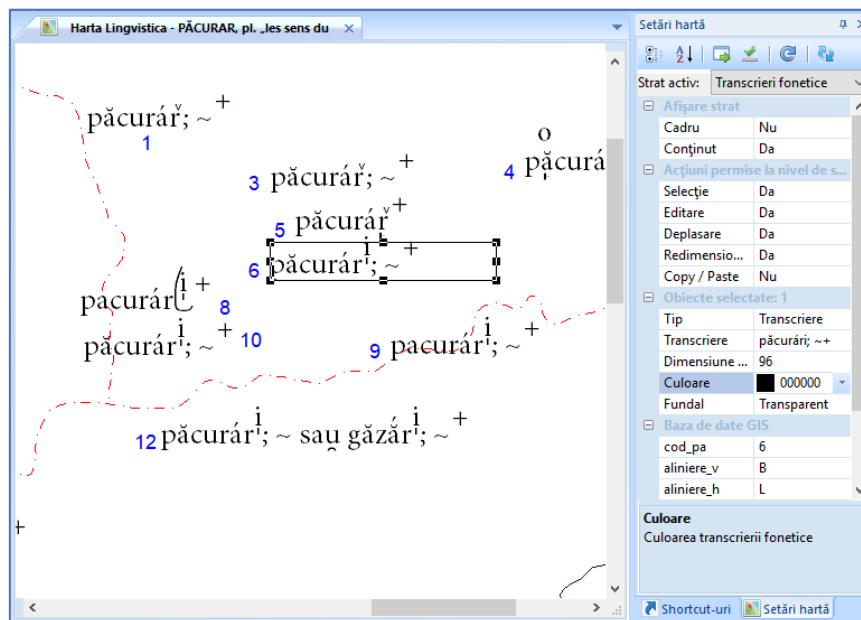


Figure 4: Maps editing interface in the *AlrMaps* application

The *AlrMaps* application allows the user to create not only linguistic maps but also to automatically generate indexes and pages with the cartographed or uncartographed linguistic material in a tabular style. The main gain of this project is that the linguistic information is stored in a database which makes possible its automatic processing in future.

4. *Atlas Liguarum Europae (ALE)*

The use of languages is not limited by political and administrative borders, thus a more general approach is required especially in cases when the studied area exceeds the borders of a single area or region. Such a project is that of the European Linguistic Atlas (the official name is *Atlas Languarum Europae - ALE*) started in 1965, when its realization was proposed at The Geolinguistics International Congress in Marburg (ALE, 2018). *ALE* is periodically issued by an association of 47 national committees. It offers a rigorous typology of languages and dialects spoken in Europe and it contains semasiological, onomasiological and motivational maps. Motivational maps are an innovation in the field of linguistic geography and a modern form of interpreting lexical data. *ALE* is a lexical atlas, but certain phonetic and morphological phenomena are also presented (ALE, 2018).

The complexity of *ALE* results from the great number of languages. In Europe are used more than 90 languages and dialects which belong to 22 language groups and they are in turn grouped in six families: Altaic, Basque, Indo-European, Caucasian, Semitic and Uralic. Regarding the number of survey points, it must be mentioned that material was collected in 2631 localities from Iceland to the Ural Mountains (ALE, 2018).

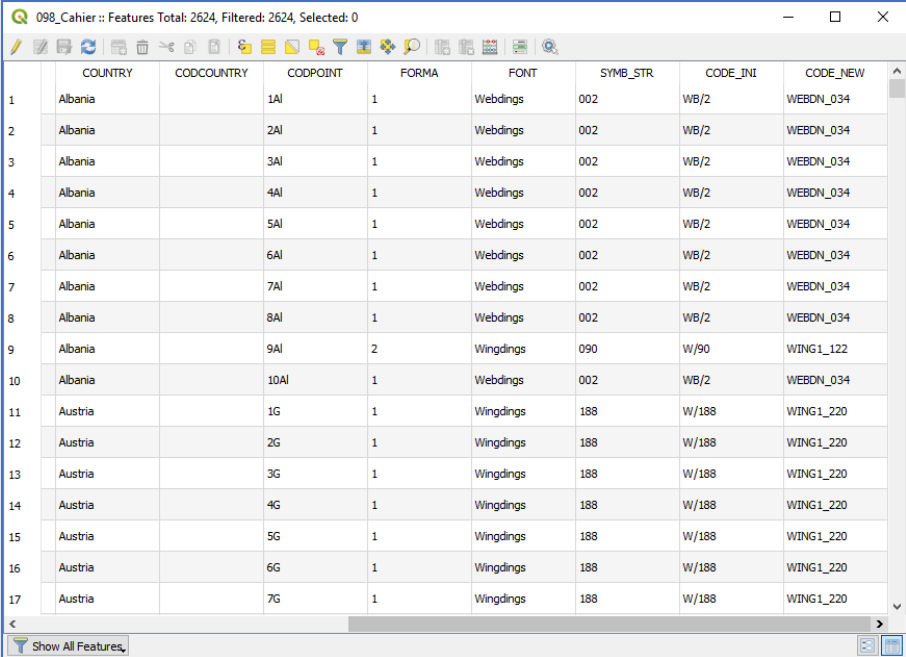
Over time, the previous editions of *ALE* have been published by different publishing houses using the material provided by the *ALE* national committees. The linguistic material was stored in electronic format in text documents which does not allow automatic processing without using specialized parsers. Since 2017, the Institute of Computer Science and the “Iorgu Iordan – Al. Rosetti” Institute of Linguistics cooperate to develop a new GIS based system for *ALE* publishing.

Because of the large number of survey points (more than 2600), it is not possible to show on the map the words or their phonetic transcription using a size of the fonts which makes them readable. This is the reason for which in *ALE* the language variation is shown using symbols and legends, which makes it easier to use a GIS to completely manage and print the linguistic maps. Thus, the QGIS, a free and open-source cross-platform GIS (QGIS, 2018) is used in this project. However, a separate application (*eALE*) is developed for the linguistic database management. Even if the *eALE* application is under development, for testing purposes it can be used to create the links between the external linguistic database and the digital map by transferring the required information into the internal database.

4.1. *eALE* databases

The *ALE* digital map is prepared using QGIS. Comparing to *ALAR* maps, it contains fewer layers, because all the maps components related map title, page borders and page number are parts of the layout, which is separated from the map in QGIS. The following layers are used (Bejinariu *et al.*, 2018), (Saramandu *et al.*, 2017):

- Survey points – which are characterized by their number in *ALE*,
- Frontiers – borders of the countries visible on the map (EuroGeographics, 2018),
- Capitals – capitals of the countries visible on the *ALE* map,



	COUNTRY	CODCOUNTRY	CODPOINT	FORMA	FONT	SYMB_STR	CODE_INI	CODE_NEW
1	Albania		1AI	1	Webdings	002	WB/2	WEBDN_034
2	Albania		2AI	1	Webdings	002	WB/2	WEBDN_034
3	Albania		3AI	1	Webdings	002	WB/2	WEBDN_034
4	Albania		4AI	1	Webdings	002	WB/2	WEBDN_034
5	Albania		5AI	1	Webdings	002	WB/2	WEBDN_034
6	Albania		6AI	1	Webdings	002	WB/2	WEBDN_034
7	Albania		7AI	1	Webdings	002	WB/2	WEBDN_034
8	Albania		8AI	1	Webdings	002	WB/2	WEBDN_034
9	Albania		9AI	2	Wingdings	090	W/90	WING1_122
10	Albania		10AI	1	Webdings	002	WB/2	WEBDN_034
11	Austria		1G	1	Wingdings	188	W/188	WING1_220
12	Austria		2G	1	Wingdings	188	W/188	WING1_220
13	Austria		3G	1	Wingdings	188	W/188	WING1_220
14	Austria		4G	1	Wingdings	188	W/188	WING1_220
15	Austria		5G	1	Wingdings	188	W/188	WING1_220
16	Austria		6G	1	Wingdings	188	W/188	WING1_220
17	Austria		7G	1	Wingdings	188	W/188	WING1_220

Figure 5: The ‘Survey points’ table shown in the QGIS interface

- DEM – digital elevation model, raster image used to show the relief. In fact, there are 3 layers: DEM_Europe, DEM_Asia and DEM_Africa, because apart from Europe's map, small areas of Africa and Asia are visible in the full *ALE* map (USGS, 2018).
- Limits – the map limit; used to draw a frame around the map.

In contrast to *ALAR* maps, where the map objects and the corresponding labels are on different layers, in *ALE* the labels are attributes of the objects and are drawn by QGIS in the most convenient position to avoid overlapping. Therefore, the symbol used to show the word used in each survey point is encoded as one of its attributes in the internal database. The table Survey points in the internal database contains the fields shown in Fig. 5 using the QGIS interface. In the external linguistic database the symbols are specified by a string (field CODE_INI) in which the ASCII code of the symbol and the font name are encoded. The current version of the *eALE* application decodes this string and translates its content into the fields FONT, SYMB_STR and CODE_NEW in the internal database. QGIS is able to draw only symbols stored in the SVG (Scalable Vector Graphics) file format, thus in the CODE_NEW field is specified the name of the corresponding SVG file. All the other drawing properties, like colour, size, buffer, shadow, label position, label colour, etc. are specified as settings of the layer and stored in the QGIS project file.

The external linguistic database has a structure which is similar to that used in *ALAR*. However, some *ALE* specific fields are included. For example, each record in the 'Survey point' table contains information about the country and the language used. The interface used for the external database management is quite similar to that of *AlrMaps* application.

The symbols library can be considered as a component of the external database. In fact it is a collection of SVG files manually created using the *FontForge*, a free font editor application.

4.2. *eALE* maps

In QGIS, for each map, more printing layouts have been defined. Each layout is specific for a paper size because all map's components must be drawn using a scale which is relative to the paper size. The layout includes the legend prepared in HTML format and it specifies the map number, word name in six languages, map authors and other specific notes (Fig. 6).

Volume I.9, carte 98	I.98
QI: 495	
Cahier	
Тетрадь	
Notebook	
Heft	
Cuaderno	
Cuaderno	
Carte de motivations	
Autheurs: Nicolae Saramandu, Manuela Nevaci	
Voir Commentaires, I. fasc. 9, LXXVI	

Figure 6: Legend of map 'Cahier' from *ALE* (Nevaci and Radu, 2012)

Other drawing properties like map extent, map title, map frame, geographic grid and geographic coordinates are specified in the layout. The maps can be generated with different

levels of detail by modifying the map extent, e.g. in Fig. 7 is depicted the full *ALE* map, and in Fig. 8 is drawn only the area of Romania.

Another component of the QGIS package is the QGIS map server which allows the user to modify and to render the maps directly in a Web browser. This feature will be used in the future for the *ALE* project.

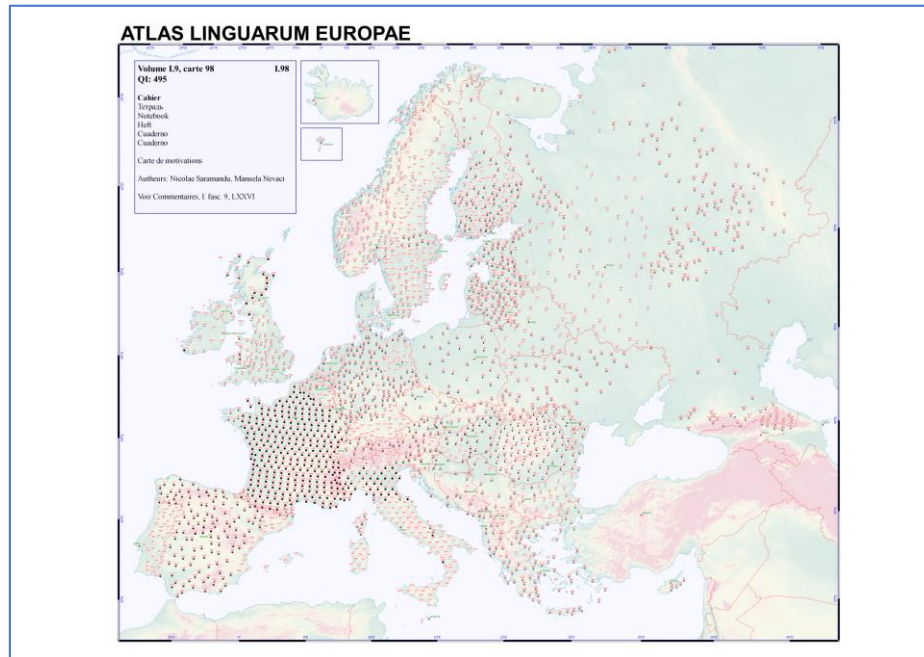
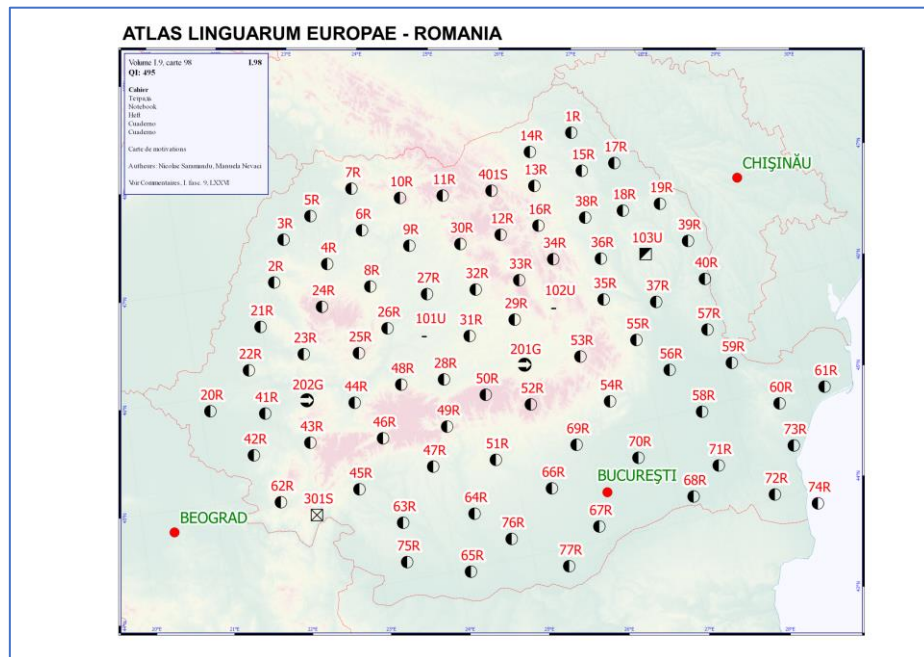


Figure 7: Map ‘Cahier’ from *ALE*, usually printed using the A2 paper size (Nevaci and Radu, 2012)



5. Conclusions

In linguistics, computer technologies are becoming more and more popular. The computer technology allows the development of new and modern research tools and facilitates the consulting of the results obtained in the research activity. In this paper, two new GIS based applications for the linguistic atlases publishing are presented. They are the result of the collaboration between two institutes of the Romanian Academy: Institute of Computer Science and "Iorgu Iordan – Al. Rosetti" Institute of Linguistics. The immediate result consists in the publication of the Linguistic Atlas of the Aromanian Dialect (ALAR) and the European Linguistic Atlas (ALE), but we consider this to be a first step in the use of new technologies for other linguistic and dialectal analysis.

References

- ALE (2018). European Linguistic Atlas (Atlas Linguarum Europae). https://www.lingv.ro/index.php?option=com_content&view=article&id=81&Itemid=107, last accessed Oct. 1, 2018.
- ALF (2017). Linguistic Atlas of France (Atlas linguistique de la France). <http://ligtdcge.imag.fr/cartodialect3/index.html>, last accessed Oct. 1, 2018.
- Apopei, V., Bejinariu, S.-I., Saramandu, and N., Nevaci, M. (2017). Computer-assisted editing of the Aromanian Linguistic Atlas (Editarea asistată de calculator a planșelor Atlasului Lingvistic Aromân). *Fonetică și Dialectologie*, XXXVI, 5-16.
- Bejinariu, S.-I., Apopei, V., Nevaci, and M., Saramandu, N. (2018). eALE - GIS based software tools for the European Linguistic Atlas publishing. Presented at the 51st ALE-Symposium, Atlas Linguarum Europae, June 27-30, Baia Mare, România.
- Bejinariu, S.-I. and Olariu, F.-T. (2017). Romanian Linguistic Atlases in digital format – A new approach. *Philologica Jassyensia*, XIII, 25:1, 13-23.
- Bejinariu, S.-I., Luca, R. and Olariu F.-T. (2016). A GIS based approach for information management in geolinguistics. *Memoirs of the Scientific Sections of the Romanian Academy, Computer Science*, XXXIX, 37-45.
- Bejinariu, S.-I., Apopei, V., Dumistrăcel, S., and Teodorescu, H.-N. (2009). Overview of the integrated system for dialectal text editing and Romanian Linguistic Atlas publishing. In *Proceedings of the 13th International Conference "Inventica 2009"*, June, 4-6, Iași, Romania, 564-572.
- Bejinariu, S.-I., Apopei, V., Luca, R., Botoșineanu, L., and Olariu, F.-T. (2007). Electronic Linguistic Atlas (Atlas lingvistic electronic). In Pistol, I.C., Cristea, D., and Tufiș, D. (eds.), *Resurse lingvistice și instrumente pentru prelucrarea limbii române*, Iași, 14-15 dec., Editura Universității "Al. I. Cuza" din Iași, 11-20.
- Bejinariu, S.-I., Apopei, V., Luca, R., Botoșineanu, L., and Olariu, F.-T. (2006). Tools for Linguistic Atlas consulting and dialectal texts editing (Instrumente pentru consultarea Atlasului Lingvistic și editarea textelor dialectale). In C. Forăscu, D. Tufiș, D. Cristea (eds.), *Resurse lingvistice și instrumente pentru prelucrarea limbii române*, noiembrie, Iași, Editura Universității "Al. I. Cuza" din Iași, 107-112.

- Botoșineanu, L., Olariu, F.-T., and Bejinariu, S.-I (2010). A project in Romanian linguistic cartography: The New Romanian Linguistic Atlas by Regions - Moldavia and Bukovina in electronic format, (e-NALR) - achievements and perspectives (Un projet d'informatisation dans la cartographie linguistique roumaine: Noul Atlas Lingvistic Român, pe regiuni. Moldova și Bucovina en format électronique, (e-NALR) – réalisations et perspectives). In *Proceedings of the XXVIe Congrès International de Linguistique et de Philologie Romanes*, CILFR, Sep. 8-11, València, Spain, 456-457.
- Clarke, K.C., (1986). Advances in geographic information systems. *Computers, Environment and Urban Systems*, 10(3-4), 175-184.
- EuroGeographics (2018). Administrative boundaries. © EuroGeographics, <http://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/countries#countries14>
- Junker, M-O. and Stewart, T. (2011). A linguistic atlas for endangered languages: www.atlas-ling.ca. *Proceedings of EDULEARN 11: International Conference on Education and New Learning Technologies*, July 4-6, Barcelona, Spain, 1-11.
- Lee, J., and Kretzschmar, W., (1993). Spatial analysis of linguistic data with GIS functions. *International Journal of Geographical Information Systems*, 7 (6), 541-560.
- Nevaci, M., (2016). Research of linguistic geography in the Romanian dialects of the South-Danube (Cercetări de geografie lingvistică în dialectele românești sud-dunărene). *The Annals of Ovidius University Constanta, Philology*, 425-438.
- Nevaci, M., and Radu, C.-I. (2012). Motivational aspects in the "notebook" map of European Linguistic Atlas (Aspecte motivaționale în harta "caiet" din Atlas Linguarum Europae). *Al XIV-lea Simpozion internațional de dialectologie*, Editura Argonaut Scriptor, Cluj-Napoca, 307-318.
- Nevaci, M., (2013). *Romanian identity in the Balkan context (Identitate românească în context balcanic)*. Editura Muzeului Național al Literaturii Române, București.
- Olariu, F.-T., (2017). *Variation and varieties in Romanian. Studies of dialectology and sociolinguistics (Variație și varietăți în limba română. Studii de dialectologie și sociolingvistică)*. Editura Institutul European, Iași.
- Olariu, F.-T., Olariu, V., Clim, M.-R., and Luca, R. (2016). Romanian linguistic mapping and computerization: some projects and results (La cartographie linguistique roumaine face à l'informatisation: quelques projets et résultats). In Chauveau, J.-P., Barbato, M., Fernández-Ordóñez, I. (eds.), *Actes du XXVIIe Congrès international de linguistique et de philologie romanes. Section 8: Linguistique variationnelle, dialectologie et sociolinguistique*, July 15-20, Nancy, ATILF, 193–207.
- QGIS (2018). QGIS, A Free and Open Source Geographic Information System. <https://www.qgis.org/en/site/>, last accessed Oct. 1, 2018.
- Saramandu, N., Apopei, V., Bejinariu, S.-I., Nevaci, M., and Geană, I. (2017). Digitalisation Software for ALE Project. Presented at 50 Congreso Internacional del Atlas Linguarum Europae, June, 19-23, Santiago de Compostela, Spain.
- Saramandu, N. (author) and Nevaci, M., (editor), (2014). *The linguistic atlas of the Aromanian dialect (Atlasul lingvistic al dialectului aromân)*. Editura Academiei Române, București.

- Saramandu, N. and Nevaci, M. (2006). The south-danubian Romanian dialects from the spatial perspective. *Revue des études sud-est-européennes*, XLIV, nr. 1–4, 367–383.
- Urişescu, D., (2014). The Dialect of Oas in a Computational Approach (Graiul din Țara Oaşului în perspectivă informatică). In Chivu, G., and Uţă, Bărbulescu, O., (eds.), *Ioan Coteanu – in memoriam*, Bucureşti, Editura Universităţii din Bucureşti, 369–380.
- USGS (2018). Digital Elevation Model (DEM) of Europe, Asia and Africa. © U.S. Geological Survey's Center for Earth Resources Observation and Science (EROS), <https://databasin.org/search/#query=30%20arc-second%20DEM%20of%20Europe>, <https://databasin.org/search/#query=30%20arc-second%20DEM%20of%20Asia>, <https://databasin.org/search/#query=30%20arc-second%20DEM%20of%20Africa>, last accessed Oct. 1, 2018.

LET’S TEACH COMPUTER POETRY: AUTOMATIC RHYME DETECTION

VICTORIA BOBICEV¹, CĂTĂLINA MĂRĂNDUC^{2,3}

¹*Technical University of Moldova, Chişinău*

²*Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iaşi*

³*"Iorgu Iordan – Al. Rosetii" Institute of Linguistics, Bucharest*

victoria.bobicev@ia.utm.md, catalinamaranduc@gmail.com

Abstract

Although lately it is less evident, poetry is an important part of our language. In this paper we study folk poetry annotated in the framework of UD-Romanian-Nonstandard Dependency Treebank creation. The Treebank contains Old Romanian and folk poetry from Romania and from Republic of Moldova, with morphological and syntactic annotation. The folklore part of the treebank consists of rhymed ballads and song lyrics in which their rhymes were annotated manually. The manual annotation were used in machine learning experiments with the goal to test their ability of automate rhyme detection. The experiment results demonstrated that rhyme may be detected automatically in well formed regular poems. However, folklore is less regular and contains various types of metrics and imperfect rhymes. We plan to enrich the folklore corpus and also to detect rhymes in other styles of the poetic language.

Key words — annotated corpora, dependency grammar, universal dependencies, folklore annotation, gold corpus, rhyme annotation, automate rhyme detection.

1. Introduction

A treebank is a corpus of sentences in natural language, with morphological annotation and syntactic structures in form of trees. Each branch is marking the syntactic relation and the head of each token.

The treebank used in this paper is part of the UAIC-RoDia Treebank, i.e. the Dependency Treebank for Romanian Diacronic of the Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iaşi. The corpus was also presented in (Măranduc and Perez, 2015).

A part of our corpus is formed of folklore from Romania and Moldova. The folklore is a part of the cultural heritage and we are working on its preservation by including various samples of folk art in the annotated corpus.

A sub-corpus of the treebank, affiliated with Universal Dependencies (UD) group of treebanks, is annotated in CONLLU format. In this format we introduced rhyme annotation. In overall, 941 sentences of the annotated folklore had annotated rhymes. We analysed these rhymes and applied machine learning algorithm to automatically detect rhymes.

The paper is organised as follows: after an Introduction, the Related Work section lists several works on poetry and rhyme analysis. Section 3 contains the description of the corpus (3.1), its folklore part (3.2) and manual rhyme annotation (3.3). Section 4 describes the experiments on

automatic rhyme detection and the paper is concluded by a Discussion and Conclusion section (5).

2. *Related Work*

Various treebanks for multiple natural languages have been created during last several decades in order to serve as training corpora for syntactic parsers, important tools for natural language processing.

The first attempts to create Romanian treebanks were made by Hristea and Popescu (2003) and Călăcean and Nivre (2008). In the framework of BALRIC-LING project Hristea and Popescu presented a language independent tool for corpora annotation based on the dependency grammar formalism and some annotated Romanian samples. Călăcean & Nivre (2008) used a small treebank of 4,042 sentences and less than 35,000 tokens to train the statistical MaltParser. The corpus consisted of carefully selected sentences, the aim being to contain only main clauses. Their length was 8.94 words per sentence on average.

CoRoLa (Barbu Mititelu *et al.*, 2018) is a corpus that contains contemporary Romanian texts, covering four functional language styles (belletristic, publicistic, official and scientific) and having five sections with comparable dimensions: prose, juridical, medical, academic and journalistic. The RRT Treebank in UD is part of CoRoLa, with the intention to construct a syntactic parser with good accuracy and to automatically annotate the entire CoRoLa corpus. In 2015, at its creation, the UAIC Treebank has contributed with 4000 sentences to the UD_Romanian-RRT Treebank..

UD is a portal containing now 150 treebanks in Dependency Grammar for 71 languages

¹. The annotation conventions are identical for all 150 components. The general intention of the project was the creation of an universal syntactic parser, trained on all 71 languages (Nivre, 2015).

Rhyme analysis is an area quite neglected by modern stylistics. In a PhD Thesis, Harley (1965) made "An analysis of rhyme in poetry for children". She used the computer only for some statistics.

Hirjee and Brown (2010) presented their study of imperfect and internal rhymes in rap music, having a corpus of 40 rap lyrics albums, a wrapper built around the Carnegie Mellon University Pronouncing Dictionary and a probabilistic method to calculate similarity scores for any given pair of syllables.

Greene *et al.* (2010) concentrate on statistical methods applied to the analysis, generation, and translation of poetry. By analysis, the authors meant extracting patterns from existing online poetry corpora. They admitted that not much research effort has been spent on poetry analysis. English love poetry has been studied with the task formulated as follows: given poetic lines in a known meter (such as sonnets written in iambic pentameter), assign a syllable-stress pattern to each word in each line.

They worked with the raw text written in rhymed lines and their task was to find out if these lines had classical rhyme at the ends. The poem writing with line breaks was already a great source of information as the rhymes were searched only for the last words in lines. In our case, we lost this information about line breaks which made our task much more difficult.

¹ <http://universaldependencies.org/>

Tanasescu *et al.* (2016) presented their work on poems classification by poetic meter and their own rhyme detector based on the Carnegie Mellon University Pronouncing Dictionary as the source of pronunciation information. They experimented with different classifiers and discovered that the J48 decision tree algorithm (Quinlan, 1993) provided the most accurate results. In our experiments we also obtained the best results with the J48 decision tree algorithm. Unfortunately, the described works were performed with English texts using specific English sources and tools and could not be easily ported to other languages.

Automatic scansion of Portuguese poetry was presented in (Von Wangenheim *et al.*, 2016). Scansion is the process of discovering the number of syllables in a poem's verses. The paper presents a rule-based system that performs scansion automatically, providing an analysis of rhymes in a Portuguese poem. The authors experimented with complete works of one poet which provide fairly uniform poem structures.

In 2016, Cristina Vertan conducted a project at the Hamburg University, which had as purpose the automatic annotation of rhymes, but we did not find any publication of the results.

3. Corpus and Annotation

3.1. The Corpus UD_Romanian-Nonstandard

Standard Romanian contemporary language is the subject of a large corpus mentioned in Related Work, CoRoLa (the Computational Reference Corpus for Contemporary Romanian Language, <http://corola.racai.ro/>), which also contains a treebank named RRT (Reference Romanian Treebank) with 4,000 trees from the UAIC corpus, this being the result of Augusto Perez's doctoral research (2014). But natural language is predominantly manifested by nonstandard communications.

Nonstandard, informal communication is not carried out in accordance with the academic standards of DOOM2 2005 (Vintilă-Rădulescu, 2005), which is the current language standard, but the chat or the oral communication can be as varied as inventive and creative are the interlocutors. On the other hand, it may be subject to some specific communication rules as, for example, religious texts, regional folklore or professional slang. The chat communication (Perez *et al.*, 2015) is the written aspect of familiar communication and is even more creative. Therefore, any type of social media message (comments, twitter, chat, messenger, etc.) is nonstandard communication and it grows vertiginously in volume.

The Old Romanian language, preserved in prints or manuscripts from the XVIth - XIXth centuries, is, naturally, as correct as it can be, but we cannot consider it a standard language because we do not know what standards it is related to. The latest trends of historical heritage digitalisation give the direct access to these texts to everyone who has desire to scrutiny them. A group of researchers from the Institute of Mathematics and Computer Science of Chişinău is working on this task and they have developed a tool set for optical recognition of characters from scanned texts written in the XVIth or XVIIth century with old Cyrillic letters (Colesnicov *et al.*, 2016, Cojocaru *et al.*, 2017).

The phrases were previously annotated in the UAIC conventions established in 2007 when the FII-UAIC treebank has been created. This format is the pivot for UD format. The rule-based tool TREEOPS (Colhon *et al.*, 2017) automatically transforms the annotation in the UD format.

The UD annotation is made in the CONLLU format, standard for all 140 corpora for 71

languages referred on the UD site. Its format is textual with one word per line and all annotation in tab delimited columns. Column 1: the word id; column 2: word form as it appears in the text; column 3: the lemma; columns 4-6: morphological information in the form of: U-POS-tag (fourth column) X-POS-tag (fifth column) and then the expanded explanation of the X-POS-tag in the sixth column; columns 7-8 contain dependency relation description: the id of the head and the name of the dependency relation. Column 9 is reserved for enriched syntactic analysis, and column 10 contains information about the (non) existence of the space between the current token and any other type of information. The rhyme is indicated in this column in the folklore corpus.

3.2. Folklore Annotation

In this study we used the folklore sub-corpus, annotated in the Universal Dependencies with additional annotation of rhymes.

Folklore is a part of cultural heritage, a phenomenon that quickly disappears as everyone has access to the written culture. The old books will last for several hundred years, but in a few years, the last elderly people who know by heart once orally transmitted texts will pass away and this oral cultural heritage will be lost.

Some folklore has been collected and printed in Romania and Moldova in the 19th and 20th centuries. 693 phrases introduced now in UD were extracted from the study of Bostan (1998) with popular texts from Moldova and Bucovina. 208 phrases have been annotated and introduced in the UD from the collection of folklore texts of different genres from the villages of the central region of the Republic of Moldova, called *Codri*, collected during 1962-1972, in folk expeditions organized by the Academy of Sciences of Moldova, the folkloric section, and published in (Botezatu *et al.*, 1973). The book was printed in Cyrillic letters, thus it had been scanned, then interpreted using an OCR tool, and finally transposed into the Latin alphabet (Bobicev *et al.*, 2016).

By now, we have annotated only 230 folklore phrases from Romania, disparate quotes from (Brăiloiu *et al.* 1978), collected between 1885-1889 and between 1936-1977, texts from (Bălășel, 1967) between 1869-1902, from (Datcu, 1968), collected between 1901 and 1964, and texts from (Amzulescu, 1964), collected between 1883-1939.

3.3. Rhyme Annotation

Finally, we experimented with two files with annotated folklore: 902 sentences of Moldavian and 230 sentences of Romanian folklore. Only 754 of the 902 Moldavian sentences and 187 of the 230 Romanian sentences had rhymes. Thus, we experimented with these 941 sentences. Several types of rhymes were annotated. The classical one, when two words rhymed, was marked paired: Type=Paired. Below is presented an example.

According to a classic definition, rhyme is the similarity of the sounds at the end of some verses, beginning with the accented vocals (Constantinescu, 1973).

Annotation of the rhyme was done manually on the tenth column, indicating the id of the word which the annotated word is rhymed with and the type of rhyme. Below is a fragment of CONLLU text annotated with rhyme between the word id=12 (bănat) and the word id=16 (sat). The annotation of the rhyme is marked here in bold. The text is: *Să nu fie cu bănat, Unde sună scripca în sat?* (meaning no offence, where the violin sounds in the village?):

```
8 SăsăPART QsPartType=Sub 12mark _ _
9 nunuADV QzPolarity=Neg 12advmod _ _
```

LET'S TEACH COMPUTER POETRY: AUTOMATIC RHYME DETECTION

```

10 fie fi VERB Vmsp3 Mood=Sub|Person=3|Tense=Pres|VerbForm=Fin 12 cop _
11 cucu ADP Spsa AdpType=Prep|Case=Acc 12 case _ _
12 bănat bănat NOUN Ncmsrn
   Case=Acc,Nom|Definite=Ind|Gender=Masc|Number=Sing 6 ccomp _
   Rhyme=ID18|SpaceAfter=No|Type=Paired
13 , , PUNCT COMMA _ 15 punct _ _
14 Unde unde ADV RwPronType=Int,Rel 15 advmod _ _
15 sună suna VERB Vmis3s
   Mood=Ind|Number=Sing|Person=3|Tense=Past|VerbForm=Fin 6 parataxis
16 scripca scripcă NOUN Ncfsry
   Case=Acc,Nom|Definite=Def|Gender=Fem|Number=Sing 15 nsubj _ _
17 în în ADP Spsa AdpType=Prep|Case=Acc 18 case _ _
18 sat sat NOUN Ncmsrn
   Case=Acc,Nom|Definite=Ind|Gender=Masc|Number=Sing 15 obl _
   Rhyme=ID12|SpaceAfter=No|Type=Paired

```

In the Folk poetry, the types of rhymes are only Paired and Mono (or Multiple) indicated in the case of the same word at the end of verses, Type=Rep. (repetitive). There are also Type=Imperf. (imperfect). The statistics for the rhymes is presented in Table 1.

Constantinescu (1973) considers that the folklore specific rhyme is the Type=Mono, or Multiple (Repetitive). However, the most common is the Paired rhyme; 65% of rhymes are of this type and only 15-20% of total number of rhymes are of the multiple type.

As the corpus of folklore collected in the Republic of Moldova is newer than the one collected in Romania, it is influenced by the literature and we have found some cases of more complex types of rhymes, that are not specific to the folklore. It is still a proof of the dissolution of the traditional model. We present below two examples, one of Type=Cross (*ânt/it/ât/it*) and one of Type=Imbr. (imbricated) (*at/ui/ui/at*):

<i>Sub o țărână de pământ</i>	<i>Under an earthy soil</i>
<i>Este-un bujor înflorit</i>	<i>There is a blooming peony</i>
<i>Toată lumea l-a urât</i>	<i>Everyone hated him</i>
<i>Dumneata l-ai îndrăgit</i>	<i>You loved him</i>
<i>Casă mândră am gătat</i>	<i>A proud house I made ready</i>
<i>Cu dulceața grâului,</i>	<i>With the sweetness of wheat,</i>
<i>Cu lacrima vinului,</i>	<i>With the tear of wine,</i>
<i>Cu-n colac de grâu curat</i>	<i>With a pure wheat loaf</i>

In order to verify whether the rhymes of these verses are regular, we compared the distance between the rhymes, which should be approximately equal (4-5 words), then we compared the total distance between the rhymes with the total length of the sentence. If the total distance is much smaller than the length of the sentence, it appears that the phrase has a portion without rhymes.

Table 1. Types of rhymes and their frequencies in the corpus

Rhyme type	Frequency in Moldavian texts	Frequency in Romanian texts
Paired	840	171
Mono	206	35
Repetitive	84	4
Imperfect	150	43
Intern	8	7
Cross	5	0
Imbr	3	0

In these statistics, we have approximately calculated the correct distance among the rhymes. We should calculate the number of syllables in the popular verses, which should be 6 or 8. Considering that the average number of syllables in a word in popular poetry is two, we have approximated that four words would represent 8 syllables, a correct distance. Actual statistics of distances between rhymed words measured in words is presented in Table 2.

Table 2. Frequency of distances between rhymed words measured in no. words

Number of words	6	5	4	7	3	2	8	9	10
Frequency in corpus	404	381	294	224	137	91	76	23	13

4. Automatic Rhyme Detection

The most obvious rhyme feature is the match of words' endings at the end of the line. Therefore, we used this as a first feature set for machine learning. Taking into consideration that we had already annotated texts in the one-word-per-line format, we lost the classical verse's arrangement in rhymed lines. Thus, we compared every word of a sentence to every preceding word in the same sentence, starting with the distance between them equal to two words and increasing it to the distance of ten words.

The set of features for each pair of words consisted of:

- (1) a feature which was equal to 1 if these words matched, and 0 otherwise;
- (2) a feature which was equal to 1 if the last letters of these words matched, and 0 otherwise;
- (3) a feature which was equal to 1 if the last two letters of these words matched, and 0 otherwise;
- (4) a feature which was equal to 1 if the last three letters of these words matched, and 0 otherwise;
- (5) a feature which was equal to 1 if the last four letters of these words matched, and 0 otherwise;

- (6) The last feature was the distance between these words measured in no. words.
- (7) Finally, the class was 1 if these two words rhyme and 0 otherwise.

In this way, we obtained 122,201 pairs of words which were converted in instances with sets of above described features. However only 1,598 word pairs rhymed; this was 1.3% of the whole set. With such an unbalanced data set, machine learning was extremely difficult as the baseline (when all instances were attributed to the majority class) was 98.7% accuracy.

We used the Weka² toolkit in our experiments. Weka is a collection of machine learning algorithms for data mining tasks which includes various Bayes based methods, Support Vector Machine implementation, Decision Trees, Lazy Learning algorithm variations and many other methods. We used 10-fold cross-validation in all our experiments.

Decision tree slightly improved the baseline result, detecting 597 rhymes of 1,598, but attributing rhyme class to 382 non rhyming word pairs. The most important features in this case were (in decreasing order): last 2 letters match, word match, distance, last 3 letters match, last 4 letters match.

To improve the accuracy of rhyme detection, we have to reduce the number of word pairs that do not rhyme. We noticed that we could still detect the poems line breaks by the words which start with a capital letter in the middle of sentences. Thus, we could check only words preceding those with capital letter. With such a condition implemented, we obtained 5,353 word pairs. Unfortunately, extracting only these words which precede words with capital letters we managed to obtain 1,207 pairs of rhymed words; in some cases the capitalization of the first letter of words at the beginning of the new line of the verse was lost in the process of part of speech and syntactic annotation.

With this data, a decision tree algorithm correctly detected 878 rhymes of 1,207, reaching an F-measure of 0.9 in comparison with the baseline of 0.74.

Finally, we added a feature that in our opinion was one the most important: distance between rhymes measured in syllables. The task of syllable calculation in Romanian words was difficult due to the phenomena of diphthongs, triphthongs, and hiatus. Diphthong is a combination of two adjacent vowel sounds and triphthong is a combination of three adjacent vowel sounds within the same syllable. Examples of diphthongs: *doi*, *cuib*, *soare*, *deal*, *iute*; examples of thriphthongs: *doreau*, *beau*, *inimioară*, *lăcrămioară*. Hiatus refers to two vowel sounds occurring in adjacent syllables, with no intervening consonant. Examples of hiatus: *po-ezi-e*, *totdea-una*, *a-er*, *a-ur*. While a general rule says that the number of syllables in a word is equal to the number of vowels, in Romanian this rule is wrong in many words. Thus, we had to implement a simple rule-based algorithm for syllable count. We listed several most frequent diphthongs and thriphthongs; added the rule of collapsing for words connected with hyphen as, for example, *și-a*, *că-i*, *ți-am* and created a dictionary for less regular words indicating their number of syllables. Finally, we obtained a statistics on the number of syllables between rhymes, presented in Table 3.

² <https://www.cs.waikato.ac.nz/ml/weka/>

Table 3. Frequency of distances between rhymed words measured in syllables

Number of syllables	8	7	9	6	5	4	10	11	12
Frequency in corpus	909	790	347	174	172	165	107	43	34

It is seen from the table that, albeit the most frequent is the classical distance of 8 syllables, this regular measure is respected in approximately 75% cases (if we consider 7 and 9 regular meter as well). In other cases, the irregular variation of distances affected the performance of our automatic rhyme detection.

5. Discussion and Conclusion

Long history of computational linguistics and natural language processing research demonstrated that our human language is quite difficult to formalize and process by computers. Poetry is an even more difficult phenomenon, and folk poetry is even more challenging for automatic processing.

On the contrary, on the case of literary poetry we expect more or less regular meter and matching rhymes at the ends of poem lines. In our folklore corpus approximately a quarter of verses were far from regular. More than 10% of rhymes were annotated as Imperfect (see Table 1).

Example of such without-rhyme verses are:

<i>Murgii-mpodobiți</i>	<i>Horses adorned</i>
<i>Cu șea de argint</i>	<i>With silver saddles</i>

Example of Imperfect rhyme
(ad/at):

<i>De la frunza cea de brad,</i>	<i>From the leaves of fir,</i>
<i>De la puiul cel lăsat</i>	<i>From the little one left behind</i>

It can be seen from these examples that here rhymes are almost non-existent. As a result, most of the Imperfect rhymes had not been detected by the automatic method.

Another problem is the distance. Below are presented several examples with various non-standard metrics:

<i>Sus la ușă-s flori de rujă,</i>	<i>Up at the door rose flowers</i>
<i>Jos la prag ȳs flori de mac,</i>	<i>Down at the threshold poppy</i>
	<i>flowers</i>
<i>Pe podele-s viorele,</i>	<i>On the floor violets,</i>
<i>Pe afară-s flori de vară,</i>	<i>Outside summer flowers</i>
<i>La porțiță-i hulubiță.</i>	<i>On the gate a dove</i>

In the example above rhymes are inside the rows and there is no rhyme between rows. Also, many verses are written in free form where some parts are rhymed and some have no even

regular measures. All these affect the performance of the automatic rhyme detection. However, a part of rhymes can be detected automatically, imposing that the result of the automatic annotation be verified by humans.

In future, we plan to explore various types of metrics used in folklore and to make statistics on word endings that rhyme. We will look for dictionaries with phonetic information, division of words into syllables and word stresses. This information will help us to detect rhymes with more confidence.

References

- Amzulescu, Al. I. (1964). *Romanian folk ballads*. Bucharest, Publishing House for Literature, 1964.
- Barbu Mititelu, V., Tufiş, D., Irimia, E. (2018). The Reference Corpus of the Contemporary Romanian Language (CoRoLa). In *Proceedings of Language Resources and Evaluation Conference - 2018*, Japan, p. 1178-1185.
- Bălăşel, T., (1967) *Folklore from Oltenia and Muntenia. Selected texts from unusual collections*. Vol. II. Literature Publishing House.
- Bobicev, V., Bumbu, T., Lazău, Maxim, V. and Istrati, D. (2016). Folk poetry for computers: Moldovan Codri's ballads parsing. In *Proceedings of the 12th International Conference Linguistic Resources and Tools for Processing the Romanian Language*, ConsILR, p. 39-50.
- Bostan, G. C. (1998). The Romanian folk poetry in the Carpathian-Nistrian space, Cantes Publishing House, Iaşi.
- Botezatu G., Cirimpei V.A., and Ciobanu I.D. (1973). *Folklore from the Wood's region*, Science's Publishing House, Chişinău.
- Brăiloiu, C., Comişel, E. and Găluşcă-Cârşmariu, T. (1978). *Folklore from Dobrogea*. Bucharest, Minerva Publishing House.
- Călăcean, M., Nivre, J. (2008). Data-driven Dependency Parsing for Romanian, *Acta Universitatis Upsaliensis*, p. 65-76.
- Cojocaru, S., Colesnicov, A., Malahov, L. (2017). Digitization of old Romanian texts printed in the Cyrillic script. In *Proceedings of Digital Access of Textual Cultural Heritage*, Göttingen, ACL Publisher, p. 143–148.
- Colesnicov, A., Malahov, L., Bumbu, T. (2016). Digitization of Romanian printed texts of the 17th century. In *Proceedings of the 12th International Conference Linguistic Resources and Tools for Processing the Romanian Language*, ConsILR, Iaşi, A.I.Cuza University Publishing House, p. 1–11.
- Colhon, M. Măranduc C., Mititelu, C. A Multiform Balanced Dependency Treebank for Romanian, in *Proceedings of Knowledge Resources for the Socio-Economic Sciences and Humanities, (KnowRSH), Varna, Bulgaria September 8, 2017 workshop at the Recent Advances in Natural Language Processing (RANLP)*, p. 9-19.
- Constantinescu, N. (1973). *Rhyme in Romanian folk poetry*, Minerva Publishing House, Bucharest.
- Datcu, I. (1968). *Folklore from Oltenia and Muntenia. Selected texts from unusual collections*. Vol. III. Literature Publishing House.

- Harley, A. V. (1965). An analysis of rhyme in poetry for children. Master of Arts Thesis, The University of British Columbia.
<https://open.library.ubc.ca/cIRcle/collections/ubctheses/831/items/1.0078088>
- Hirjee, H. and Brown, G.D. (2010). Using automated rhyme detection to characterize rhyming style in rap music. In *Empirical musicology review*, 5. 10.18061/1811/48548.
- Hristea, F. and Popescu, M. (2003). A Dependency Grammar approach to syntactic analysis with special reference to Romanian. in Hristea, F., Popescu, M. (eds.) *Building awareness in language technology*, University of Bucharest Press, Bucharest, p. 9-34.
- Greene, E., Bodrumlu, T., and Knight, K. (2010). Automatic analysis of rhythmic poetry with applications to generation and translation. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, p. 524-533.
- Mărănduc, C., and Perez, A.-C. (2015). A Romanian Dependency Treebank. In the *International Journal of Computational Linguistics and Applications* Vol. 6 No. 2 July-December, p. 25-40.
- Nivre, J. (2015). Towards a Universal Grammar for natural language processing. Computational Linguistics and Intelligent Text Processing.
- Quinlan, R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, San Mateo, CA.
- Perez, A.-C. (2014). Linguistic Resources for Natural Language Processing, PhD thesis, Alexandru Ioan Cuza University of Iași.
- Perez, A.-C., Mărănduc, C., Simionsecu, R. (2015). Including social media – a very dynamic style, in the Corpora for Processing Romanian Language. In D. Trandabăț and D. Gîfu (eds.): *Proceedings of EUROLAN - 2015*, CCIS 588, Springer International Publishing, Switzerland, p. 1–15.
- Tănăsescu, C., Paget, B., Inkpen, D. (2016). Automatic classification of poetry by meter and rhyme. *Florida Artificial Intelligence Research Society Conference*, North America, March³.
- Wangenheim, A. von, Santos, A. L. dos, Mitmann, A. (2016). A system for the automatic scansion of poetry written in Portuguese.
- Vintilă-Rădulescu, I. (2005 - coord.) *DOOM2. Orthographic, orthoepic and morphological dictionary of the Romanian language*. 2nd Edition revised and added. Enciclopedic Universe, Bucharest

³ Available at: <<https://www.aaai.org/ocs/index.php/FLAIRS/FLAIRS16/paper/view/12923>>

EXTRACTING ACTIONS FROM ROMANIAN INSTRUCTIONS FOR IoT DEVICES

BIANCA I. NENCIU¹, STEFAN RUSETI¹, MIHAI DASCALU^{1,2}

¹ *University Politehnica of Bucharest, Computer Science Department, 313 Splaiul
Independentei, 060042, Bucharest, Romania*

² *Academy of Romanian Scientists, 54 Splaiul Independenței, 050094, Bucharest, Romania
bianca_ioana.nenciu@stud.acs.upb.ro, {stefan.ruseti, mihai.dascalu}@cs.pub.ro*

Abstract

In recent years, multiple vendors have started manufacturing Internet-of-Things devices that can interact with users through natural language. Thus, many products that are advertised as personal assistants have emerged and are relying on artificial intelligence and machine learning techniques to process natural language. In the most frequently encountered scenarios, speech is first converted to text using Speech-to-Text algorithms (STT), and afterwards it is processed by a Natural Language Understanding (NLU) engine to extract actions from text. In this article, we combine the concept of microworlds with an NLU engine adapted for Romanian language in order to create a system that is capable to understand and handle user's inputs. This project is built on top of Snips, an open-source intent classifier, developed to run on any device, which already supports English, French, German, Spanish, Korean, Japanese and Chinese.

Key words — action extraction, Natural Language Understanding, Romanian language.

1. Introduction

Internet-of-Things (IoT) devices are more and more popular in our daily life. What makes them so convenient and easy to use is exactly their human-interface. Speaking with a device is much easier than using a remote control, a computer or a smartphone. In contrast, the classic programming approaches restrict the list of potential users to people who have some technical aptitudes. Controlling a home appliance by using spoken words is far simpler and comes natural to humans who are used to speak on a day-to-day basis. There are numerous tools, both open-source and commercial, that support Speech-to-Text (STT) functionalities. One of the most popular open-source solutions is CMUSphinx (n.d.), having excellent results even when compared with some of the most popular commercial systems like Google's Cloud Speech-to-Text service (n.d.), Wit.ai by Facebook, Watson Speech-to-Text by IBM (n.d.).

However, most available solutions for the second part of the pipeline which requires Natural Language Understanding (NLU) work with more popular languages (e.g., English, French, German, Spanish, etc.) and do not offer any kind of support for Romanian. This article introduces such an engine that is able to process Romanian language, and also presents a general approach on how support for other languages may be implemented.

This paper continues with the description of existing solutions that turned out to be insufficient for Romanian language. The following sections present the corpora used for training our model, the implementation of the proposed solution together with corresponding

results. The paper ends with analyses and discussions of results, as well as ways to further improve the performance of our system.

2. Related Work

Two of the most popular open source solutions that serve as an engine for building personal assistant-like systems are Rasa NLU (n.d.) and Snips (n.d.). Both of these projects are libraries that expose a simple interface trained for a specific micro-world (i.e., specific domain or usage scenarios), and then take any related phrase as input, understand it, and output an actionable item along with its parameters. For example, “Turn on the light in living room” would be understood as an action of type “Turn on” having two parameters “The light” (what) and “living room” (where).

One of the biggest differences between the two projects is that Snips has a dynamic processing pipeline and uses as little resources as possible. This design decision makes customizations easy, further increasing the range of devices where it can run, including many small-scale IoT devices with limited local resources. In contrast, Rasa NLU is a more mature solution offering more features and is built on top of spaCy (n.d.) that integrates Natural Language Processing (NLP) pipelines for each supported language. Both systems are written in Python which leads to early prototypes and quick iterations.

With regards to closed source projects, we only analyzed Wit.ai (n.d.), the online engine developed by Facebook that has beta support for Romanian language. All other solutions did not provide any interface that could have been used for training on a specific micro-world in Romanian language; thus, it would have been an unfair comparison.

3. Method

3.1. Corpus

Our corpus consists of a list of phrases representing a specific action and the associated labels that could be used in the learning process. Ideally, all of the actions in the corpus are a part of a single micro-world. At the time of development of this project, no such open source corpora were available for Romanian language, so a specific corpus had to be constructed. Therefore, our corpus was specifically designed for a home assistant and contained around 250 phrases representing 35 actions, falling under 12 categories like controlling home appliances, navigation and weather, calendar and time management, etc. It often happens that the phrases from the same category are very similar and differ only by few words, which only made the classification process more difficult.

3.2. Architecture

The proposed solution is built on top of Snips NLU engine presented in Figure 1 which extended to offer support for Romanian language. Starting with English and French as baselines, Romanian has also complex grammar rules and one differentiator is that Romanian language’s syntax leads to more inflections. For example, the most common verbs (i.e., “to do” and “a face”) have the same meaning in English and respectively Romanian. When conjugated, “to do” becomes “I do/ You do/ He, she, it does/ We do/ You do/ They do” – so only two forms of the verb. In Romanian, the same verb is conjugated as “eu fac/ tu faci/ el,

ea fac/ noi facem/ voi faceti/ ei, ele fac” – 4 forms, twice as many as English. The same happens for other tenses and grammatical cases.

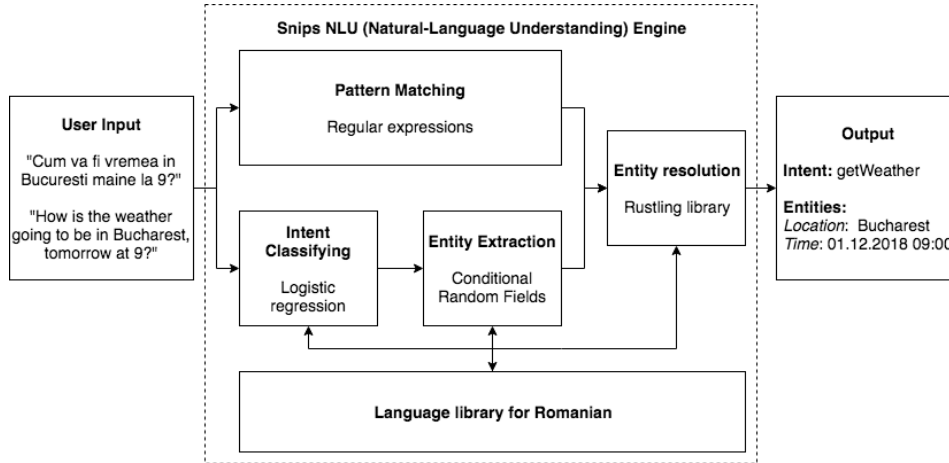


Figure 1: Overall architecture of Snips

Some of the most important components of the NLU pipeline are the two intent classifiers, the entity parsers and the library for Romanian vocabulary. A simple and fast intent classifier relying on regular expressions is used to match user’s inputs. In case of a positive match, the result is passed directly to the entity resolution component. Otherwise, the probabilistic classifier is used to classify the intent. After the intent is found, an entity extractor is used to identify entities from the original text. Afterwards, the entity resolution library (i.e., Rustling) parses the entities and everything is put together and returned as a single JSON object representing user’s overall intent.

3.3. Intent classifiers

3.3.1. Deterministic Classifier

The deterministic classifier implements a simple idea of iterating over each phrase from the dataset and building regular expressions. However, it turns out that, if the training dataset is large enough or the possible ways of expressing a command is limited, then the algorithms actually performs extremely well, offering a high accuracy and performance in terms of processing speed. For example, the phrase “Turn on the light in [room]” would be learnt as “Turn on the light in (.*)”. In addition, this classifier is easy to implement, light-weight and easy to debug.

Once the classifier is constructed, using it is as simple as iterating over all the regular expressions and trying to match the input with the current expression. Entities are automatically extracted by the groups defined in the regular expressions. Any result returned by this classifier is automatically considered correct and is assigned a confidence score of 100%.

3.3.2. Probabilistic Classifier

Although the deterministic classifier is a viable option in most cases, it is quite rigid, and it only works well only when the training examples match the input. In other cases, it offers no result and a second probabilistic classifier is used to cover them.

The probabilistic classifier contains a term frequency – inverse document frequency (TF-IDF; Manning, Raghavan, & Schütze, 2008) vectorizer and a logistic regression classifier with stochastic gradient descent training. The purpose of the TF-IDF vectorizer is to convert the input phrase to a numerical input that can serve as input to the logistic regression classifier. As soon as intent has been identified, a Conditional Random Field (CRF) classifier (Lafferty, McCallum, & Pereira, 2001) is used to recognize and extract entities out of the input phrase.

3.4. Entity extraction and resolution

Another important part of this engine is entity extraction and resolution, which is handled by CRFs. Conditional Random Fields are usually used in Machine Learning for pattern recognition and chain CRFs can be used to parse natural language.

In order to use CRF, several elements are required:

- attribute extractor to convert words to a numerical representation;
- weights for each attribute, which are computed during the learning process;
- labels used during learning (e.g. “room” in “Turn on the light in (kitchen) [room]”).

Attribute extractors are functions that take text input as text and return numbers that represent text characteristics. For example, it can be a function that checks if a word starts with an upper-case letter. For this system, a few extractors are being used:

- `ngram(n)`: returns an attribute built using n adjacent words;
- `is_digit()`: returns a non-null value if word is a number;
- `is_first()`: returns a non-null value if this is the first word;
- `is_last()`: returns a non-null value if this is the last word;
- `shape_ngram(n)`: returns an attribute built using the shape (lower case, upper case, title case, reversed title case) of n adjacent words.

Once entities were successfully extracted, the system can proceed to resolve those entities and convert them to an internal representation. For example, an entity that could represent the time as “2 minutes” would be converted to a structure containing an integer (2) and time unit (minutes). For this purpose, Rustling (n.d.) is used, which is a Rust implementation of Facebook’s Duckling library (<https://github.com/facebook/duckling>). This library is a collection of regular expression that can parse multiple types of entities, like numbers, time, duration, intervals, dates, etc. The design of this library permits using recursive regular expressions. For example, an interval may be defined as “from [time:beginning] to [time:end]”.

3.5. Library for Romanian vocabulary

In addition to the previous processing stages, a library of Romanian words was used to improve the system’s knowledge base in order to increase its accuracy. This library was generated using the DexOnline.ro database (n.d.), a popular Romanian dictionary. The dictionary itself provides a comprehensive list of fields, but the most important consisted of word definitions, alongside with word types, popularity and inflections. After several iterations, the library consisted in a few files containing:

- top 10,000 most used words;
- inflections of top 10,000 words;
- top 2,000 verbs and lexemes;
- stop words (i.e., words with no contextual information);
- randomly generated word lists (i.e., noise used for data augmentation and training the probabilistic intent classifier).

4. Results

In order to train and evaluate the system, the corpus mentioned in the previous section was split into two parts: 80% for training and 20% for test. After the system was trained, the remaining 20% of the data was used to measure precision and recall, together with corresponding F1-scores. Even though the corpus could be considered rather small, we reached the conclusion that it was adequate for evaluation purposes since there are not that many alternative manners of expressing some intent.

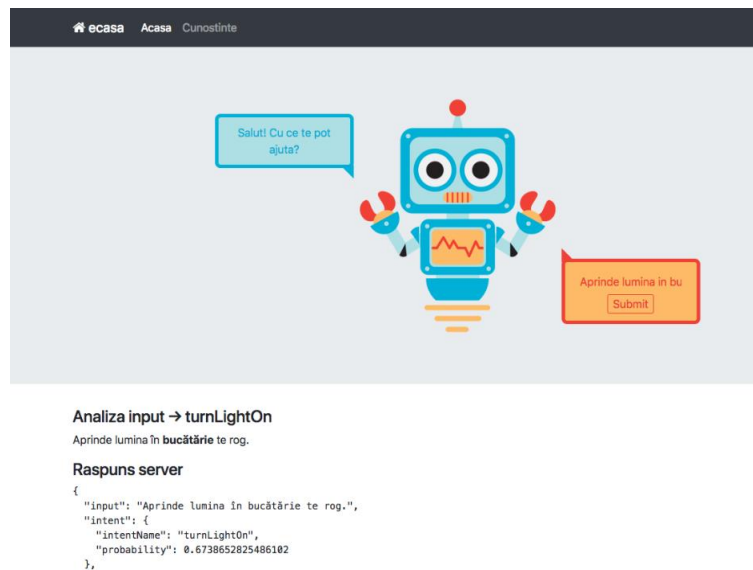


Figure 2: User interface using NLU engine.

In order to match a real-world situation, we created a basic user interface that serves as a dashboard for a home assistant. This interface offered us a way of improving the knowledge base of the system, query its engine, visualize results and debug. The screen capture from Figure 2 depicts the results for the input query “Aprinde lumina in bucatarie, te rog” (en. “Please turn on the light in kitchen.”). The action was correctly classified as “turnLightOn” with a probability of 67% and the system correctly identified “bucătărie” (en. “kitchen”) as the corresponding entity.

During evaluation, the system managed to score a 99% on F1-measure scale and produced the confusion matrix from Figure 3. The confusion matrix itself shows promising results, with few exceptions for some similar actions like “turn TV on” and “turn TV off” which had the lowest F1 scores of 80%, respectively 94%. Moreover, the system was capable of reaching a perfect score of 100% when reducing the dataset to only 8 actions, as it can be seen in Figure 4.a (lighter shades outside the main diagonal denote problems). In comparison, Wit.ai only

scored 78.75% in the F1-measure. The confusion matrix in Figure 4.b shows that Wit.ai had difficulties even in classifying unambiguous intents like “search address”.

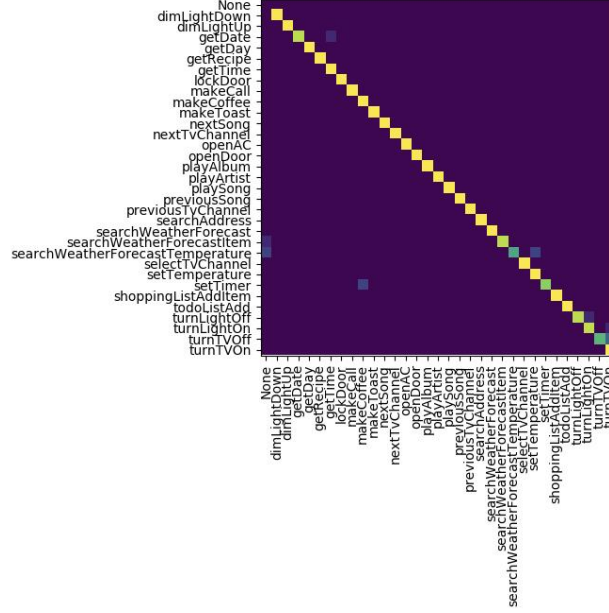


Figure 3: Snips confusion matrix when used with the entire dataset

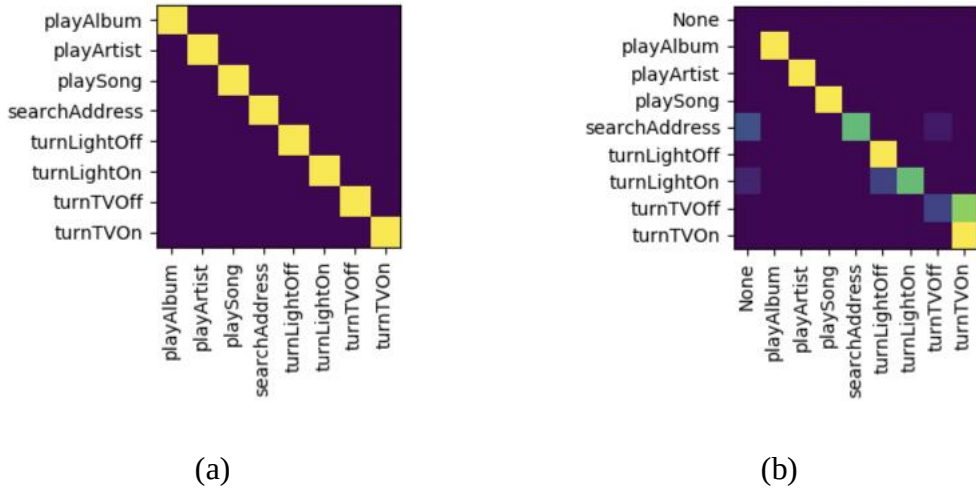


Figure 4: a) Snips vs. b) Wit.ai confusion matrices.

5. Discussions

Our results show that even a small amount of data is enough to classify intents for a specific micro-world. This most probably happens because there is a limited number of phrases that can be used to express an intent, and most alternatives are similar. An interesting remark is that the system still manages to differentiate between phrases like “Turn the TV on” and “Turn the TV off”, even if they differ by only one preposition. Moreover, even though language libraries are not mandatory for classifying intents, their usage proved to improve accuracies on average by 30% and overall F1-score between 10% and 15%. In addition, the language library proved to be useful and necessary for extracting entities. Romanian phrases

were mostly classified correctly with the default English library, whereas entities were almost always extracted in an incorrect manner, often including too many surrounding words, or too few. Without the specific Romanian vocabulary, the system had no understanding regarding the syntactic boundaries of entities.

Another significant result is the overall performance of the system. In contrast to close source alternatives, our project runs locally requiring few resources. Intent classifying with Wit.ai is easier to implement, but the processing takes a considerably longer time because the ideal round trip time (about 150ms) is already about 100 times slower than the usual processing time of Snips (1-2ms).

6. Conclusions

This paper introduced an NLU engine for Romanian texts that is capable to quickly classify intents with high accuracy for a given micro-world. Even though the results are promising for small micro-worlds, we noticed that the overall accuracy decreases when the system is trained on a larger corpus.

We consider that one of the most important improvements the NLU engine would benefit from consists of using different classifiers and tweaking the parameters of existing ones. Another element to be improved is the library of Romanian texts. For the current version of this system, the vocabulary was built using literary texts that do not contain even basic concepts like “kitchen”, which are actually used in the tested micro-world.

Our approach was evaluated independently from a speech-to-text engine, which could induce additional errors, while our training set only contains correct phrases that do not contain any typos. Additional testing of the method in a scenario closer to a real-world situation is required.

Acknowledgments

This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS – UEFISCDI, project number PN-III 72PCCDI / 2018, ROBIN – “Roboții și Societatea: Sisteme Cognitive pentru Roboți Personali și Vehicule Autonome”.

References

- Cloud Speech-to-Text. (n.d.). Retrieved October 1st, 2018, from <https://cloud.google.com/speech-to-text/>
- CMUSphinx: Open Source Speech Recognition. (n.d.). Retrieved October 1st, 2018, from <https://cmusphinx.github.io/>
- DexOnline - Database schema. (n.d.). Retrieved October 1st, 2018, from <https://github.com/dexonline/dexonline/wiki/Database-Schema>
- Lafferty, J., McCallum, A., & Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML '01 Proceedings of the Eighteenth Int. Conf. on Machine Learning*, 282–289. Retrieved February 20, 2018, from https://repository.upenn.edu/cis_papers/159

- Manning, C. D., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- Rasa NLU: Language Understanding for chatbots and AI assistants. (n.d.). Retrieved October 1st, 2018, from <https://rasa.com/docs/nlu/>
- Rustling: Rust implementation of Duckling. (n.d.). Retrieved October 1st, 2018, from <https://github.com/snipsco/rustling>
- Snips - Using Voice to Make Technology Disappear. (n.d.). Retrieved October 1st, 2018, from <https://snips.ai/>
- spaCy - Industrial-strength Natural Language Processing in Python. (n.d.). Retrieved October 1st, 2018, from <https://spacy.io/>
- Watson Speech to Text. (n.d.). Retrieved October 1st, 2018, from <https://www.ibm.com/watson/services/speech-to-text/>
- wit.ai - Natural Language for Developers. (n.d.). Retrieved October 1st, 2018, from <https://wit.ai>

IDENTIFYING FAKE NEWS ON TWITTER USING NAÏVE BAYES, SVM AND RANDOM FOREST DISTRIBUTED ALGORITHMS

CIPRIAN-GABRIEL CUȘMALIUC, LUCIA-GEORGIANA COCA, ADRIAN IFTENE

*“Alexandru Ioan Cuza” University of Iasi, Faculty of Computer Science
{cusmuliuc.ciprian.gabriel, coca.lucia.georgiana, adiftene}@info.uaic.ro*

Abstract

In our days using social networks like Facebook or Twitter in order to find news is something usual. Users from these networks create a lot of content: posts (with text, images, videos, and links), comments, likes, but also redistribution of information with retweet option or with simple copy/paste operations. Nowadays, a common problem is the detection of fake users and of fake news and it is often difficult to distinguish the credibility of news that appears in a social networking. Sometimes users are in the position of believing the certain news at their first appearance, only because the information is written in the network and they do not check to see if it's true or not. The present paper aims to analyze different techniques of detecting fake news; their performance and how we can do fine tuning in order to improve the actual results.

Key words — Fake news, Naïve Bayes, Random Forest, SVM, Twitter.

1. Introduction

The popularity of social networks has increased significantly in recent years and reading news on these social networks has become a natural activity for all users. The news is instantly transmitted to these networks, which are read quickly, marked with opinions (see Facebook), retransmitted (retweet on Twitter, share on Facebook) without having to check many times whether they are true or false news.

Over the years there have been different approaches to the classification of fake news. From *linguistic approaches* (with machine learning) to *social network approaches* researchers apply different techniques to identify fake news (Conroy *et al.*, 2015), (Atodiresei *et al.*, 2018). In *linguistic approaches*, statistics on n-grams were created and analyzed to identify fake information (Hadeer *et al.*, 2017), or sentences were transformed into more advanced forms of information representation (such as parsing trees), which then analyze probabilities attached to identify anomalies (Perez-Rosas *et al.*, 2018), or it analyses semantically the contents of a user's statements, constructs pairs of the attribute form: descriptor and calculates compatibility scores (Shu *et al.*, 2018), or relations between the linguistic elements are built, which help determine the proximity to the centers of truth or deception (Popoola, 2017; Rubin and Lukaianova, 2018), or use neural networks that identify fast the fake news (Sneha *et al.*, 2018), or similar to our approach, SVM classifiers or Naïve Bayesian-type classifiers are used to predict future clutter-based fraud and distances in (Rubin *et al.*, 2016; Singh *et al.*, 2018). In *social networking approaches*, knowledge networks are exploited to identify the lie (Idehen, 2017), or the fact that users are forced to authenticate when using the social network, provides increased confidence to the data that appears here (Shu *et al.*, 2018; Wu and Liu, 2018).

Our proposal is similar with work presented in (Strehl *et al.*, 2000), but we increase the data set and the quality of it. The authors used SVM and they created a dataset of 345 valid news articles, dataset including an equal number of news reports from three well known and largely respected news agencies: *National Public Radio*, *New York Times*, and *Public Broadcasting Corporation*. The most important features used by their SVM algorithm are the following: *author*, *published*, *title*, *text*, *language*, *crawled*, *site_url*, *country*, *thread_title*, *spam_score*, *main_img*. Their experiments obtained an overall accuracy of 0.87. Additional to increasing the data set; in our experiments we implement more algorithms, like Random Forest, SVM and Naïve Bayes in order to see which is best for this problem.

In Chapter 2 we will see details about our data and the implemented algorithms. Chapter 3 presents the system architecture and results obtained with the algorithms. Chapter 4 presents relevant use cases and analysis performed on tweets with correct classification and those with incorrect classification. The last chapter contains the conclusions of this paper and future work.

2. Proposed solution

2.1. Data set

In order to apply our classification techniques, we built a dataset that was annotated manually with the help of human annotators. All the data was collected from Twitter from a period of one month, in August 2011. The data was not doubly annotated, the criteria used by the humans in order to label a tweet were related to the type of tweet, if the tweet was **normal chatter** and the user was real (e.g. had followers, real name and some past tweets) they would label it as not-fake, if the tweet was a **news segment** or an **ad** a thorough investigation of the tweet would have been done in order to decide if the tweet information was real (the user analysis was also relevant, if it seemed like a fake account there were high chances of labeling the tweet as false). In some cases, one annotator would ask another for opinion in order to best classify a tweet; the dataset used makes no exception from subjective decisions. In the end, there were about 10.000 tweets, that were split into two categories, training and test, they were manually labelled into fake or not and they will be the basis of all the classifications that will be done from here on.

The tweet content is of multiple types: (1) **normal chatter** that can be found on twitter (e.g.: *Good morning friends! Manila looks like London again! Bring out the wellies! Stay dry!*), (2) **news** (e.g.: *London Riots #riots: send in water cannon to clear streets_ Theresa May told: Theresa May_ the Home Secretary_ was where... bio link @ <http://bit.ly/qRNBkO>*), (3) **other ads** (e.g.: *#jJOB: Expert Witness Director (Forensic Delay background) in London_ West End_ bio link @ :P :P <http://www.rengineeringjobs.com/career/311991>*) and (4) **fake news** (e.g.: *We are at #LondonRiotsZoo Attacking the animal at Zoo @a_fisho @blckBss @LondonDude NICE ONE_ Like we for wedge coolly for GH. Man come dey chase #LondonRiotszoo bus!!!!*). This variety of tweets, that include personal touches, but also impersonal speech and fake news, creates a diversity that is very characteristic of the Twitter platform and is very important for the classification algorithms, in order to be able to learn from all possible inputs and scenarios.

In order to not be biased for the algorithm, toward users with high favorites and retweets count (i.e.: *only the very popular on Twitter say the truth*) the dataset does not contain

outliers; the highest retweet count is around 200 and the favorites are around 100, very natural numbers for average twitter users (from various sources, the average being 278 retweets). Also, users that use many hashtags and reference other users in their tweets are more likely to tell the truth, as what they say is based either on the hashtag (that might be, for example, a trend) or in response to another user. Like email spam filters, the algorithm calculates how many tweets contain spam word, in order to increase the elimination rate.

2.2. *Cleaning and pre-processing of data*

However various modifications had to be done to these datasets in order to have consistent data. Such modifications included: discard of tweets that: were not in English, do not have all the attributes necessary (were incomplete), were malformed, had invalid characters, were clearly outliers that would have strongly affected any algorithm or would have led to overfitting. From initial 15.468 raw tweets considered using (Gupta and Kumaraguru, 2012), the processing resulted in about 10.000 tweets that have been splint 70-30, meaning for 7.000 training and 3.000 for test. This pre-processing was done manually, by eliminating the unwanted tweets.

The pre-processing technique uses NLP in order to create a metric of the tweet. The algorithm considers *how many favorites the tweet has, retweets, hashtags, mentions of other users, if it has any URL, if the user swears in the tweet and how many words could be spam* (that are based on statistical evidence, they could be spam, e.g.: “F R E E”).

2.3. *Tweets Features*

The main tweet features were: *Date, Tweet_Text, Tweet_Id, User_Id, User_Name, User_Screen_Name, Retweets, Favorites, Class*. An example of such a tweet is the following: (Date: 2011-08-06 19:59:59, which is in standard date format, Tweet_Text: “RT @joeashtonsing New Guidelines for Product Branding at London 2012 Olympics - Bike Europe <http://bit.ly/pFRMqG>”, which is the tweet content, Tweet_Id: 989000000000000000, User_Id: 67898811, User_Name: *The London bike bot*, User_Screen_Name: 67898811, Retweets: 2, Favorites: 3, Class: 1, which represent the fact that the tweet was manual classified, and it represents a fake tweet).

2.4. *Algorithms*

Testing the model involves selecting appropriate algorithm for the task at hand. The chosen algorithms are: Random Forest, Support Vector Machines and Naïve Bayes, all of which will be tested using different parameters in order to measure the accuracy and performance. The classification algorithms are backed by Spark⁴⁶, a parallel environment, using different techniques provided by the platform and multiple machines communicating in the network, a distributed execution is possible, training and classifying data at a much faster pace than traditional execution environments.

⁴⁶ <https://spark.apache.org/>

3. System Architecture

The system architecture is composed of: the algorithmic part, that is implemented in Python using PySpark, a Flume⁴⁷ agent that collects data from twitter and inserts it into a database (see Figure 1) and a Java web service that users can make request on, in order to collect and classify random tweets from the social platform. The proposed solution follows a Service Oriented Architecture, as we want to have a decoupled system in place.

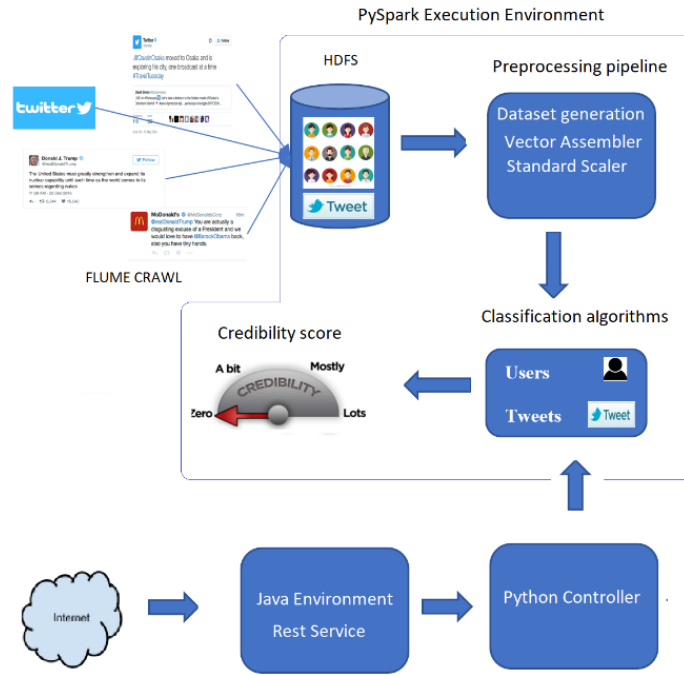


Figure 1: System architecture

3.1. Naïve Bayes

Naïve Bayes by default uses the “multinomial Naïve Bayes” internally, as it classifies data based on every input feature (*Date*, *Tweet_Text*, *Tweet_Id*, *User_Id*, *User_Name*, *User_Screen_Name*, *Retweets*, and *Favorites*). The Naïve Bayes classifier considers each of these features to contribute independently to the probability. The smoothing was set to a default value of 1. Naïve Bayes resulted in an accuracy of 92.43% and it is very fast, on our data with 10.000 tweets (split into 70% for training and 30% for testing). The method is very fast, the execution time being around 2 seconds. The results are presented in Figure 2 (left), where with *red*: we mark Non-Fake tweets from training data set, with *green*: Fake tweets from training data set, with *black*: we mark Non-Fake tweets from test data set and with blue: we mark Fake tweets from test data set.

The confusion matrix for Naïve Bayes classifier on about 3 thousand test data is presented below in Table 1.

⁴⁷ <http://flume.apache.org/>

IDENTIFYING FAKE NEWS ON TWITTER USING NAÏVE BAYES, SVM AND RANDOM FOREST DISTRIBUTED ALGORITHMS

Table 1: The confusion matrix for Naïve Bayes classifier

Naïve Bayes	Predicted No	Predicted Yes
Actual No	135	133
Actual Yes	94	2638

3.2. Support Vector Machine

The Linear SVM Classifier uses a hyperplane to segregate the classes as Fake or Non-Fake in the given training data, i.e., given the labelled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes the test data as Fake or Non-Fake. The SVM is using a linear kernel, the decision function shape is set to “one vs rest” and the error threshold was set to $2^{(-5)}$ in order for the algorithm to perform faster, as a higher error was considered acceptable as long as the execution time was very good. The accuracy of our SVM classifier 95% and the execution time is 8.98 seconds. In Figure 2 (bottom), we mark with *green*: Non-Fake tweets from test data set, and we mark with *blue*: Fake tweets from test data set. The confusion matrix for SVM algorithm on about 3 thousand test data is in Table 2.

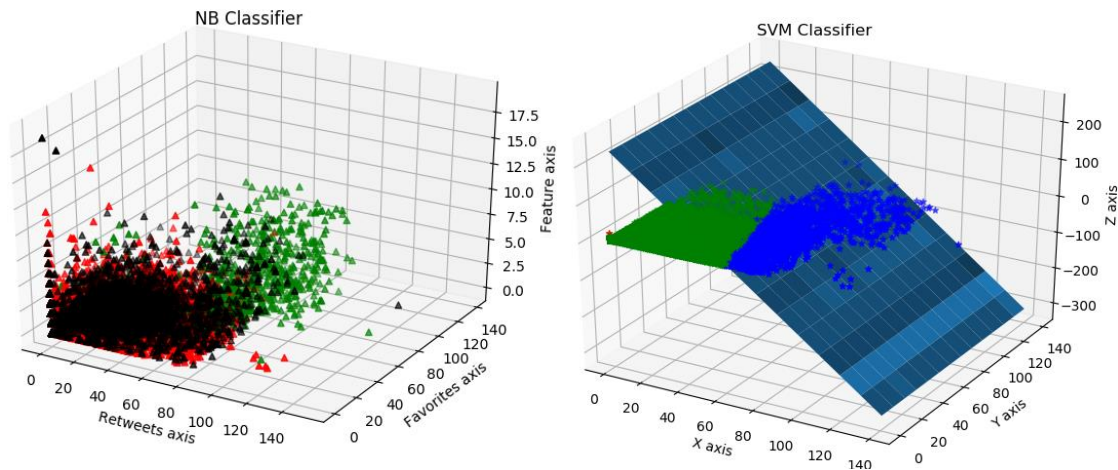


Figure 2: Results obtained with Naïve Bayes classifier (left) and with SVM classifier (right)

Table 2: The confusion matrix for SVM classifier

SVM	Predicted No	Predicted Yes
Actual No	148	71
Actual Yes	79	2702

3.3. Random Forest

The Random Forest classifier operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. Random decision forests correct for decision trees habit of overfitting to their training set. The Random Forest used was set to have a number of 10 decision trees, the impurity was set to Gini⁴⁸ index and the maximum depth was left to a default of 5. The execution time is 3.5 seconds and the accuracy is 95.93%.

⁴⁸ <https://www.investopedia.com/terms/g/gini-index.asp>

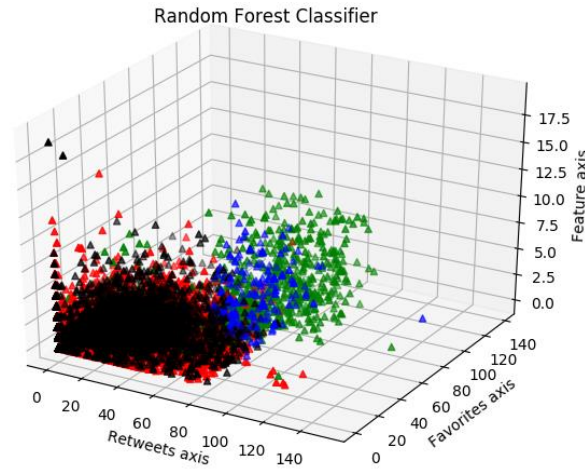


Figure 3: Results obtained with Random Forest classifier

In Figure 3, we have the results, where with *red*: we mark Non-Fake tweets from training data set, with *green*: Fake tweets from training data set, with *black*: we mark Non-Fake tweets from test data set and with *blue*: we mark Fake tweets from test data set. The confusion matrix on about 3 thousand test data is in Table 3.

Table 3: The confusion matrix for Random Forest classifier

Random Forest	Predicted No	Predicted Yes
Actual No	167	60
Actual Yes	62	2711

3.4. Results

Overall, the accuracy for all three methods is presented in Table 4. We can see how the best result was provided by Random Forest algorithm, followed by SVM algorithm.

Table 4: Overall accuracy

Algorithm	Naïve Bayes	SVM	Random Forest
Accuracy	92.43%	95%	95.93%

3.5. Similar solutions

One of the main similar applications available is the Thomson Reuters' Reuters News

Tracer⁴⁹, which is a proprietary algorithm which looks at over 700 features in order to determine whether a tweet is truthful or not. The claimed accuracy is 78 percent for tweets, hence slightly lower than our own algorithm. Unfortunately, the algorithm behind the Reuters News Tracer is proprietary, hence cannot be compared on the same dataset to view key differences. Most solutions, such as widely popular B.S. Detector⁵⁰ rely on curated datasets instead of machine learning, hence classifying the site and not the content itself, which may be an issue depending on how the classification may be awarded. Some look for indicators of

⁴⁹ <https://blogs.thomsonreuters.com/answerson/making-reuters-news-tracer/>

⁵⁰ <http://bsdetecter.tech/>

sensationalism in the titles in order to detect whether it might be a title meant for user to follow the link to the website. This is done using natural language processing, and is limited with the matter currently analyzed, with no context regarding any trends it may be a part of.

The approach closest to the one we propose is the use of Deep Neural Networks (Bajaj, 2017). These allow classifications by looking at multiple features. These types of applications seem to have better results than our solution (for example, in (Bajaj, 2017) authors report a precision of 97%).

4. Use Cases

In next use cases, we will see when our algorithms classify correctly the tweets in fake or non-fake news, and cases when our algorithms incorrectly classify them. We also explain the reasons for which our algorithms decide the type of the tweet: *real* or *fake*.

4.1. Cases with correct classification

Tweets with real information

For tweet with details: Tweet_Text: *“Moments ago, President @realDonaldTrump answered reporter questions about DACA as a government shutdown looms. <http://fxn.ws/2FB0WFa>”*, Date: *“2018-01-15 3:00:21”*, User_Screen_Name: *“foxnews”*, Retweets: *“101”*, Favorites: *“260”* our system decide that the information is real. The tweet is correctly labelled because it refers to actual news, fact that the US president stated some information about a popular topic, DACA. The fact that Fox News is one of the biggest news stations in US weighs a lot in the algorithm final decision (i.e. over 17.1 million of followers).

The tweet with details: Tweet_Text: *“We thank President Trump,'say #IranProtesters, who praise the American public for their support. Here\'s my exclusive with #Iranprotest activists who vow to bring freedom to #Iran and look toward the #USA and @POTUS @realDonaldTrump for help: Watch:”*, Date: *“2018-01-15 3:03:16”*, User_Screen_Name: *“foxnews”*, Retweets: *“81”*, Favorites: *“216”* is correctly labelled, since it refers to a pretty trending subject at the time being. However, the algorithm decision was also biased by the fact that the *“foxnews”* user is pretty popular amongst the social environment in cause, which is a genuine fact, at a first glance upon the user’s number of followers or the posts’ number of retweets and favorites.

For tweet with details: Tweet_Text: *“Dear @POTUS: If you want to honor our troops, here are things more useful than an expensive #militaryparade: - Have a coherent N Korea strategy, - Nominate US Amb to S Korea, - Have a coherent Afghanistan strategy (we've been in 16 years), - Have a coherent Syria strategy, - Deter Russia, <https://t.co/Y2NdFaYdjw>”*, Date: *“2018-02-17 3:12:21”*, User_Screen_Name: *“tedlieu”*, Retweets: *“4345”*, Favorites: *“10726”*, due to the high number of retweets and favourites the algorithm classifies it as real. Also, the writing is very correct and probably the term *“#militaryparade”* was very popular at the moment. The user *“tedlieu”* may also not be reported as a spammer in our dataset.

For tweet with details Tweet_Text: *“There are tens of thousands of homeless veterans on our streets, rather than waste millions of dollars on a #militaryparade use that money to help the women & men that fought for us.”*, Date: *“2018-02-07 3:00:00”*, User_Screen_Name: *“sahluwal”*, Retweets: *“472”*, Favorites: *“956”*, our algorithms correct identify the fact that it

is very popular and also it's associated with a very popular hashtag. The writing is correct probably there are other non-fake tweets in the database that it can match.

Tweets with fake information

The tweet with details: Tweet_Text: "*A cannabis business in Canada is now accepting @StellarOrg \$XLM as a form of payment.*", Date: "2018-01-14 14:05:33", User_Screen_Name: "*StellarRumors*", Retweets: "7", Favorites: "14" has all the trademarks of a fake message. The user doesn't have a lot of followers (i.e. 89) and the tweet has only 7 retweets. The previous pieces of information influence, in a great measure, the label that will be provided by the classification algorithm. The fact that the tweet doesn't contain any link that can be used to verify the authenticity of the information along with the lack of other Twitter accounts references, weighs in the algorithm decision.

The tweet with details: Tweet_Text: "*(Audrey) I champion #Moms #Women #Girls the #Homeless becoming #financially self sufficient under 2018 #Business #Tax laws & regulations! #workfromhome #Resist*", Date: "2018-01-31 3:00:00", User_Screen_Name: "*especiallyme50*", Retweets: "0", Favorites: "0" has the lack of sense and also the lack of interest of the public and suggest the fact that the tweet is shady. Also, the very often use of hashtags makes the algorithm think this is a spam and mark the tweet as fake.

The tweet with details Tweet_Text: "*Must read! #WednesdayWisdom #Military #Budget2018 https://t.co/E0wKI1KMsq*", Date: "2018-02-07 3:45:21", User_Screen_Name: "*momofmonday*", Retweets: "3", Favorites: "3" has a very low number of retweets and of favorites. The tweet is also very short, isn't comprehensive, only some exclamation marks and some hashtags and a link, the algorithm classifies it as clickbait. It is possible that the user is also known in the database as a frequent spammer.

The tweet with details Tweet_Text: "*AMAZING to watch #Democrats & other #Liberals REJOICING over the drop in The Dow Jones Industrial Average. https://t.co/udyQibJnoD*", Date: "2018-02-05 20:38:36", User_Screen_Name: "*T_W_Haines*", Retweets: "0", Favorites: "0" has all the trademarks of a fake message. The usage of capital letters while referring to data. The user has a rather large number of followers, which means that the classification is focused on the message, not on the user himself. The tweet does have no retweets, asserting its controversial status. Also, it refers to a rather large population doing a certain activity, which is rather unverifiable. Although the post is satirical, it is definitely misleading to a user not familiar to the author. Given the negative connotation of the message and the fact that it talks about a large category of individuals, whose actions cannot be properly and profoundly verified, this leads to the conclusion that the message is most likely fake news. The most identifying features of the messages are the capitalized keywords, which the author uses in order to express amazement, while it could also be interpreted as an extraordinary situation under which the subjects of the tweet did not act as expected of them.

4.2. Cases with incorrect classification

Tweets with real information

For the tweet with details: Tweet_Text: "*#NEWS: Exoplanets found to orbit iron-rich stars tightly: http://tinyurl.com/y8p2k2dy*", Date: "2018-01-14 14:23:23", User_Screen_Name: "*spaceanswers*", Retweets: "2", Favorites: "1", the incorrect tag regarding the falsehood of the news' veracity marks the above case as negative labelled, since the exposed subject is not

IDENTIFYING FAKE NEWS ON TWITTER USING NAÏVE BAYES, SVM AND RANDOM FOREST DISTRIBUTED ALGORITHMS

popular/appealing among users. This situation is a typical event, where trends and popularity are a big factor of influence with respect to the news' genuineness verdict. The very low number of retweets and favorites, along with the user's inconspicuousness, makes this piece of news (originally published by NASA) a perfect, common example of incorrect labelling cause.

The tweet with text: Tweet_Text: *"The citizens of #Hodeidah #Yemen suffer greatly bcz they lost their #jobs due to #Saudi #US #UAE blockade, which closed the port of Hodeidah and made them suffer #news #media #press #1000DaysOfWarOnYemen #UN #UNCIF #Trump #FightForRights #FixTrumpIn5Words"*, Date: "2018-01-14 10:18:52", User_Screen_Name: "_0YemenPrincess", Retweets: "17", Favorites: "5" is also incorrectly labelled. It contains accurate information about a current crisis that occurred in Yemen. Even if the text contains a lot of hashtags (e.g. #1000DaysOfWarOnYemen, #FightFor Rights) that increase the tweet credibility, the algorithm decision is strongly biased by the reduced number of retweets and followers.

For the tweet with details: Tweet_Text: *"Happy #NationalSigningDay y'all. Friendly reminder: These are high school kids choosing a college. DO NOT threaten or tweet insults at teenagers because they didn't choose the school you, an adult on twitter, wanted. With that said, #GoVols #NSD18"*, Date: "2018-02-07 17:00:22", User_Screen_Name: "dianneg", Retweets: "58", Favorites: "234", one thing that pops is the excessive use of hashtags, the algorithm might have been biased to think lots of hashtags are associated with fake news. Also the writing is a bit odd, the user uses "DO NOT" and "y'all". The user may be a bit unknown and the tweet doesn't have a lot of retweets or favorites, its average.

For the tweet with details: Tweet_Text: *"추운 겨울의 봄이 되어준 우현아 생일 축하해"*, Date: "2018-02-07 3:00:00", User_Screen_Name: "wooohyunz", Retweets: "12", Favorites: "136", the algorithm doesn't know Korean and this is the main reason for its classification as fake. Second of all the post has a small number of retweets and favorites, and when the algorithm classifies, it matches the tweet with a fake one which is in its database.

Tweets with fake information

The tweet with details: Tweet_Text: *"@6BillionPeople @PACcoinOfficial @PACcoinMan\n50k PAC giveaway!! Today ONLY. Contests ends at 12:01 am January 15!! \n2 winners 25k each!! Let's welcome the launch of PACFYLE!!! \n Share, retweet and put your wallet # in comments.\n #Blockchain #giveaway #bitcoin\n#PacCoin #News\n#PAC"*, Date: "2018-01-14 11:15:42", User_Screen_Name: "PACcoinCZAR", Retweets: "94", Favorites: "55" is incorrectly labelled as real message. This message has all the trademarks of a real message. It contains information about an alleged contest, the required steps that need to be fulfilled to take part and the prize of the contest. Because the user has 3.000 followers along with the fact that the tweet has only 94 retweets in a short time period, biased the algorithm to provide the given label.

The tweet with details: Tweet_Text: *"#Iran #News: urgent, For The world to see,\n This is the reality of Islamic Republic of #Khamenei \n Down with dictator #FreeIran #Iranprotests #Regime Change #UNGA #HumanRights @Asma_ Jahangir @realDonaldTrump @nikkihaley"*, Date: "2018-01-15 3:08:52", User_Screen_Name: "IranNewsUpdate1", Retweets: "220", Favorites: "153" has all the odds of being marked as genuine (numbers are

high on any feature our algorithm analyses). The fact that the tweet is short and contains lots of hashtags, may produce a misfit (as in current example), which is error prone.

The tweet with details: Tweet_Text: *“That dark-skinned #Briton story is more complex than it appears at first glance. Sure the scientists have concluded that Cheddar Man had black skin, but I've just seen a comment from GaryBrexit88 saying it's bollocks so I don't know who to believe.”*, Date: *“2018-02-07 3:32:00”*, User_Screen_Name: *“somegreybloke”*, Retweets: *“1033”*, Favorites: *“2491”* has the hashtag #Briton, which suggests that, as said in the tweet Cheddar Man had black skin. The hashtag is very popular but this tweet seems to contradict everybody. The algorithm assumes it is ok because of high public interest and the very well written tweet, also the #Briton have influenced it.

The tweet with details: Tweet_Text: *“DONT FUCK UP YOUR LIFE #WednesdayWisdom”*, Date: *“2018-02-07 3:45:21”*, User_Screen_Name: *“the_ ironsheik”*, Retweets: *“246”*, Favorites: *“461”* uses profanity and the fact that nothing was truly said makes us label it as a fake news. But the algorithm matched the high retweet count and the favorites to think the tweet is not fake. The user seems rather popular so the algorithm might have associated his popularity with trust.

4.3. Error analysis

After gathering the results from the aforementioned examples, close examinations were considered to ascertain the causes of incorrect labelling. The most conspicuous reason for inaccurate classifications is the user’s feedbacks, which include retweets and the number of people that favorited the article concerned. Since the greatest trust is in the social application’s users, the detection system is heavily influenced on this side, even though the tweet describing the article body of a trustworthy and genuine topic is an actual reliable subject.

Other examples, which were also mentioned in this paper, include tweets posted in a different language than English that were marked as fake, since the Machine Learning model built behind the whole system only contains training data consisting of tweets posted in English.

5. Conclusions

With the comparison of the results among all the classifiers like Support Vector Machine, Random Forest, Naïve Bayes, we have found that Random Forest has the maximum accuracy followed by SVM. Our research work is an attempt to solve a problem commonly encountered in society, namely the ambiguity of understanding information and manipulating citizens.

We plan to expand the work by adding neural network algorithms such as Long Short-Term Memory and other algorithms. The goal is to make continuous improvements to this topic with better precision. Also, in the future, we plan to refine and engaging the algorithms we already use through a more rigorous classification. We plan to create a user-friendly interface that allows easy and intuitive access to fake news: by entering the news title the user will find out whether this is credible or not.

Acknowledgements

We would like to thank the master students involved in developing the first version of the application presented in this paper. This work is partially supported by POC-A1-A1.2.3-G-2015 program, as part of the PrivateSky project (P_40_371/13/01.09.2016).

References

- Atodiresei, C. S., Tănăselea, A., Iftene, A. (2018). Identifying Fake News and Fake Users on Twitter. In *Proceedings of International Conference on Knowledge Based and Intelligent Information and Engineering Systems, KES2018*, 3-5 September 2018, Belgrade, Serbia. Procedia Computer Science 126, 451-461.
- Bajaj, S. (2017). “*The Pope Has a New Baby!*” *Fake News Detection Using Deep Learning*. Stanford University, CS 224N - Winter 2017.
- Conroy, N. J., Rubin, V. L., Chen, Y. (2015). Automatic Deception Detection: Methods for Finding Fake News. In *ASIST 2015*, November 6-10, St. Louis, MO, USA.
- Hadeer, A., Issa, T., Sherif, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In *Proceedings of International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments ISDDC 2017*, 127-138.
- Gupta, A., Kumaraguru, P. (2012). Credibility Ranking of Tweets during High Impact Events. In *PSOSM '12 Proceedings of the 1st Workshop on Privacy and Security in Online Social Media*, 2, 7 pages.
- Idehen, K. U. (2017). Exploitation of a Semantic Web of Linked Data, for Publishers. Open Link Virtuoso Universal Server, <https://medium.com/virtuoso-blog/exploitation-of-a-semantic-web-of-linked-data-for-publishers-295f16ee8525>
- Perez-Rosas, V., Kleinberg, B., Lefevre, A., Mihalcea, R. (2017). *Automatic Detection of Fake News*. Cornell University Library, <https://arxiv.org/abs/1708.07104>.
- Popoola, O. (2017). Using Rhetorical Structure Theory for Detection of Fake Online Reviews. In *Proceedings of the 6th Workshop Recent Advances in RST and Related Formalisms*, 58–63, Santiago de Compostela, Spain, September 4 2017. ACL.
- Rubin, V. L., Lukoianova, T. (2014). Truth and deception at the rhetorical structure level. In *Journal of the American Society for Information Science and Technology*, 66 (5), DOI: 10.1002/asi.23216.
- Rubin, V. L., Conroy, N. J., Chen, Y., Cornwell, S. (2016). Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. In *Proceedings of the Workshop on Computational Approaches to Deception Detection at the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-CADD2016)*, San Diego, California, June 17, 7-17.
- Shu, K., Sliva, A., Wang, S., Tang, J., Liu, H. (2017). Fake News Detection on Social Media: A Data Mining Perspective. In *ACM SIGKDD Explorations Newsletter*. 19 (1): 22-36, DOI: <https://doi.org/10.1145/3137597.3137600>.

- Singh, V., Dasgupta, R., Sonagra, D., Raman, K., Ghosh, I. (2017). Automated Fake News Detection Using Linguistic Analysis and Machine Learning. In *Conference SBP-BRIMS*, 1-3, DOI: 10.13140/RG.2.2.16825.67687.
- Sneha, S., Nigel, F., Shrisha, R. (2017). 3HAN: A Deep Neural Network for Fake News Detection. In *24th International Conference on Neural Information Processing (ICONIP 2017)*, Springer International Publishing AG 2017, Part II, LNCS 10635, 1–10, https://doi.org/10.1007/978-3-319-70096-0_59.
- Wu, L., Liu, H. (2018). Tracing Fake-News Footprints: Characterizing Social Media Messages by How They Propagate. In *WSDM 2018, Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 637-645, DOI: 10.1145/3159652.3159677.

INDEX OF AUTHORS

- Adrian Iftene, 177
Adrian Tosca, 135
Adriana Stan, 33
Alexandru Colesnicov, 53, 77
Alexandru-Lucian Georgescu, 25
Ana-Maria Barbu, 101
Andrei Scutelnicu, 125
Bianca I. Nenciu, 169
Carmen Grigoraș, 13
Cătălina Mărânduc, 113, 159
Cecilia Bolea, 77
Ciprian-Gabriel Cușmaliuc, 177
Corneliu Burileanu, 25
Dan Cristea, 77, 125
Dan Tufiș, 91
Daniela Gîfu, 77, 135
Doina Jitcă, 3
Horia Cucu, 25
Lucia-Georgiana Coca, 177
Ludmila Malahov, 53, 77
Lyudmila Burtseva, 53
Manuela Nevaci, 147
Maria-Dorinela Sirbu, 135
Marius Popescu, 77
Mihaela Onofrei, 77
Mihai Alex Moruz, 125
Mihai Dascalu, 61, 169
Mihai Dascălu, 135
Mircea Giurgiu, 33
Nicolae Saramandu, 147
Otilia Paduraru, 43
Radu Ion, 69
Silviu-Ioan Bejinariu, 147
Stefan Ruseti, 61, 169
Stefan Trausan-Matu, 135
Svetlana Cojocaru, 53, 77
Teodor-Mihai Cotet, 61, 135
Vasile Apopei, 13, 43, 147
Vasile Păiș, 91
Victor Grigoraș, 13
Victoria Bobicev, 113, 159